Interpretability and Privacy for Trustworthy Machine Learning: Bridging Theoretical and User-Centric Perspectives

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Tobias Leemann

aus Ingolstadt

Tübingen

2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	27.03.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Gjergji Kasneci
2. Berichterstatter/-in:	Prof. Dr. Seong Joon Oh

Abstract

With data-driven applications increasingly deployed in high-stakes scenarios, the importance of trustworthy and socially responsible machine learning (TSRML) has been recognized and gained substantial research attention. Moreover, TSRML requirements are successively mandated by recent regulations such as the European Union's General Data Protection Regulation and the Artificial Intelligence (AI) Act. Most prominently, these regulations require careful consideration of explainability and privacy aspects of AI systems. They also make clear that explanations should allow human oversight and protect end users, taking a human-centric perspective. For accountable deployment in user-facing applications, TSRML systems therefore need theoretical guarantees regarding interpretability and privacy, but should be designed around human users. Despite this necessity, we find that the fields of interpretable and private machine learning suffer from a disconnect between foundational theoretical work and user-centered research in practice. To tackle this gap, we explore interpretable and private machine learning from both a theoretical and user-centric viewpoint with the goal of reconciling common misalignments between the two. In the first part of this thesis, we study interpretability through conceptual explanations, counterfactual explanations, and feature attribution methods. We outline how user-friendly conceptual explanations suffer from theoretical identifiability issues and Counterfactual explanations often neglect the requirement for human oversight and the user perspective as they incentivize adversarial actions over real improvements. Finally, popular feature attributions can be theoretically misaligned with underlying predictive models such as transformers. We propose rigorous techniques to reconcile both perspectives for these three explainability techniques. In the second part, we study computational strategies to protect privacy in machine learning and their side effects on users. We show how handing back control over specific features to the users results in a discrimination risk and

how implementing privacy notions such as differential privacy (DP) may result in users being subjected to excessively noisy decisions. We propose strategies to mitigate the two issues, showing that privacy guarantees are often possible without significant impact on predictive performance. Our findings suggest that it is necessary to consider theoretical and user-centric perspectives in conjunction and that tractable tools to reconcile both perspectives often exist. To conclude, we discuss how interpretability and privacy can be combined, explore connections between the two, and outline remaining steps towards practical implementations of TSRML.

Kurzfassung

Durch die zunehmende Nutzung datengestützter Modelle zum Treffen risikoreicher Entscheidungen erfährt das Gebiet des vertrauenswürdigen und sozial verantwortlichen maschinellen Lernens (TSRML) erhöhte Aufmerksamkeit in der Wissenschaft. Darüber hinaus werden die Anforderungen an TSRML nach und nach in Vorschriften der Europäischen Union wie der Datenschutzgrundverordnung (GDPR) und dem gerade in Kraft getretenen AI Act festgeschrieben. Diese Verordnungen verlangen eine sorgfältige Prüfung von auf künstlicher Intelligenz (KI) basierenden Systemen hinsichtlich ihrer Erklärbarkeit und dem Schutz der Privatsphäre. Die Verordnungen machen außerdem deutlich, dass Erklärungen menschliche Kontrolle ermöglichen und die Endnutzer schützen sollten und stellen so menschliche Anwender explizit in den Mittelpunkt. Für einen verantwortungsvollen Einsatz in nutzerorientierten Anwendungen benötigen TSRML-Systeme daher belastbare theoretische Garantien hinsichtlich Erklärbarkeit und Schutz der Privatsphäre, sollten aber trotzdem auf menschliche Nutzer ausgerichtet werden. Trotz dieser Notwendigkeit kommt es in der Forschung im Bereich des erklärbaren und privatsphäreschützenden maschinellem Lernen häufig zu einer Diskrepanz zwischen grundlegenden theoretischen Arbeiten und nutzerzentrierter Forschung. Um diese Lücke zu schließen, wird erklärbares und privatsphäreschützendes maschinelle Lernen hier sowohl von einem theoretischen als auch von einem nutzerzentrierten Standpunkt beleuchtet, um Unstimmigkeiten zwischen den beiden Bereichen zu beseitigen. Im ersten Teil dieser Arbeit wird Erklärbarkeit in Form von konzeptionellen Erklärungen, Erklärungen durch Counterfactuals und Methoden zur Einflussschätzung der Eingangsgrößen betrachtet. Es wird aufgezeigt, wie benutzerfreundliche konzeptionelle Erklärungen unter theoretischen Identifizierbarkeitsproblemen leiden. Im Gegensatz dazu vernachlässigen Erklärungen durch Counterfactuals oft das Erfordernis der menschlichen Aufsicht und der Nutzer-

perspektive, da sie ein Ausspielen des Systems anstelle einer echten Verbesserung fördern. Zuletzt wird gezeigt wie einige Methoden zur Einflussschätzung theoretisch nicht zu den zugrunde liegenden Vorhersagemodellen, wie z. B. Transformern, passen können. In dieser Arbeit werden Verbesserungen für diese drei Erklärbarkeitstechniken hergeleitet, um die beiden Perspektiven wieder in Einklang zu bringen. Im zweiten Teil werden Strategien zum Schutz der Privatsphäre beim maschinellen Lernen und ihre Nebeneffekte für die Nutzer untersucht. Es wird gezeigt, wie die Rückgabe der Kontrolle über bestimmte Daten an die Nutzer zu einem Diskriminierungsrisiko führen kann und wie die Nutzung von Schutzmechanismen wie Differential Privacy (DP) zur Folge haben kann, dass die Nutzer übermäßig vielen Fehlentscheidungen ausgesetzt sind. Es werden Strategien zur Entschärfung dieser beiden Probleme vorgeschlagen, die zeigen, dass der Schutz der Privatssphäre oft ohne signifikante Auswirkungen auf die Vorhersagequalität möglich ist. Die im Rahmen dieser Arbeit gewonnenen Ergebnisse deuten darauf hin, dass es notwendig ist, theoretische und nutzerzentrierte Perspektiven gleichwertig und gemeinsam zu betrachten, und dass es oft praktikable Lösungen gibt, um die beiden Perspektiven wieder miteinander in Einklang zu bringen. Abschließend wird erörtert, wie Strategien zu Erklärbarkeit und Privatssphärechutz kombiniert werden können. Es werden die Verbindungen zwischen beiden Bereichen betrachtet und verbleibende Hindernisse auf dem Weg zur praktischen Umsetzung von TSRML aufgezeigt.

Acknowledgments

A lot has happened between the start of my graduate studies in 2021 during the pandemic and today. I had the opportunity to live in three different cities and to work with three amazing teams. I met many new people and have been supported by colleagues, friends, and family to whom I owe the deepest gratitude.

First and foremost, I would like to thank my supervisor, Gjergji Kasneci, for providing me with this opportunity and for his unwavering support throughout the PhD program. Gjergji gave me a lot of freedom to experiment and develop my own ideas, but always had an open ear when I needed advice. Gjergji was supportive and encouraging of new ideas, even in challenging times. Thank you for being an extraordinary advisor and for always believing in me and my ideas.

I also thank all my great colleagues and co-authors, without whom our work would not have been possible. Particularly, I thank my former colleagues Martin Pawelczyk and Yao Rong, who supported me during the start of my PhD at the University of Tübingen and helped me set foot in academic research. By welcoming me with open arms and including me in their projects, I could really learn a lot. Thank you for being amazing mentors. I would further like to thank Sergül Aydöre for giving me the chance to spend a fantastic summer internship with her team at Amazon Web Services (AWS) AI Labs in New York City.

Even in exhausting times, I could always count on my family's support. I would like to thank my father, Christoph Leemann, and my mother, Katrin Leemann, who always encouraged my curiosity and supported my passion for science and technology wherever they could. I want to thank my sister Annika Leemann, whom I could always rely on.

Lastly, I would like to thank Miriam Felis, who has accompanied me through the last few years with much love, care, and joy. Thank you for reminding me to find the right balance in life and to also look beyond work. I am eager to do so more often with you in the future.

Contents

1

Introduction

1.1 Towards Trustworthy Machine Learning

The number of automated decisions is steadily increasing to keep pace with economic growth. For example, this is illustrated by the number of credit card transactions reaching a record high of 687 billion in 2023, equivalent to 21.7 million transactions *every second* (Statista, 2023). To prevent credit card fraud, data-driven models are deployed by payment infrastructure providers to analyze transactions in real-time (PayPal Editorial Staff, 2024). Commonly known as Machine Learning (ML), these techniques fit complex models to describe past observations and yield the most accurate results on this challenging task (Alfaiz and Fati, 2022). While performance is important (e.g., too many wrongfully blocked transactions lead to customer churn), there are several other characteristics that models deployed in such critical tasks need to fulfill. For instance, they should not discriminate against certain groups and be interpretable and auditable to comply with regulations. This is exemplified in the financial sector, where it has led to simple logistic regression models being preferred over more powerful deep learning approaches (Dastile et al., 2020).

As an increasing number of such high-stakes decisions – including decisions regarding, e.g., hiring (Bogen and Rieke, 2018; Raghavan et al., 2020), credit lending (Dastile et al., 2020; Xia et al., 2017), or even within the justice system (Angwin et al., 2016; Dressel and Farid, 2018) – is being automated, societal consequences of algorithmic decisions must be carefully gauged. This has sparked significant interest in the field of Trustworthy and Socially Responsible Machine Learning (TSRML), which is concerned with developing such reliable and transparent algorithms for high-stakes applications.

1.1.1 Requirements for TSRML in Academic Literature

TSRML is far from being a homogeneous concept. Scholars provide different definitions regarding the elements that it encompasses. For instance, a recent research initiative[1] defines the goal of trustworthy ML as *"to develop and deploy ML models and algorithms that are not only accurate, but also explainable, fair, privacy-preserving, causal, and robust"*. Toreini et al. (2020) identify fairness, explainability, auditability, and safety (FEAS) as important dimensions to build user trust, leveraging theories established in the social sciences. In their framework, confidentiality of data is seen as a subcategory of safety. After reviewing several relevant definitions from previous literature, Rutinowski et al. (2024) name fairness, robustness, integrity, explainability, and safety as the key components to TSRML and further refine safety as *the protection of confidential or proprietary model architectures and parameters as well as data*. In this context, integrity refers to the correctness and reliability of the data and the processing pipeline. While the confidentiality of an ML model's inner working might also be worth protecting, from a user's perspective, we deem the protection of private data most essential. In summary, there seems to be a consensus that TSRML should include fairness, interpretability[2], and notions of safety, including privacy preservation. This aligns well with the perspective taken in key legal frameworks in the European Union (EU), which we consider next.

1.1.2 Requirements for TSRML in European Legal Frameworks

Deploying machine learning systems with trustworthiness properties is not only desirable but is also mandated by law in many jurisdictions. Its principles are manifested in much of recent legislation, particularly in the EU. The EU's General Data Protection Regulation (GDPR, European Parliament, 2016) was seen as one of the most comprehensive privacy and data protection laws at the time of its entry into force in 2018 (Peukert et al., 2022). In this section, we identify key requirements constraining the core ML process, i.e., the training of an ML model.

Requirements in the GDPR. The GDPR's main concern is the protection of personal data from *data subjects*, defined as an *"identified or identifiable natural person from whom or about whom information is collected"* in Art. 4(1) of the GDPR. A fundamental paradigm built into the GDPR is the principle of *data minimization*, which requires personal data only to be stored and processed when it is strictly

[1] https://www.trustworthyml.org/ (accessed 5 January 2025)

[2] We follow Molnar (2019) and use the terms "explainability" and "interpretability" interchangeably, but name "explanations" of individual predictions as such.

necessary for its purpose (Art. 5(1)). However, anonymized data does not fall under the GDPR with its usage limitation. As many data mining and analytics tasks leveraging customer data with ML may not be strictly necessary, anonymization techniques (Stadler et al., 2022) are used to convert personal data into anonymous data. In the legal context, anonymization requires the re-identification of individuals to be unreasonably effortful and highly improbable (Gruschka et al., 2018). This links to the literature on ML privacy, which aims to mathematically quantify the effort and probability of re-identifying certain individuals in either data or trained models (Shokri et al., 2017; Carlini et al., 2023).

Moreover, the GDPR contains a proclaimed "right to explanations", which is more faithfully described as a "right to information" about the logic and consequences of automated data processing (Wachter et al., 2017) in Art. 13–15. Other notable rights include data control rights such as the "right to be forgotten" in Art. 17, which allows users to withdraw their consent to the processing of their data and demand its deletion (Biega and Finck, 2021; Pawelczyk et al., 2023c) and the "right to rectification" in Art. 16, allowing data subjects to correct false information. Finally, the GDPR introduces restrictions for automated individual decision-making. It demands human oversight for high-stakes decisions, de facto ruling out fully automated processing in these scenarios with significant effects or legal implications (Art. 22(1)). In a case against German credit score provider SCHUFA Holding AG, the European Court of Justice ruled that even automated scores, which usually only form a basis for decisions, can breach human oversight requirements (Aza, 2024). In summary, we identify human oversight, a right to explanation/information, data control rights (e.g., the right to be forgotten), and privacy when data is used outside the scope of its original purpose as the main requirements. These requirements are visualized on the left of Figure 1.1.

Requirements in the Artificial Intelligence Act (AIA). The EU's more recent AIA (European Parliament, 2024), which entered into force 1 August 2024, also contains requirements regarding the interpretability and security of data-driven systems, in particular for those categorized as *high-risk artificial intelligence (AI) systems* (Panigutti et al., 2023; Nolte et al., 2024). High-risk AI systems have a particularly severe impact on individuals, e.g., systems deployed in educational and legal contexts, aligning with the application domain of TSRML. For those systems, Section 2 of the AIA formally lays down the following requirements:

- *Risk management system* (Art. 9)
- *Data and data governance* (Art. 10)
- *Technical documentation* (Art. 11)
- *Record-keeping* (Art. 12)

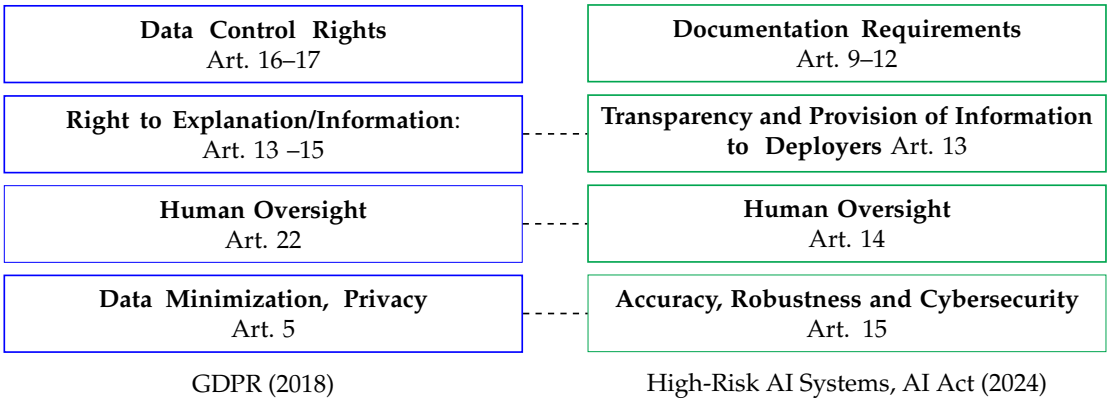


Figure 1.1: Relevant requirements for trustworthy ML found in recent European legal frameworks. We observe that transparency and explanation rights, requirements on privacy preservation and security, and mandates for human oversight can be found in both regulations (indicated by dashed connections).

- *Transparency and provision of information to deployers* (Art. 13)
- *Human oversight* (Art. 14)
- *Accuracy, robustness and cybersecurity* (Art. 15)

While Art. 9 – 12 mainly detail transparency and documentation requirements for the AI system, Art. 13 includes some aspects of interpretability and Art. 14 lays down requirements for human oversights and intervention that need to be implemented in the system. Finally, Art. 15 demands robustness against certain attack scenarios, including *confidentiality attacks* in Art 15(5). We list the corresponding articles in the AIA on the right of Figure 1.1 and focus on the interpretation of the last three articles that we deem most relevant from an ML perspective.

Interpretability in the AIA. Bordt et al. (2022) specifically discuss the role of interpretability in the AIA.[3] They stress that the AIA requires a form of transparency that allows effective oversight by human agents to make an informed decision. Pavlidis (2024) states that explanations are *"a prerequisite for accountability, fairness, public trust, and effective regulation and supervision."*, despite their exact form not being explicitly specified in the AIA. Nevertheless, it is clear that the goal of the explanation is to provide humans with enough context to assess AI decisions, e.g., to identify and intervene on potentially erroneous or dangerous decisions. Additionally, Bordt et al. (2022) outline that explanations are often relevant in adversarial contexts where the explanation provider has an in-

[3]The AI Act was in a draft state at the time of publication of their work. However, the discussed articles on interpretability became part of the final version without modification.

centive to obfuscate the model behavior to defend themselves, e.g., in discrimination lawsuits. They further show that many post-hoc explanation methods are under-determined and the outcome is linked to a variety of parameters that can be adversarially manipulated. We believe that this highlights the importance of unique and well-defined explanations. Non-unique explanations may be easily exploited by adversaries that choose an explanation that fits their "needs" from a potentially infinite space of possible explanations. This is a goal that we contribute to this thesis. The work by Bordt et al. (2022) further highlights that – from a legal perspective – whether humans can successfully work with the explanations is more important than a particular explanation paradigm.

Human oversight in the AIA. Similar to the GDPR, the AIA introduces human oversight requirements. Art. 14(1) requires high-risk AI systems to be developed *"with appropriate human-machine interface tools, that they can be effectively overseen by natural persons during the period in which they are in use."*. Art. 14(4) requires that the system be provided with mechanisms allowing the human controller *"to decide, in any particular situation, not to use the high-risk AI system or to otherwise disregard, override or reverse the output of the high-risk AI system"*. These statements underline the regulators' perspective that human-centric AI is essential for ensuring health, safety, and fundamental rights thereby making the AIA one of the first regulatory works to include a definition of human-centric AI (Enqvist, 2023).

Privacy and safety in the AIA. Nolte et al. (2024) study the role of safety and robustness in the AIA. They argue that the requirements outlined in Art. 15 of the AIA mandate both *adversarial* and *non-adversarial* robustness. Adversarial robustness means resilience to maliciously modified inputs, where non-adversarial robustness refers to resilience against naturally noisy inputs or distribution shifts. They also analyze the different attacks explicitly mentioned in Art. 15(5), namely model poisoning, data poisoning, adversarial examples and model evasion, confidentiality attacks, and other model flaws. These attacks are well-known in the ML literature, where confidentiality attacks include all means to extract information about private data and the model itself, which was not meant to be publicly shared Papernot et al. (2018). Most notably, these include common privacy threats such as training data extraction (Carlini et al., 2023).

From the multitude of requirements, we have thus identified interpretability and privacy of machine learning systems as the most essential technical constraints in the recent AIA and the GDPR. At the same time, the regulators make clear that humans should be in control and oversee the AI systems by formulating strict oversight requirements (cf. Figure 1.1). To implement these regulatory principles, it is evident that solutions for ML interpretability and privacy need to be not only technically sound and rigorous (e.g., to allow for applications such

as model auditing), but also user-centric.

In this thesis, we will consider technically sound implementations of ML interpretability and privacy that also explicitly consider the perspective of human end-users. We discuss connections to other requirements of TSRML, such as fairness and robustness, in the discussion section.

1.2 Technical vs. User-Centric Research

Despite the agreement of computer science and legal scholars on the need for interpretable and privacy-preserving ML, it seems that the importance of including the user perspective for compliance with regulatory frameworks has not been fully realized by parts of the TSRML community yet. We often find a disconnect between purely technical research, mainly pursued by the ML community, and user-centered research conducting human-subject studies, for instance, in the Human-computer Interaction (HCI) community.

1.2.1 Interpretable ML

After comprehensively reviewing user evaluation for explainable AI (XAI),[4] Rong et al. (2023) *"advocate that user-centered methods should be used not only to assess XAI solutions (e.g., through user studies) but also to design them"*. However, user-centered design alone is not sufficient either, as users are subject to several cognitive biases, most notably confirmation bias (Nickerson, 1998). Such biases may be reinforced without rigorous connections between models and explanations. In this thesis, we argue that both points need to be considered in conjunction. The literature on XAI and privacy-preserving ML is peppered with examples where an overemphasis on either human intuition or technical correctness has led to significant oversights. We will revisit some of these learnings in the following.

Explanations can be parameter-invariant. One of the most prominent examples of such an overreliance can be found in the work by Adebayo et al. (2018). The authors introduce the *parameter randomization test* for feature attribution maps. This form of explanation assigns individual inputs (e.g., image pixels) an importance score to quantify its contribution to a decision and can be visualized

[4] While we specifically consider explainable/interpretable ML in this thesis, we still use the standard acronym XAI for this purpose.

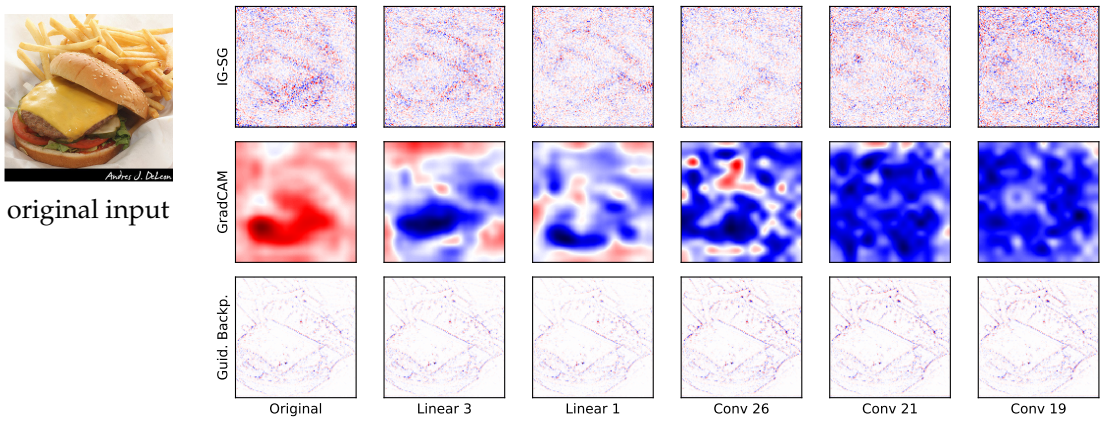


Figure 1.2: Parameter randomization test as proposed in Adebayo et al. (2018) (experimental result reproduced for this thesis). The parameters of network layers are successively replaced with random numbers and the explanations are updated. While IG-SG (Smilkov et al., 2017) and GradCAM (Selvaraju et al., 2017) attributions become increasingly noisy and uninformative, GB (Springenberg et al., 2014) attribution maps remain invariant to parameter randomization, revealing that they are only weakly linked to the underlying prediction model.

through heatmaps. In the proposed test, the weights of the network are successively replaced by random matrices going from the last layer to the first layer. Thereby, the network's performance is reset to the level of random guessing even after replacing only the weights in a single layer. As the modified network only outputs random probabilities we expect the explanations to be similar to random noise as well. Surprisingly, some techniques like Guided Backpropagation (GB, Springenberg et al., 2014) produce explanations that barely change throughout the randomization test as shown in Figure 1.2. This observation highlights that the explanation rather explains the input (similar to edge detectors) than the classification. In their discussion section, Adebayo et al. (2018) conclude that coincidental similarities between edge detectors and saliency methods may be a result of a confirmation bias. It seems very plausible that a machine learning model may use edges to make its decision, potentially leading to the development of model-insensitive techniques. This can be considered a case of *overtrust*, where users are overly confident in explanations (cf. metrics in Rong et al., 2023). Adebayo et al. (2018) note that *to differentiate such methods from model-sensitive explanations, visual inspection is insufficient*, making a case for rigorous theoretical guarantees.

End-to-end explanations are prone to shortcuts. Another prominent example where XAI techniques seem intuitive but do not fulfill their purpose is provided

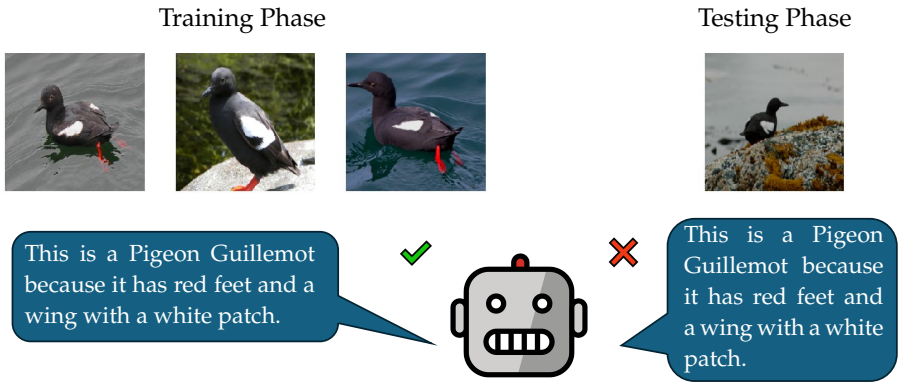


Figure 1.3: End-to-end learned explanations are prone to be unfaithful as they might exploit shortcuts and prior probabilities. Illustrative example based on behavior observed for real models in Hendricks et al. (2018).

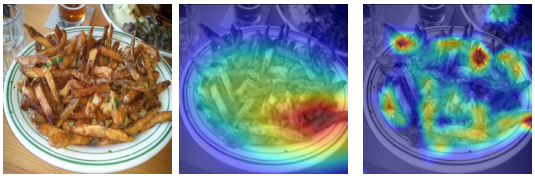
by Hendricks et al. (2018). A stream of works in computer vision (CV) considers the task of providing natural language explanations for image classification models. Following the end-to-end deep learning paradigm, we can learn a self-explaining model with supervision from natural language explanations generated by human annotators: Suppose we have a dataset of triples (x, y, e), where x denotes an input, y denotes a label, and e denotes a textual explanation provided by a human annotator. We can train a self-explaining model $f(x) = (y, e)$ on the task of producing the right prediction together with an explanation by treating the task as a supervised learning task with two targets. This is a simplified version of what was proposed in Hendricks et al. (2016); Park et al. (2018). Unfortunately, such explanation algorithms are prone to exploiting spurious correlations in the data ("shortcuts") and producing unfaithful explanations, as there is no constraint to link the provided explanations to the predictive model. An illustrative example of the mentioned behavior is provided in Figure 1.3. These issues render it important to not only evaluate empirically against a ground truth explanation – early works use scores such as ROUGE (Lin and Hovy, 2003) to assess the correspondence or overlap between generated explanation and human explanations – but to have technical guarantees linking the explanation to the processing or the behavior of the ML model as introduced by the authors in their follow-up work (Hendricks et al., 2018).

While these examples highlight the importance of rigorous links between model and explanation, there have been numerous cases in the literature where technically sound explanations do not align with user needs as well. We will detail two in the following.

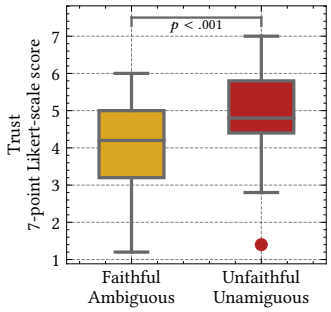
The illusion of explanatory depth. Explanations have been shown to easily

make users overconfident in a model's decision. This corresponds to a phenomenon described as the *illusion of explanatory depth* (IOED, Rozenblit and Keil, 2002) in the cognitive science literature. IOED refers to the observation that *"people feel they understand the world with far greater detail, coherence, and depth than they really do"* (Rozenblit and Keil, 2002). However, when asked to explain certain observations themselves or when their knowledge is systematically tested, their confidence substantially decreases. The effect of this phenomenon was closely studied in the context of XAI by Chromik et al. (2021). To investigate the IOED in XAI, the authors performed a controlled study where users interacted with an AI system for loan assignments. Their subjective (self-rated) understanding was measured through a questionnaire at various stages of the interaction. First, the participants explored an interactive explanation interface based on SHAP feature attribution explanations (Lundberg and Lee, 2017). At a later step, their knowledge was tested by asking the users to simulate the prediction of the model for a novel instance (*forward simulation*) and to pick an instance for which the model would assign the highest risk (*relative simulation*). Their subjective understanding was measured after the interaction, after taking the test, and after the test results were presented to users. The authors discovered a statistically significant decrease ($p < 0.001$) in subjective understanding when participants were confronted with their test results, both compared to the measures taken after the interaction with the system and after taking the test (without knowing the results). This impressively confirmed that the IOED is prevalent in XAI as well.

Unfaithful explanations can still convince users when the outcomes are expected. One of our own works further underlines how fragile user trust can be (Leemann et al., 2023b). In this study, we provided users with two types of explanations on a dataset for food classification. We used standard Grad-CAM feature attribution explanations (Selvaraju et al., 2017), but also created unfaithful explanations of the same type using an entirely different model that had been trained to have similar accuracy but different explanations (see Heo et al. (2019) for details on the procedure). We show an example of the explanations in Figure 1.4a. We recruited $N = 320$ participants online for our study. The total number of participants was divided between four conditions in a 2-by-2 design: First, explanations can be either *faithful* or *unfaithful*. For faithful explanations, we provided the output of the original model together with its explanation. In the condition of unfaithful explanations, we provided an explanation for the other model such that the model whose prediction is shown did not match the model from which the explanations were derived. Second, we showed the users one out of two sets of example images: *unambiguous*, intuitive classification decisions and *ambiguous* examples, where participants in a pre-study disagreed on the correct class. All combinations of the two variables above result in a total of four conditions. Our



(a) Example of faithful and unfaithful explanations used in the described experiment. Unfaithful explanations stem from a differen model. Figure reproduced from Leemann et al. (2023b).



(b) Users place more trust on models with unfaithful explanations on non-ambiguous examples than on faithful explanations with ambiguous examples.

Figure 1.4: Users can trust unfaithful explanations. For instance, if the model decisions confirm their prior beliefs. Figure based on data from Leemann et al. (2023b).

study proceeded as follows: First, we asked the users for classification decisions for five example images (either ambiguous or unambiguous) and showed them corresponding model explanations (either faithful or unfaithful). In the end, the users were presented with a questionnaire to assess their perceived trust in the model. Analyzing the results, we make several observations. While the faithful explanations generally lead to higher trust ($p = 0.04$), we discover that the effect of faithfulness is strongly outweighed by the choice of selected examples: Examples that are unambiguous classification decisions shown along with unfaithful explanations still induce higher trust than examples that are ambiguous along with faithful explanations (Figure 1.4b). We attribute this effect to the role of confirmation bias, leading to higher trust in the model that seems to produce the expected outcome. Our finding highlights that a number of variables determine how users perceive and use explanations. The selection of presented examples has been identified to play a key role, despite not being related to the explanation technique or the model. Confounding factors like the selection of example images may have a substantially higher impact on trust than explanation quality alone. An overly strong focus on technical aspects of XAI techniques may also explain findings by Sixt et al. (2022), who observe that modern XAI techniques do not perform better in terms of human understanding than a simple baseline of showing examples from both classes. We conclude that modeling the user perspective is mandatory for sustained progress in XAI and to fulfill legal requirements.

1.2.2 Privacy-preserving ML

Similar examples can be found in the literature on machine learning privacy. The technical side is usually well-covered in the privacy literature, which has roots in formal disciplines such as information theory and statistics. Common definitions like Differential Privacy (Dwork et al., 2006) are standard in the literature and technically well-understood. However, we observe that the human user is often not explicitly modeled or considered part of the system, which can have severe side effects.

Recurring behavior patterns can compromise anonymization. A prominent example for such a situation is the 2006 Netflix Price described in Narayanan and Shmatikov (2008). Specifically, the data contained movie ratings and the respective timestamps created by around 480,000 Netflix subscribers.[5] The data was anonymized by removing personal data and only releasing a random subset of the ratings. First, the authors showed that knowing only a few movies and ratings is sufficient to identify an individual in the dataset. For instance, knowledge of eight ratings and approximate dates allows to uniquely identify more than 99 % of the individual records in the dataset. Notably, this even holds when two of the ratings are incorrect. The authors further proved that obtaining such incomplete knowledge is quite realistic by crawling profiles from popular movie rating site IMDB. They matched 2 out of 50 crawled IMDB profiles to the Netflix dataset with a confidence near certainty (significance level corresponding to more than 15 standard deviations), thereby leaking the Netflix history associated with these IMDB accounts. Furthermore, some IMDB accounts can be easily linked to real-world names and other social media accounts. This incident resulted in considerable backlash, lawsuits, and a production cancellation at the time (Kamath, 2020). Notably, this is not the only incident of this type, with similar leaks occurring at AOL (Barbaro et al., 2006) and the NYC Taxi & Limousine Commission (Whong, 2014). Such incidents highlight how background knowledge and user actions, if not explicitly modeled, may lead to severe data leakage and business risk. While the anonymization of the Netflix dataset was not directly to blame (although it did certainly not fulfill definitions such as k-anonymity for sensible values of $k > 1$), the fact that user behavior was correlated with information on publicly visible platforms was certainly overlooked, testifying the need to explicitly consider and model the user in privacy-preserving machine learning systems.

While the privacy constraints were too lax in the examples mentioned, works like Kulynych et al. (2023) highlight that overconstraining privacy results in "ar-

[5] While known as a streaming platform today, at the time, Netflix was a DVD-rental service operating via mail

bitrary decisions", meaning that reducing classification performance will result in many inaccurate decisions. Accuracy is still a requirement of TSRML systems, making this angle equally important. This thesis aims to contribute to both ends by explicitly modeling the user and by calibrating privacy notions to realistic attacks.

1.3 Summary of Contributions in This Thesis

We have identified interpretability and privacy as fundamental requirements for TSRML in European legal frameworks such as the recent AIA. Furthermore, these frameworks clearly mandate that explanations should allow human oversight and protect users, taking a human-centric perspective. Many works in the recent literature solely consider either the user perspective or the technical perspective and neglect to model one or the other. This has led to significant oversights in the past.

In this thesis, we are concerned with theoretically grounded algorithms for interpretable and privacy-preserving ML that are catered to user needs and legal requirements. While we perform user research through subject studies in two workshop publications (Leemann et al., 2022, 2023b), in the main part of this thesis we approach the problem by incorporating models of user behavior in the algorithm design. We leverage insights from social sciences where applicable. We consider three common types of explanations: (1) conceptual explanations, (2) counterfactual, and (3) feature attribution explanations. In the privacy literature, we study the de facto standard of Differential Privacy (DP) and define a new notion of how privacy can be preserved when decisions of which data to share are handed back to the user. Specifically, our research work makes the following contributions:

- **A rigorous study of identifiability for conceptual explanations.** Conceptual explanations (Kim et al., 2018) have been proposed as a more human-friendly alternative to attribution methods. However, they often lack technical grounding as concepts often cannot be uniquely identified from data. We contribute to the technical side by studying the uniqueness and identifiability of these explanations, paving the path to making user-centric explanations more theoretically sound and reliable.
- **Counterfactual explanations in light of expert oversight.** The "right to explanations" has motivated the development of counterfactual explanations (Wachter et al., 2017). While the problem is theoretically well understood, counterfactuals have not been studied in light of the human oversight re-

quirements of the GDPR and the AIA, which mandates a human operator in the loop. We contribute to this new direction and derive requirements for counterfactuals that benefit users in this modified scenario.

- **High-fidelity token attributions for transformers.** The recent rise of Large Language Models (LLMs) such as the GPT models (Radford et al., 2018; Bubeck et al., 2023) also brings new challenges for interpretability. We formally prove that models based on the transformer architecture struggle to represent additive models that underpin many explanation techniques. To re-establish the theoretical link between models and explanations, we propose a novel surrogate model for the transformer architecture. We show that our surrogate model corresponds well to the user perception of importance.

- **Handing privacy back to the user: Protecting consent in models with optional features**. This contribution follows the seemingly simple idea of handing privacy back to the user. Instead of trusting an organization with data protection, we consider an ML scenario where users can choose which features they would like to share with an ML model. However, this may not be directly effective as the value of the missing feature might still be inferred and induce a discrimination risk. We propose a strategy to train models that do not make such inferences with minimal performance loss.

- **Calibrating privacy to relevant threat models.** Finally, we consider the ubiquitous definition of differential privacy (DP) and find that its assumptions cover many attack scenarios that may not be realistic in practice. This results in overly strict technical privacy requirements and an inaccuracy risk, disproportionally impacting users through low predictive performance. Instead of following this threat model, we derive an algorithm to provably implement privacy against more realistic membership inference attacks (Shokri et al., 2017). We show that this notion of privacy allows for substantially increased performance, i.e., accuracy.

In summary, we identify several challenges that hinder the integration of technical and user-centric perspectives, thereby slowing progress towards practical TSRML. By explicitly incorporating user models in ML scenarios with interpretability and privacy constraints, we offer constructive recommendations for developing trustworthy and more compliant TSRML solutions. Our findings suggest that theoretical grounding and user needs are not inherently conflicting but can often be addressed simultaneously with the appropriate tools at hand. Equipping practitioners with these tools and raising awareness for common trade-offs and misconceptions are the primary objectives of this thesis.

1.4 Publications

This thesis encompasses *five* first-author publications to which I contributed significantly during the PhD program. Four were published at the internationally renowned conferences UAI, AAAI (awarded an oral presentation), NeurIPS, and XAI, while the fifth was published in the journal TMLR.

Publication 1

Tobias Leemann, Michael Kirchhof, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci: When are Post-hoc Conceptual Explanations Identifiable? *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2023.

Publication 2

Tobias Leemann, Martin Pawelczyk, Bardh Prenkaj, and Gjergji Kasneci: Towards Non-Adversarial Algorithmic Recourse. *World Conference on Explainable Artificial Intelligence (XAI)*, 2024.

Publication 3

Tobias Leemann, Alina Fastowski, Felix Pfeiffer, and Gjergji Kasneci: Attention Mechanisms Don't Learn Additive Models: Rethinking Feature Importance for Transformers. *Transactions on Machine Learning Research (TMLR)*, 2024.

Publication 4

Tobias Leemann, Martin Pawelczyk, Christian Eberle, and Gjergji Kasneci: I Prefer not to Say: Protecting User Consent in Models with Optional Personal Data. *AAAI Conference on Artificial Intelligence*, 2024.

Publication 5

Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci: Gaussian Membership Inference Privacy. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

2

Preliminaries and Related Work

2.1 Interpretability in Machine Learning

Early works on interpretability date back even before the current deep learning boom that started in the early 2010s. Baehrens et al. (2010) proposed to use the local gradient of the classifier at a specific input as an *explanation vector*. Following the rise of deep learning with works such as AlexNet (Krizhevsky et al., 2012) more research contributions considered the topic of ML interpretability: Simonyan et al. (2014) used the input gradients with respect to a class's logits to study the contribution of the inputs to a network's output. This also gave rise of the first group of explanation methods considered in this thesis, which are known as *feature attribution methods*. Early successes sparked a first wave of enthusiasm and research funding. Notably, in 2017 the United States DARPA introduced the explainable AI challenge (Gunning and Aha, 2019), a four-year program to foster ML algorithms with effective explanations and established the term "explainable AI" (Saranya and Subhashini, 2023), abbreviated as XAI. 18 research teams received significant funding to pursue different subtopics in XAI, ranging from example-based explanations to interactive visualizations. The program sparked new ideas and led to promising results. Some of the most common methods to date, including Shapley Additive Explanations (SHAP, Lundberg and Lee, 2017) and Local Interpretable Model-Agnostic Explanations (LIME, Ribeiro et al., 2016) were developed around this time.

However, it was soon realized that the problem of explaining ML models was more difficult than initially anticipated. Methods were critically examined, such as in the study by Adebayo et al. (2018) mentioned in the introduction. Fundamental problems of XAI include the lack of a clear definition for interpretability and agreement on metrics for quantifying it (Nauta et al., 2023). On the user-

centric side, studies like Sixt et al. (2022) showed that modern XAI techniques are not more helpful for users to detect biases in models than a simple baseline of showing different examples of the classes in their scenario. Thus, while significant research effort has been invested in XAI, many fundamental problems remain unresolved.

In the following, we will briefly introduce the most relevant XAI techniques used in the remainder of this thesis. The techniques we consider in this work are post-hoc techniques, i.e., they are applied on a trained model. We can differentiate between *global* and *local* techniques that are valid for the entire model or only for a specific input-output pair, respectively. Our focus lies on local methods.

2.1.1 Feature Attribution Methods

In the post-hoc explanation setting, we consider a non-linear function $f : \mathbb{R}^n \to \mathbb{R}^d$ that is element of a function space $\mathcal{F}$. Local feature attribution methods $\mathbf{a}$ map a model, an input, and a target class to a numerical importance score, i.e., $\mathbf{a} : \mathcal{F} \times \mathbb{R}^n \times \{1, \ldots, d\} \to \mathbb{R}$. One of the simplest ways to assign feature importances is by simply using the gradient of the logit for class $i \in 1, \ldots, d$ as feature attribution (Simonyan et al., 2014), i.e.,

$$\mathbf{a}_{\text{grad}}(f, \mathbf{x}, i) = \nabla f_i(\mathbf{x}). \tag{2.1}$$

However, this primitive form of explanation has been shown to be quite noisy (Smilkov et al., 2017) as input sensitivity is not directly linked to the prediction (Shah et al., 2021). Improved techniques have been developed to deal with these issues. An inherent challenge of feature attribution methods is their evaluation. Evaluation is notoriously difficult as there usually are no ground truth importances to compare with (Rong et al., 2022). A common technique is to measure the impact of feature removal (i.e., replacing a feature by a baseline value) on the predictive outcome (Hooker et al., 2019; Tomsett et al., 2020).

Such removals or feature perturbations can also be directly used to derive explanations. Besides gradient-based techniques, there is a class of model-agnostic explanations that can be applied to any ML model and only require black-box function access. Most notably, they include Local Interpretable Model-Agnostic Explanations (LIME, Ribeiro et al., 2016) and SHAPley Additive Explanation (SHAP, Lundberg and Lee, 2017). LIME is a framework for *surrogate model* explanations. Surrogate models are simple, directly interpretable models such as linear models and trees (Molnar, 2019). A surrogate model ζ describing the neighborhood around instance $\boldsymbol{x} \in \mathcal{X}$ is determined by minimizing the following

objective (Ribeiro et al., 2016):

$$\zeta(\boldsymbol{x}) = \arg\min_{g \in \mathcal{G}} \mathcal{L}(f, g, \pi_x) + \Omega(g), \tag{2.2}$$

where f denotes the original predictive model, g denotes the surrogate model from function class $\mathcal{G}$. $\mathcal{L}$ describes a loss function that quantifies the alignment between f and g in a local neighborhood $\pi_{\boldsymbol{x}}$ around the instance $\boldsymbol{x}$. For instance, we can define

$$\mathcal{L}(f, g, \pi_x) = \mathbb{E}_{z \sim \pi_{\boldsymbol{x}}} \left[(f(\boldsymbol{z}) - g(\boldsymbol{z}))^2 \right]. \tag{2.3}$$

The neighborhood $\pi_{\boldsymbol{x}}$ can be a continuous distribution, e.g., a Gaussian, around $\boldsymbol{x}$ (Agarwal et al., 2021), or feature discrete removals of certain features (Ribeiro et al., 2016). The second component $\Omega(g)$ constitutes a regularization term constraining the complexity of g to keep it as interpretable as possible. We study surrogate model explanations in this thesis and uncover that despite the general formulation, popular linear surrogate models may fail to effectively approximate some predictive models such as the transformer (Vaswani et al., 2017).

Contrarily, SHAP explanations (Lundberg and Lee, 2017) are rooted in game theory based on the well-established Shapley value (Shapley, 1953), proposed to fairly distribute payouts across a team of players in a game. In the context of XAI, the individual input features can be interpreted as the team members, and the payout corresponds to the model's prediction. The Shapley value for a feature is computed as a weighted average of this feature's marginal contributions, i.e., the change in output when adding the feature to each possible subset of other features. This formalization has some convenient theoretical properties, e.g., additivity (also known as efficiency), stating that the contribution of the individual features sums up to the final score. On the other hand, an exact approximation of the Shapley value can be prohibitively expensive as the number of marginal contributions grows exponentially with the number of features. However, reasonably accurate and efficient approximations exist (Lundberg and Lee, 2017; Kolpaczki et al., 2024).

2.1.2 Conceptual Explanations

After a substantial amount of feature attribution methods were proposed (cf. Arrieta et al., 2020), researchers became increasingly aware of their inherent limitations. For instance, they cannot model higher-level interactions and concepts (Kim et al., 2018). Their noisiness further makes them hard to interpret (Selvaraju et al., 2017).

Kim et al. (2018) proposed conceptual explanations in a framework *Testing with Concept Activation Vectors (TCAV)* as a more human-friendly alternative. Instead of tracing back a decision to input features, their goal is to quantify the contribution of high-level concepts to a decision. Most prominently used in computer vision, concepts refer to parts of an image with a higher-level meaning, e.g., "feet", "stripes", or "water" (Bau et al., 2017) that are more relatable to human decision-making. Based on the observation that neural networks often encode meaningful semantic information in linear directions in their learned latent space (Bau et al., 2017; Szegedy, 2013), the authors suggested using directions in the latent space to represent the concepts and refer to them as *concept activation vectors* (CAVs). To identify CAVs, the authors train a linear "probe" model to identify the direction of a concept from annotated samples and their latent representations. Few human labels specifying the absence or presence of a concept are needed to train a linear classifier to infer the presence of the concept in an image from its latent representation. The authors use this classifier's weights as the CAV.

Improved Conceptual Explanations. The original method by Kim et al. (2018) comes with a few drawbacks. First of all, their method requires human labels. However, human annotations are prone to bias and may be incomplete, i.e., the human labelers may not identify a relevant concept used by the classifier. This requirement has been relaxed or completely removed in follow-up works (Ghorbani et al., 2019; Akula et al., 2020). Furthermore, some researchers argue that concepts tend to be "local" in images, e.g., some part of the image shows the feet while another part shows the head of a bird (Mu and Andreas, 2020). Locality is not incorporated in the original approach by Kim et al. (2018). A solution for both problems is proposed in Yeh et al. (2020). The authors present completeness-aware conceptual explanations. They propose an objective to discover meaningful concepts by clustering latent representations of image patches. Their objective also features a completeness term, which ensures that the concept contributions are predictive of the target class and make sure that no essential concepts used by the predictive model are missed. They further quantify the contribution of each concept through the well-established SHAP framework (Lundberg and

Figure 2.1: Two discovered concepts from the Animal with Attributes (AwA) dataset (Xian et al., 2018). While some concepts have a semantic meaning, others do not appear to be coherent. Reproduced from Leemann et al. (2022).

Lee, 2017). While empirical results are promising, the concepts discovered are not guaranteed to have a high-level meaning, as shown in Figure 2.1. In this thesis, we will investigate and mitigate some underlying causes from an identifiability perspective.

2.1.3 Counterfactual Explanations

The paradigm of counterfactual explanations was initially proposed by Wachter et al. (2017) as a potential way to satisfy the acclaimed "right to explanations" in the EU's GDPR for black-box classifiers. Provided with an input $\boldsymbol{x} \in \mathcal{X}$, counterfactual explanations are concerned with finding an instance $\boldsymbol{x}'$ similar to $\boldsymbol{x}$ leading to a different classification decision. The counterfactual explanation problem can be formalized as (Freiesleben, 2022):

$$\underset{\boldsymbol{x}' \in \mathcal{X}}{\operatorname{argmin}}\; d_1(\boldsymbol{x}, \boldsymbol{x}') + \lambda d_2(f(\boldsymbol{x}'), y_t) \tag{2.4}$$

In the above formulation, d_1 and d_2 are distance functions in the input and the label space, respectively, and $\lambda \in \mathbb{R}_+$ is a weight parameter. The objective finds instances $\boldsymbol{x}'$ close to the original instance $\boldsymbol{x}$, resulting in a prediction close to the target class label y_t. The parameter λ controls the weighting of the two objectives. For discrete labels, d_2 is often chosen as the Dirac-distance, which is infinite for $f(\boldsymbol{x}') \neq y_t$, thereby constraining solutions to have the desired label. d_1 can be any reasonable distance in the input space, for instance, l_1 or l_2 norms.

Counterfactual explanations can be used to provide *recourse* to users adversely affected by automated decisions: For instance, in case of a loan denial, a counterfactual can provide insights helping users to be awarded the loan in the future, by showing them an instance with similar features and a positive decision. Unfortunately, there are several drawbacks of the simple formulation in Eqn. (2.4) that have been addressed in recent work. We list some of the challenges below:

- The standard objective does not incorporate feature constraints, e.g., a person's age and highest education level can only increase monotonically. Works like Ustun et al. (2019) provide frameworks to incorporate such constraints in the objective.
- The counterfactuals generated are not necessarily in-distribution, i.e., the objective may produce an individual of age 16 holding PhD degree. This has sparked interest in producing on-manifold recourse (Pawelczyk et al., 2020; Poyiadzi et al., 2020). A common strategy is to use a latent variable

model such as Variational Auto Encoders (Kingma and Welling, 2013) to model the feature distribution and find suitable counterfactuals in the latent space (Pawelczyk et al., 2020).

- Counterfactual explanations are not robust to slight changes in the predictive model, for instance, triggered by updates to the predictive model (Pawelczyk et al., 2023c) or users only implementing a non-exact version of the given counterfactual (Pawelczyk et al., 2023a). Robust optimization techniques can be used to mitigate some of these problems (Upadhyay et al., 2021).

We study another challenge of counterfactual explanations in this thesis. We discover that counterfactual explanations may fail to serve their purpose when applied in use cases with human oversight, which is a fundamental requirement of the GDPR and the AIA. In this scenario, we need counterfactuals that convince the human oversight board. However, when the model and the ground truth (labels assigned by the human oversight board) slightly differ, the counterfactuals may only change the model's prediction but not convince the human experts. The concurrent application of these legal principles poses new challenges that are not well covered in the literature yet.

2.2 Privacy in Machine Learning

Besides interpretability, we identified privacy as a key requirement for TSRML. The EU AIA mentions "confidentiality attacks" (Art. 15), that are defined in Papernot et al. (2018) as *"attempts to expose the model structure or parameters [...] or the data used to train and test it"*. Recent works have shown that it is possible to extract a substantial amount of training data from large language models (Nasr et al., 2023a) or diffusion models for image generation (Carlini et al., 2023). As much of the training data may contain personal information that is not intended to be publicly disclosed, e.g., user e-mails, chats, or photos,[6] this constitutes a significant privacy breach. Protecting ML models from such attacks is the central goal of privacy-preserving ML.

While there is a broad agreement that such leaks should be prevented, defining privacy for ML remains challenging because some degree of memorization is necessary for a model's functioning as well: For instance, the model should know that the capital of France is Paris but it should not output an individ-

[6] see examples listed here: https://www.washingtonpost.com/technology/2023/09/08/gmail-instagram-facebook-trains-ai/ (accessed 5 January 2025)

ual's phone number. In the 2000s, incidents such as the 2006 Netflix Price mentioned in the introduction of this thesis (Narayanan and Shmatikov, 2008) highlighted the need for reliable privacy definitions. This resulted in notions such as k-anonymity (Sweeney, 2002) and differential privacy (Dwork et al., 2006) being proposed. Concurrently, paradigms such as federated learning (McMahan et al., 2017) ensure privacy during model training by keeping the training data distributed across multiple devices, e.g., phones. Only the updates needed to improve the ML model are shared, enhancing privacy by allowing users to keep the data on devices they control.

This thesis considers the standard setup where full or partial data is shared with the model developer under privacy concerns. These concerns include protection against the three most common threats outlined in the next section. We consider ensuring privacy at two stages or the process: We study the setups where users have some choice over which data to share prior to training in addition to the standard setup of guaranteeing privacy at training time.

2.2.1 Attacks on Machine Learning Models

Several attacks on machine learning models are formalized in the privacy literature. They usually include a specific *threat model*, that exactly specifies the background knowledge available to the attacker and the goal of the attack. This allows to compare the effectiveness of different attack implementations and the vulnerability of models and datasets against each other. We will briefly discuss the most common threat models in this section.

Membership Inference Attacks. A fundamental privacy attack on machine learning models is the Membership Inference Attack (MIA, Shokri et al., 2017). In this attack scenario, the attacker has access to a data record $\boldsymbol{x}$ and is interested in determining whether this record was part of the training set used to fit an ML model. This can be formalized as follows:

Definition 2.1 (Membership Inference Experiment, Yeom et al., 2018)**.** *Let $\mathcal{A}$ be an attacker, A be a learning algorithm, k be a positive integer, and $\mathcal{D}$ be a distribution over data points $\boldsymbol{x} \in D$, where the vector $\boldsymbol{x}$ may also denote a tuple of data and labels. The membership inference experiment proceeds as follows:*

- *Sample $S \sim \mathcal{D}^k$ (i.e., sample k points i.i.d. from $\mathcal{D}$) and train $A_S = A(S)$*
- *Choose $b \in \{0, 1\}$ uniformly at random*
- *Draw $\boldsymbol{x}' \sim \mathcal{D}$ if $b = 0$, or $\boldsymbol{x}' \sim S$ if $b = 1$.*

- *The attacker is successful if* $\mathcal{A}(\boldsymbol{x}', A_S, k, \mathcal{D}) = b$. $\mathcal{A}$ *must output either 0 or 1.*

Useful variations of the threat model are discussed in (Ye et al., 2022), which consider a fixed model or an average over different records sampled from the distribution $\mathcal{D}$. For the sake of generality, we focus on the standard threat model above in this thesis. Its intuitive formalization makes evaluation simple and makes the attack amenable to theoretical analysis. This has led to a vast literature on MIA strategies and defenses (Hu et al., 2022). Besides pure success rates, it is commendable to evaluate the attack success using a trade-off curve between false positive rate (FPR) and true positive rate (TPR). As mentioned in Carlini et al. (2022), having a non-negligible TPR at a very low FPR may result in some instances being identified with very high certainty, which may already constitute a privacy breach. This breach may not be visible in overall attack accuracy, which can still be only slightly above random guessing accuracy in this case.

Training Data Extraction Attacks. The importance of MIAs can also be derived from their use as a building block of more powerful attacks. For instance, they are a crucial step in the generate-then-rank framework (cf. Carlini et al., 2021) for training data extraction attacks. In this attack strategy, a generative model is instructed to generate potentially private data using specifically curated prompts (for instance, Nasr et al., 2023a, prompt a model to infinitely repeat the word "poem" which results in arbitrary data being generated after several repetitions). In the next attack step, a MIA is used to assess the probability of this data being part of the training dataset. This pipeline is successful in reconstructing parts of a model's training data from scratch, an attack known as *training data extraction* or *model inversion* (Zhang et al., 2020).

A common threat model for this attack is the *untargeted black-box attack* as considered in Carlini et al. (2021) for language models (LMs). Here, the attacker only has black-box access to a model and can compute the probability of arbitrary input sequences. This includes next-word prediction and completion requests and corresponds to the interface available at common LLM providers such as OpenAI.[7] The attacker is interested in obtaining as many training data sequences as possible.

Model Stealing Attacks. Besides attacks targeting the training data, the model parameters themselves can be attacked. The business model of many LLM service providers or financial companies, e.g., credit-scoring companies, relies on secret models that can only be queried in a black-box manner. Model stealing attacks try to reconstruct the model from systematic queries and have been suc-

[7] cf. API at https://platform.openai.com/docs/api-reference/chat/create (accessed 6 December 2024)

cessfully scaled up even to LLMs (Carlini et al., 2024). Linking to the ML interpretability part of this work, some explanation techniques, such as counterfactual examples, can also be leveraged to orchestrate such attacks (Aïvodji et al., 2020).

Linking MIAs and reconstruction attacks. As we are mostly concerned with protecting user data in this thesis, we focus on attacks targeting the training data. In particular, we consider MIAs. They are well-established in the literature and their use in reconstruction attacks suggests that protecting against this type of attack rules out data reconstruction threats as well. This intuition can be verified through a proof by contradiction that we would like to briefly outline here as we were not able to retrieve it in the literature:

Proof Sketch. We show that *the impossibility of MIAs implies the impossibility of reconstruction attacks*. To start the proof, we suppose that MIAs were impossible, but reconstruction of a substantial part of the training data was possible with only few false positives. In this case, one could construct a simple MIA by first running the reconstruction attack to obtain a part of the training dataset and by subsequently checking if a membership query sample is in that reconstructed part of the training set. If the sample is found, we return true. As the reconstruction is assumed to be sufficiently good and broad, this contradicts the assumption that MIAs are not possible.

We conclude that protection against MIAs also protects against other threats like reconstruction and consider this threat model a cornerstone of privacy protection. For additional references and threat models, we refer the reader to the survey papers by Rigaki and Garcia (2023) and specifically to Hu et al. (2022) regarding MIAs.

2.2.2 Differential Privacy

Despite the large variety of attacks, the literature on machine learning privacy has converged on differential privacy (DP) as a standard for formally defining privacy preservation. DP offers powerful theoretical guarantees that have been shown to prevent or substantially harden the most relevant attack scenarios, including MIAs (Thudi et al., 2022; Yeom et al., 2018). DP enforces the privacy of an algorithm by ensuring that its output distribution remains largely unchanged if a single data point in the input dataset is inserted or deleted (Dwork et al., 2006). An algorithm is said to be ε-DP if the probability of it producing a specific outcome in a set E for a dataset D is almost the same as the probability of it producing an outcome in E for a slightly different dataset D', which differs

from D by just one element (D and D' are known as *neighboring* datasets). This constraint can be relaxed by including a term δ, which allows the DP constraint to be violated for a minimal share of outcomes δ. Formally, differential privacy (DP) can be defined as follows:

Definition 2.2 (Differential Privacy, Dwork et al., 2006)**.** *A randomized algorithm M gives (ε, δ)-differential privacy if for all data sets D and D' differing on at most one element, and all $E \subseteq Range(M)$, if*

$$\mathbb{P}\left[M(D) \in E\right] \leq \exp(\varepsilon)\mathbb{P}\left[M(D') \in E\right] + \delta, \tag{2.5}$$

where the probability is taken is over the coin tosses of M. If $\delta = 0$, the guarantee is called ε-DP.

Advantages of DP include its generality that protects against various abstract threat models, and DP's desirable composition properties. These allow the combination of multiple DP mechanisms and computing the DP parameters of the joint mechanism (Kairouz et al., 2015; Dong et al., 2022). For instance, they are exploited when DP is implemented for SGD-trained models through privacy-preserving gradient descent.

Differentially-private stochastic gradient descent (DP-SGD). The most common algorithm to implement DP for machine learning models is DP-SGD (Abadi et al., 2016). DP-SGD introduces two major modifications to the gradient updates of standard SGD:

- First, after the gradient of the loss for a sample is computed, the gradient vector is cropped to a maximum norm C. As the l_∞-norm is commonly used, this corresponds to projecting all individual elements of larger absolute value than C back to an absolute value of C.
- Second, after the cropping stage, the gradients are noised by adding zero-mean Gaussian noise (usually of small magnitude).

Otherwise, the batch gradients are averaged and used to update the model parameters following the usual SGD manner. The amount of noise can be derived from the desired ε, δ-values leveraging the aformentioned composition strategies (known as privacy accountants, e.g., the moments accountant by Abadi et al., 2016) that compute the privacy leakage across many steps. Unfortunately, the algorithm may drastically reduce model performance as shown by Stadler et al. (2022) and in our own work (Leemann et al., 2023a). The DP guarantees derived for DP-SGD have been shown to be tight (Nasr et al., 2023b), suggesting that the performance cannot be improved by deriving a better algorithm, but that the definition of DP itself may need to be refined to allow for higher utility.

3

Contributions

In this chapter, we outline our five distinct contributions to the topic of TSRML. We begin with three contributions to the interpretability literature and present our two contributions concerning privacy in the latter two sections. For each contribution, we first present a summary followed by a discussion of the contribution in the realm of TSRML.

3.1 Uniquely Identifiable Conceptual Explanations

Publication 1

Tobias Leemann, Michael Kirchhof, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci: When are Post-hoc Conceptual Explanations Identifiable? *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2023.

Author Contributions. The topic of conceptual explanations initially arose in a discussion between Yao Rong, Enkelejda Kasneci, Gjergji Kasneci, and me. I contributed the specific direction for this paper and the initial formalization. Michael Kirchhof refined the idea in the identifiability direction and contributed significantly to formalizing the theoretical results. I conducted the main experimental study and the synthetic data experiments, Michael Kirchhof contributed the robustness experiment and an application to another dataset, and Yao Rong contributed the real-world experiments on the CUB dataset. Enkelejda Kasneci and Gjergji Kasneci supervised all stages of the research process and contributed significantly to the final write-up by restructuring our findings and narrative. They also helped develop visualizations to better convey our motivation.

Summary. Conceptual explanation techniques explain decisions in terms of high-level concepts such as objects, colors, or shapes. These are seen as a user-friendly alternative to feature attributions. Usually, no human concept labels on a sample level are available, so *concept discovery methods* search trained embedding spaces for interpretable concepts like objects or colors that can be used to provide post-hoc explanations for decisions. However, the conditions under which unsupervised concept discovery is reliably possible are not well understood from a theoretical standpoint. Most notably, it is unclear whether it is possible to provably re-identify concepts in learned latent representations, even when the true number of data-generating concepts is known. ML models may use arbitrary directions to encode the concepts in their latent space. Even if the concepts are known to be encoded linearly, there is no general criterion to effectively disentangle them. We find that identifiability is only possible when concepts are statistically independent, a condition that we deem unrealistic in practice. Instead, we devise more realistic conditions that make the problem identifiable: The independent mechanisms analysis (IMA) and disjoint mechanisms analysis (DMA) criteria. These methods leverage feature attribution methods and demand that different concepts should activate spatially disjoint parts of an input image. For instance, in bird classification, the concept "head" should be encoded in a different part of the image than "tail". We argue that these conditions are more realistic, particularly for compositional processes, and devise a new concept discovery algorithm that finds the concepts leading to the highest spatial disjointness and is provably identifiable.

3.1.1 Discussion

This publication contributes towards technically rigorous and well-defined conceptual explanations. Our work shows that conceptual explanations are often ill-defined as there is no way to uniquely identify the data-generating concepts even when their number is known à priori. To arrive at well-defined conceptual explanations, we need to identify the conditions under which they can be uniquely identified. Having non-identifiable concepts opens the door to abuse, as an adversary can choose an arbitrary explanation from a set of possible explanations that suits their needs (Bordt et al., 2022). It also explains observations as in Leemann et al. (2022), where non-meaningful or multi-semantic concepts are recovered as there is simply no criterion to provably recover concepts in the general case. The findings in this work also have potential implications on the recent research on LLM "mechanistic interpretability" (Saphra and Wiegreffe, 2024; Templeton et al., 2024), which attempts to learn factorized representations from latent spaces or identify neurons that correspond to a single concept. Our

work suggests this may be an ill-fated endeavor for practical data, as the identifiability conditions are unlikely to be met. This view is confirmed by works such as Hase et al. (2023), who show that knowledge is usually distributed over a larger part of an ML model. Besides showing that the standard problem is not uniquely identifiable, we contribute to the literature by providing two additional constraints that make the problem identifiable. We argue that future conceptual explanation algorithms should specify such constraints to make the conditions under which they succeed explicit.

Orthogonal to this work, we study which concepts are considered interpretable from a human perspective in a workshop contribution (Leemann et al., 2022), finding that objects are considered most interpretable by human users. This is reflected in the identifiability criteria we propose, e.g., disjoint mechanisms, which is catered towards compositional scenes, that can be decomposed into individual objects. Thereby our methods outline a promising route towards reliable and user-friendly conceptual explanations.

When are Post-hoc Conceptual Explanations Identifiable?

Tobias Leemann[1,2,†] **Michael Kirchhof**[1,†] **Yao Rong**[1,2] **Enkelejda Kasneci**[2] **Gjergji Kasneci**[2]

[1]University of Tübingen, Tübingen, Germany
[2]Technical University of Munich, Munich, Germany
[†]equal contribution

Abstract

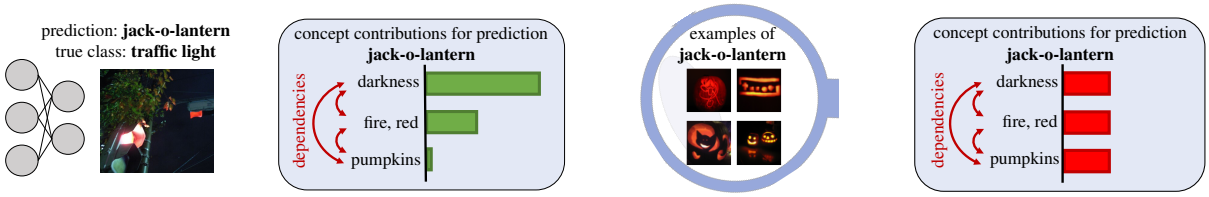
Interest in understanding and factorizing learned embedding spaces through conceptual explanations is steadily growing. When no human concept labels are available, concept discovery methods search trained embedding spaces for interpretable concepts like *object shape* or *color* that can provide post-hoc explanations for decisions. Unlike previous work, we argue that concept discovery should be *identifiable*, meaning that a number of known concepts can be provably recovered to guarantee reliability of the explanations. As a starting point, we explicitly make the connection between concept discovery and classical methods like Principal Component Analysis and Independent Component Analysis by showing that they can recover independent concepts under non-Gaussian distributions. For dependent concepts, we propose two novel approaches that exploit functional compositionality properties of image-generating processes. Our provably identifiable concept discovery methods substantially outperform competitors on a battery of experiments including hundreds of trained models and dependent concepts, where they exhibit up to 29 % better alignment with the ground truth. Our results highlight the strict conditions under which reliable concept discovery without human labels can be guaranteed and provide a formal foundation for the domain. Our code is available online.

1 INTRODUCTION

Modern computer vision systems represent images in embedding spaces. These are either constructed implicitly in higher-level layers of large models or explicitly through generative models such as Variational Autoencoders (Kingma and Welling, 2013) or recent Diffusion Models (Song and Ermon, 2019; Ho et al., 2020). To unveil why an image is considered similar to a certain class, interest in understanding these embeddings is increasing. Conceptual explanations (Crabbé and van der Schaar, 2022; Muttenthaler et al., 2022; Akula et al., 2020; Kazhdan et al., 2020; Yeh et al., 2019; Kim et al., 2018) are a popular explainable AI (XAI) technique for this purpose. They scrutinize a given encoder by decomposing its embedding space into interpretable concepts post-hoc, i.e., after training. Subsequently, these concepts form the basis of popular post-hoc explanations such as TCAV (Kim et al., 2018) or allow high-level interventions (Koh et al., 2020). Fig. 1 outlines a real-world example. A misclassification made by a pretrained model shipped with the `pytorch` library (Paszke et al., 2017) is to be explained. In the given example, the conceptual explanation allows identification of a spurious correlation that the model has picked up: Most jack-o-lanterns are found in combination with dark backgrounds, which causes it to mistake the traffic light at night for a jack-o-lantern.

Constructing such explanations is non-trivial. The key ingredient to all conceptual explanation techniques is a set of interpretable concepts, which is notoriously hard to specify (Leemann et al., 2022). It is frequently defined through human annotations (Crabbé and van der Schaar, 2022; Koh et al., 2020; Kim et al., 2018) on individual samples of the dataset that can be prohibitively expensive (Kazhdan et al., 2021). Furthermore, it is usually unknown which concepts will be leveraged by a machine learning model without a model at hand. Therefore, we consider fully unsupervised concept discovery (Ghorbani et al., 2019; Yeh et al., 2019), where the concepts are automatically discovered in the data. Concepts are frequently modeled as directions in a given embedding space (Ghorbani et al., 2019; Kim et al., 2018; Yeh et al., 2019), which have to be discovered without supervision. These embedding spaces can be highly distorted, making it hard to correctly separate the influences of individual concepts. However, this is essential to make the right inferences in practice (see Fig. 1d). This intuition is supported by prior work on generative models (Ross et al.,

Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), PMLR 216:1207–1218.



(a) **Misclassification:** A model makes an incorrect prediction. A user is interested in understanding why this incident happened.

(b) **Conceptual Explanation:** Concept contributions are computed that explain the prediction. In this example, the concept "darkness" is relevant for the outcome.

(c) **Inspection:** A closer inspection of samples from the predicted class reveals that most images in this class have a dark background; a spurious correlation picked up by the model.

(d) **Entangled Conceptual Explanation:** It is essential to correctly split up the contribution of individual concepts to allow for valid inferences.

Figure 1: Schematical use-case of conceptual explanations: A misclassification of an image classifier is explained. The example is based on a real explanation for a ResNet50 model. Details and the original explanation are provided in App. C.8.

2021), which has shown that user understanding is strongly linked to the representations' respective disentanglement.

While many methods have been empirically shown to work well, a rigorous theoretical analysis of the conditions under which concept discovery is possible is still lacking in previous works. We propose to consider concept discovery methods that are *identifiable*. This means when a known number of *ground truth components* generated the data, the concept discovery method provably yields concepts that correspond to the individual ground truth components and can correctly represent an input in the concept space. This is a crucial requirement: If a method is even incapable of recovering known components, there is no indication for its reliability in practice. In this work, we are the first to investigate identifiability results in the context of post-hoc concept discovery.

First, we find that identifiability results from Principal Component Analysis (PCA) and Independent Component Analysis (ICA) literature (Jolliffe, 2002; Comon, 1994; Hyvärinen et al., 2001) can be transferred to the conceptual explanation setup. We establish that they cover the case of independent ground truth components with non-Gaussian distributions. This is insufficient for two reasons: (1) In practice, concepts such as height and weight (Träuble et al., 2021) or wing and head colors of birds often follow complex dependency patterns. (2) Popular generative models (Kingma and Welling, 2013; Song and Ermon, 2019) frequently work with an embedding space with a Gaussian distribution.

As a second contribution, we seek to fill this void by providing an identifiable concept discovery approach that can handle dependent and Gaussian ground truth components. We can show that this is possible through taking the nature of the image-generating process into consideration. Specifically, we propose utilizing *visual compositionality properties*. These are based on the observation that tiny changes in the components frequently affect input images in orthogonal or even disjoint ways. These properties of image-generating processes also leave a "trace" in the encoders learned from a set of data samples. This insightful finding permits to construct two novel post-hoc concept discovery methods based on the *disjoint* or *independent mechanisms* criterion. We prove strong identifiability guarantees for recovering components, even if they are dependent. Our results highlight the strict and nuanced conditions under which identifiable concept discovery is possible.

In summary, our work advances current literature in multiple ways: **(1)** We present first identifiability results for post-hoc conceptual explanations. We find that results from ICA can be transferred under the assumption of independent ground truth components. **(2)** For the more intricate setting of dependent components, we propose the *disjoint mechanism analysis (DMA)* criterion and the less constrained *independent mechanism analysis (IMA)* criterion. We prove that they recover even dependent original components up to permutation and scale. **(3)** We construct DMA and IMA-based concept discovery algorithms for encoder embedding spaces with the same theoretical identifiability guarantees. **(4)** We test them (i) on embeddings of several autoencoder models learned from correlated data, (ii) with multiple and strong correlations, (iii) on discriminative encoders, and (iv) on the real-world CUB-200-2011 dataset (Wah et al., 2011). Our approaches maintain superior performance amidst increasingly severe challenges.

2 RELATED WORK

Works on the analysis and interpretation of embedding spaces touch a variety of subfields of machine learning.

Concept discovery for explainable AI. Conceptual explanations (Koh et al., 2020; Kim et al., 2018; Ghorbani et al., 2019; Yeh et al., 2019; Akula et al., 2020; Chen et al., 2020b) have gained popularity within the XAI community. They aim to explain a trained machine learning model post-hoc in terms of human-friendly, high-level concept directions

(Kim et al., 2018). These concepts are found via supervised (Koh et al., 2020; Kim and Mnih, 2018; Kazhdan et al., 2020) or unsupervised approaches (Yeh et al., 2019; Akula et al., 2020; Ren et al., 2022), such as clustering of embeddings (Ghorbani et al., 2019). However, their results are not always meaningful (Leemann et al., 2022; Yeh et al., 2019). Therefore, we suggest approaches with identifiability guarantees. We provide initial identifiability results and a novel approach, which can be used for unsupervised concept discovery under correlated components.

Independent Component Analysis (ICA). Independent Component Analysis (Comon, 1994; Hyvärinen and Pajunen, 1999; Hyvärinen et al., 2001) or blind source separation (BSS) consider a generative process $\boldsymbol{g}(\boldsymbol{z})$ as a mixture to undo and rely on traces that the distribution of the generating components $\boldsymbol{z}$ leaves in the mixture. In this work, we show that an identifiability result from ICA can be transferred to the conceptual explanation setup, but recovery is only possible under independent underlying components of which all but one are non-Gaussian. This result is not applicable to naturally correlated processes, which is why we design a novel method for this case.

Disentanglement Learning. Concurrently, literature on disentanglement learning is concerned with finding a data-generating mechanism $\boldsymbol{g}(\boldsymbol{z})$ and a latent representation $\boldsymbol{z}$ for a dataset, such that each of the original components (also known as factors of variation) is mapped to one (controllable) unit direction in $\boldsymbol{z}$ (Bengio et al., 2013). An alternative definition relies on group theory (Higgins et al., 2017) where certain group operations (symmetries) should be reflected in the learned representation (Painter et al., 2020; Yang et al., 2021). Most works in the domain enhance VAEs (Kingma and Welling, 2013) with additional loss terms (Higgins et al., 2017; Burgess et al., 2018; Kim and Mnih, 2018; Chen et al., 2018). Despite recent progress it is not always possible to construct disentangled embedding spaces from scratch: Locatello et al. (2019) have shown that the problem is inherently unidentifiable without additional assumptions. A more recent work by Träuble et al. (2021) shows that even if just two components of a dataset are correlated, current disentanglement learning methods fail. In this work, we focus on post-hoc explanations of embedding spaces of given models, which are usually entangled.

Identifiability results. Identifiability questions have been raised in domains such as Natural Language Processing (Carrington et al., 2019) or in disentanglement learning, which is most related to this work. It has been previously shown that unsupervised disentanglement, without further conditions, is impossible (Hyvärinen and Pajunen, 1999; Locatello et al., 2019; Moran et al., 2022). Hence, recent works aim to understand the conditions sufficient for identifiability. One strain of work relies on additional supervision, i.e., access to an additional observed variable (Hyvärinen et al., 2019; Khemakhem et al., 2020) or to tuples of observations that differ in only a limited number of components (Locatello et al., 2020). Gresele et al. (2021) and Zheng et al. (2022) proved identifiable disentanglement under independently distributed components and introduce a functional condition on the data generator. We also consider functional properties, but our setting is different as (1) we have access to a trained encoder only and (2) not even partial annotations or relations are available.

3 ANALYSIS

In this section, we formalize post-hoc concept discovery to provide an identifiability perspective. We find that Independent Component Analysis (ICA) and Principal Component Analysis (PCA) only guarantee identifiability when the ground-truth components are stochastically independent. We then study the intricate case of dependent components and propose using *disjoint* and *independent mechanisms analysis* (DMA / IMA) along with identifiability results. All proofs are provided in App. B.

3.1 PROBLEM FORMALIZATION

In post-hoc concept discovery, we are given a trained encoder $\boldsymbol{f} : \mathcal{X} \to \mathcal{E}$ with embeddings $\boldsymbol{e} = \boldsymbol{f}(\boldsymbol{x}) \in \mathcal{E} \subset \mathbb{R}^K$ of each image $\boldsymbol{x} \in \mathcal{X}$. We do not impose any restriction on how $\boldsymbol{f}$ was obtained; it can be the feature extractor part of a large classification model, or a feature representation learned through autoencoding, contrastive learning (Chen et al., 2020a), or related techniques. Interpretability literature seeks to understand the embedding space by factorizing it into concepts. Based on the observations that directions in the embedding space often correspond to meaningful features (Szegedy et al., 2013; Bau et al., 2017; Alain and Bengio, 2016; Bisazza and Tump, 2018), these concepts are frequently defined as direction vectors $\boldsymbol{m}_i$ (Kim et al., 2018; Ghorbani et al., 2019; Yeh et al., 2019). These are commonly referred to as concept activation vectors (CAVs). Hence, the combined output of a concept discovery algorithm is a matrix $\boldsymbol{M} = [\boldsymbol{m}_1, \dots, \boldsymbol{m}_K]^\top \in \mathbb{R}^{K \times K}$ where each row contains a concept direction.

We seek a theoretical guarantee on when these discovered concept directions align with ground truth components that generated the data. To this end, we formalize the data-generating process as shown in Fig. 2: There are K ground-truth components with scores $z_k, k = 1 \dots K$, summarized $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^K$, that define an image. The term *components* always refers to the ground truth as opposed to the *concepts*, which denote the discovered directions. A data-generating process $\boldsymbol{g} : \mathcal{Z} \to \mathcal{X}$ generates images $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z}) \in \mathcal{X} \subset \mathbb{R}^L$, $L \gg K$. A powerful algorithm should be able to recover the original components. That is, there should be a one-to-one mapping between entries of $\boldsymbol{Me}$ and the entries in $\boldsymbol{z}$, up to the arbitrary scale and order

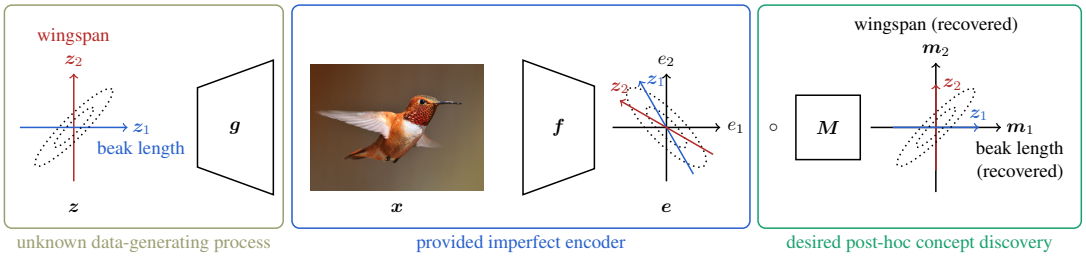


Figure 2: Overview over the concept discovery setup. We consider a process where data samples $\boldsymbol{x}$ are generated from possibly correlated ground truth components $\boldsymbol{z}$, e.g., a wingspan or beak length of a bird, by an unknown process $\boldsymbol{g}$ (left). The high-dimensional data is mapped to the to the embedding space of a given model $\boldsymbol{f}$ (center). A suitable post-hoc concept discovery yields concept vectors $\boldsymbol{m}_i$ that correspond to the original components (right).

of the entries. We say that a concept discovery algorithm *identifies* the true components if it is guaranteed to output directions $\boldsymbol{M}$ that satisfy $\boldsymbol{Me} = \boldsymbol{Mf}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{PSz}$ $\forall \boldsymbol{z} \in \mathcal{Z}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation matrix that has one 1 per row and column and is 0 otherwise, and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is an invertible diagonal scaling matrix.

To make the problem solvable in the first place, concept directions must exist in the embedding space of the given encoder, requiring $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ is of full rank. Depending on the scope of the conceptual explanation desired, it can be sufficient for the components to exist in a local region of the embedding space if the concept discovery algorithm is only applied around a region around a certain point of interest. This only changes the meaning of $\mathcal{E}, \mathcal{X}$, and $\mathcal{Z}$ but is formally equivalent.

3.2 IDENTIFIABILITY VIA INDEPENDENCE

Initially, we turn towards classical component analysis methods. We find that their identifiability results use non-correlation or even stronger stochastic independence assumptions of the ground truth components.

Principal Component Analysis (PCA) (Jolliffe, 2002) uses eigenvector decompositions to find orthogonal directions $\boldsymbol{M}$ that result in uncorrelated components $\boldsymbol{Me}$. This means that PCA is only capable of identifying the original components if the ground truth components $\boldsymbol{z}$ were uncorrelated and exist as orthogonal directions in our embedding space. In our setup and notation, this leads to the following result:

Theorem 3.1 (PCA identifiability) *Let $z_k, k = 1, \dots, K$, be uncorrelated random variables with non-zero and unequal variances. Let $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ is an orthonormal matrix. If an orthonormal post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{K\times K}$ results in mutually uncorrelated components $(z'_1, \dots, z'_K) = \boldsymbol{z}' = \boldsymbol{Me}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diagonal matrix where $|s_{ii}| = 1$ for $i \in 1, \dots K$.*[1]

All proofs in this work are deferred to App. B. It is arguably a strong condition that the ground truth directions are encoded orthogonally in the embedding space. Independent Component Analysis (ICA) overcomes this limitation and allows for arbitrary directions. However, the classic result by Comon (1994) even demands stochastically independent components. Transferred to our setup and notation, the result can be stated as follows.

Theorem 3.2 (ICA identifiability) *Let $z_k, k = 1, \dots, K$, be independent random variables with non-zero variances where at most one component is Gaussian. Let $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ has full rank. If a post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ results in mutually independent components $(z'_1, \dots, z'_K) = \boldsymbol{z}' = \boldsymbol{Me}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a perm. and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diag. matrix.*

This result shows that stochastic independence of the ground truth components leaves a strong trace in the embeddings that can be leveraged. Algorithms like `fastICA` (Hyvärinen and Oja, 1997) can find the concept directions $\boldsymbol{M}$ by searching for independence (Comon, 1994). We conclude that ICA is suited for post-hoc concept discovery under independent components.

In summary, we have transferred two results from the component analysis literature to the setup of post-hoc conceptual explanations. However, these results do not allow to recover components that are correlated or follow a Gaussian distribution. This limits their applicability in practice where concepts often appear pairwise (e.g., darkness and jack-o-lanterns, cf. Fig. 1). We will bridge this gap in the remainder of this paper by introducing two new identifiable discovery methods based on functional properties of the generation process that we term *disjoint* and *independent* mechanisms. A summary of identifiability results is provided in Tab. 1.

[1] To simplify notation, $\boldsymbol{P}$ and $\boldsymbol{S}$ mean *any* permutation and scale matrices. They do not have to be equal between the theorems.

Dependency	Marginal Dist.	Transform	Criterion
uncorr.	uneq. variances	orthogonal	non-correlation (PCA)
independent	non-Gaussian	invertible	independence (ICA)
arbitrary	arbitrary	invertible	disj. mechanisms (DMA)
arbitrary	arbitrary	invertible	indep. mechanisms (IMA)

Table 1: PCA and ICA provably identify concepts via their distributions. DMA and IMA utilize functional properties.

3.3 IDENTIFIABILITY VIA DISJOINT MECHANISMS

Instead of placing independence assumptions on $\boldsymbol{z}$, we propose a concept discovery algorithm that makes use of natural properties of the generative process $\boldsymbol{g}$. In particular, generative processes in vision are often compositional (Ommer and Buhmann, 2007): Different groups of pixels in an image, like a bird's wings, legs, and head, are each controlled by different components. Effects of tiny changes in components are visible in the Jacobian $\boldsymbol{J_g}$, where each row points to the pixels affected. Thus, a compositional process will follow the *disjoint mechanisms* principle.

Definition 3.1 (Disjoint mechanism analysis (DMA)) *$\boldsymbol{g}$ is said to generate $\boldsymbol{x}$ from its components $\boldsymbol{z}$ via disjoint mechanisms if the Jacobian $\boldsymbol{J_g}(\boldsymbol{z}) \in \mathbb{R}^{L\times K}$ exists and is a block matrix $\forall \boldsymbol{z} \in \mathcal{Z}$. That is, the columns of $\boldsymbol{J_g}(\boldsymbol{z})$ are non-zero at disjoint rows, i.e. $|\boldsymbol{J_g}(\boldsymbol{z})|^\top |\boldsymbol{J_g}(\boldsymbol{z})| = \boldsymbol{S}(\boldsymbol{z})$, where $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diagonal matrix that may be different for each $\boldsymbol{z}$ and $|\cdot|$ takes the element-wise absolute value.*

Note that this definition does not globally constrain the location of affected pixels. The components may still alter different but disjoint pixels for each image. In real concept discovery, we do not have access to the generative process $\boldsymbol{g}$ but can only access the encoder $\boldsymbol{f}$. However, an encoder corresponding to $\boldsymbol{g}$ will not be arbitrary and its Jacobian $\boldsymbol{J}_f \in \mathbb{R}^{K\times L}$ will have a distinct form in practice: First, to maintain the component information the composition $\boldsymbol{f} \circ \boldsymbol{g}$ will be of the form $\boldsymbol{f}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{D}\boldsymbol{z}$, with a yet unknown matrix $\boldsymbol{D} \in \mathbb{R}^{K\times K}$. Furthermore, we expect encoders to be rather lazy, meaning they only perform the changes to invert the data generation process but are almost invariant to input deviations not due to changes in the components. This is in line with the classic interpretability literature, where gradients of models were observed to noisily highlight the relevant input features (Baehrens et al., 2010; Simonyan et al., 2013) and form the basis of popular attribution methods such as Integrated Gradients (Sundararajan et al., 2017). Technically, the changes effected by the components form the linear $\text{span}(\boldsymbol{J_g}(\boldsymbol{z}))$, whereas entirely external changes are given in its orthogonal complement $\text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp$. Thus, for $\mathbf{v} \in \text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp \subset \mathbb{R}^L$ the encoder should not react to these change and the corresponding gradients of the encoder for these changes should be zero, i.e., $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))\mathbf{v} = \mathbf{0} \Leftrightarrow \mathbf{v} \in \ker(\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})))$.

Definition 3.2 (Faithful encoder) *$\boldsymbol{f}$ is a faithful encoder for the generative process $\boldsymbol{g}$ if the ground truth components remain recoverable, i.e., $\boldsymbol{f}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{D}\boldsymbol{z}$, for some $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ with full rank. Furthermore, $\boldsymbol{f}$ is lazy and invariant to changes in $\boldsymbol{x}$ which cannot be explained by the ground truth components, requiring $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))$ and $\boldsymbol{J_g}(\boldsymbol{z})$ to exist and $\text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp \subseteq \ker(\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})))$, $\forall \boldsymbol{z} \in \mathcal{Z}$.*

Having defined what realistic encoders look like through the notion of faithful encoders, we find that there is distinct property which can be leveraged to discover the directions in $\boldsymbol{M}$ among faithful encoders: It is sufficient to find an encoder $\boldsymbol{M}\boldsymbol{f}$ whose Jacobian $\boldsymbol{M}\boldsymbol{J}_f$ will have disjoint rows. Intuitively, this means searching for components whose gradients affect disjoint image regions.

Theorem 3.3 (Identifiability under DMA) *Let $\boldsymbol{g}$ have disjoint mechanisms and $\boldsymbol{f}$ be a faithful encoder to $\boldsymbol{g}$. If a post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{K\times K}$ of full rank results in disjoint rows in the Jacobian $\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))$, i.e., $|\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))||\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))|^\top$ is invertible and diagonal for some $\boldsymbol{z} \in \mathcal{Z}$, then $\boldsymbol{M}\boldsymbol{e} = \boldsymbol{P}\boldsymbol{S}\boldsymbol{z}$ where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a scaling matrix.*

This theorem does not impose any restrictions on the distribution $\boldsymbol{z}$, making it applicable to realistic concept discovery scenarios through leveraging the nature of the generative process. The proof of this algorithm in App. B.5 also yields an analytical solution. We will use it to verify conditions in a controlled experiment in Sec. 4.1. We have thus identified the *DMA criterion* that is sufficient to discover the component directions when the rows of $\boldsymbol{M}\boldsymbol{J_f}$ point to disjoint image regions. We can formulate this as a loss function and optimize for $\boldsymbol{M}$ via off-the-shelf gradient descent:

$$\mathcal{L}(\boldsymbol{M}) = \mathbb{E}_{\boldsymbol{x}} \|\text{arn}\left[\boldsymbol{M}\boldsymbol{J}_f(\boldsymbol{x})\right] \text{arn}\left[\boldsymbol{M}\boldsymbol{J}_f(\boldsymbol{x})\right]^\top - \boldsymbol{I}\|_F^2. \tag{1}$$

The expectation is taken over a collection of real data samples $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z})$. The arn-operator (absoute values, row normalization) takes the element-wise absolute value and subsequently normalizes the rows. This does not constrain the norms of the Jacobian's rows but only enforces disjointness.

3.4 CONCEPT DISCOVERY VIA INDEPENDENT MECHANISMS

We can perform an analogous derivation for a class of generating processes that is more general. Grounded by causal principles instead of compositionality, the independent mechanisms property has been argued to define a class of natural generators (Gresele et al., 2021).

Definition 3.3 (Independent mechanism analysis (IMA)) *$\boldsymbol{g}$ is said to generate $\boldsymbol{x}$ from its components $\boldsymbol{z}$ via independent mechanisms if the Jacobian $\boldsymbol{J_g}(\boldsymbol{z})$ of $\boldsymbol{g}$ exists and its columns (one per component) are orthogonal $\forall \boldsymbol{z} \in \mathcal{Z}$, i.e., $\boldsymbol{J_g}^\top(\boldsymbol{z})\boldsymbol{J_g}(\boldsymbol{z}) = \boldsymbol{S}(\boldsymbol{z})$, where $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diagonal matrix that may differ for each $\boldsymbol{z}$ (Gresele et al., 2021).*

Gresele et al. (2021) and Zheng et al. (2022) used this characteristic to find disentangled data generators, but we can again transfer characteristics via faithful encoders: This time we find that searching for an $\boldsymbol{MJ}_f$ with *orthogonal* (instead of disjoint) rows permits post-hoc discovery of concepts. We refer to is property of $\boldsymbol{MJ}_f$ as the *IMA criterion*.

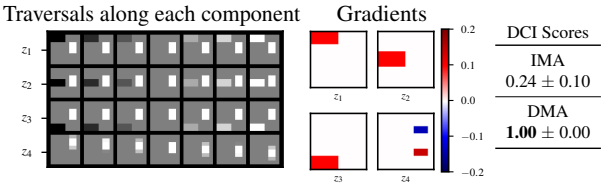
However, as the class of admissible processes has been increased, it is not strong enough to ensure identifiability in the most general case. This is prevented under an additional technical condition on the component magnitudes, which we refer to as *non-equal magnitude ratios* (NEMR). Intuitively, the magnitudes of the component gradients have to change non-uniformly between at least two points for the conditions to be sufficient. If there were two factors that always attribute to input pixels in the same way (imagine the sky being partitioned into two components termed "left sky" and "right sky"), they cannot be told apart anymore since there can be other mixtures which would result in orthogonality (they could equally be "lower sky" and "upper sky").

Theorem 3.4 (Identifiability under IMA) *Let $\boldsymbol{g}$ adhere to IMA. Let $\boldsymbol{f}$ be a faithful encoder to $\boldsymbol{g}$. Suppose we have obtained an $\boldsymbol{f}' = \boldsymbol{Mf}$ with a full-rank $\boldsymbol{M} \in \mathbb{R}^{K\times K}$ and orthogonal rows in its Jacobian $\boldsymbol{MJ_f}(\boldsymbol{g}(\boldsymbol{z})) \coloneqq \boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))$, i.e, $\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))^\top = \boldsymbol{\Sigma}(\boldsymbol{z})$ where $\boldsymbol{\Sigma}(\boldsymbol{z})$ is diagonal and full-rank at two points $\boldsymbol{z} \in \{\boldsymbol{z}_a, \boldsymbol{z}_b\}$. If additionally $\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}$ has unequal entries in its diagonal (NEMR condition), then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a scaling matrix.*

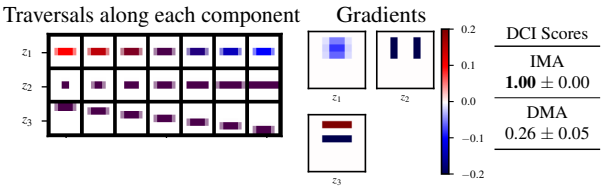
The constructive proof in App. B.6 can also be condensed into an analytical solution. Alternatively, one can again construct a suitable optimization objective for the IMA criterion, i.e., orthogonal Jacobians. This is achieved by removing the absolute value operation from the arn-operator in Eqn. (1), so that it solely performs a row-wise normalization. In summary, we have established the novel DMA and IMA criteria that allow concept discovery under dependent components.

4 EXPERIMENTS

In the following, we perform a battery of experiments of increasing complexity to compare the practical capabilities of approaches for identifiable concept discovery. We start by verifying the theoretical identifiability conditions (Sec. 4.1), then perform evaluation under increasing multi-component correlations for embedding spaces of generative and discriminative models (Sec. 4.2 to 4.4), and finally use a large-scale, discriminatively-trained ResNet50 encoder (Sec. 4.5).



(a) `FourBars`: DMA datasets can be solved by the DMA criterion.



(b) `ColorBar`: IMA datasets can be solved by the IMA criterion.

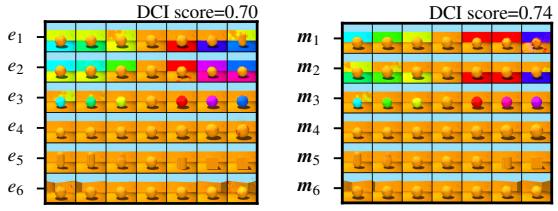
Figure 3: Experiments on two synthetic datasets: We confirm our analytical results and show that DMA (a) and IMA (b) cover visual concepts such as colors and translations.

We borrow the DCI metric (Eastwood and Williams, 2018) from disentanglement learning with scores in $[0, 1]$ to measure whether each discovered component predicts precisely one ground-truth component and vice versa. Following Locatello et al. (2020), we report additional metrics with similar results in App. D, along with results on additional datasets and ablations. For reproducibility, each experiment is repeated on five seeds and code is made available upon acceptance. In total, we train and analyze over 300 embedding spaces, requiring about 124 Nvidia RTX2080Ti GPU days. More implementation details are in App. C.

4.1 CONFIRMING IDENTIFIABILITY

We first confirm our identifiability guarantees with the analytical solutions. To this end, we implement two realistic synthetic datasets with differentiable generators. This allows computing the closed form of $\boldsymbol{J_g}$ and deliberately fulfilling or violating the DMA, IMA, and NEMR conditions.

`FourBars` consists of gray-scale images of four components: Three bars change their colors (black to white) and one bar moves vertically, showing that the image regions affected by each component may change in each image. The plot of $\boldsymbol{J_g}$ in Fig. 3a shows that each component maps to a disjoint image region. This fulfills DMA and thus also IMA. However, all factors have the same gradient magnitudes, making it impossible to find two points with NEMR. According to our theory, we expect DMA optimization to work and IMA to fail as NEMR is essential to make to proof of Thm. 3.4. The second dataset, `ColorBar`, contains a



(a) Autoencoder (DipVAE) (b) Autoencoder + DMA (ours)

Figure 4: DMA discovers directions $\boldsymbol{m}$ that control individual concepts (wall & floor color) of Shapes3D although they are confused in the original embedding space (e_1, e_2, ...).

single bar that undergoes realistic changes in color, width, and its vertical position, see Fig. 3b. It conforms to IMA and NEMR but not DMA. Our proofs indicate that IMA should work, and DMA should fail. Completing the problem formalization in Sec. 3.1, we compute analytical faithful encoders $\boldsymbol{f}$ for these datasets distorted by a random matrix $\boldsymbol{D}$. The solutions behave as expected: On `FourBars` only the DMA criterion delivers perfectly recovered components (DCI=1) whereas on `ColorBars` only IMA succeeds.

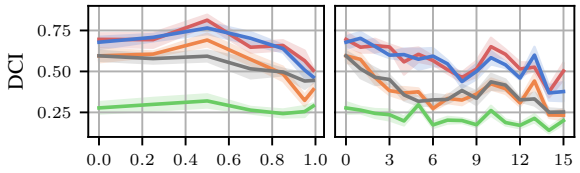
4.2 CORRELATED COMPONENTS

We now move to the common Shapes3D (Burgess and Kim, 2018) dataset. It shows geometric bodies that vary in their colors, shape, orientation, size, and background totaling six components. Compared to the previous section we train real encoders. We start our analysis where disentanglement learning is no longer possible: When components are correlated. Following Träuble et al. (2021), the dataset is resampled such that two components $z_i, z_j \in [0, 1]$ follow $z_i - z_j \sim \mathcal{N}(0, s^2)$. Lower s results in a stronger correlation where only few pairs of component values co-occur frequently. We choose a moderate correlation of $s = 0.4$ here and three pairs z_i, z_j that are nominal/nominal, nominal/ordinal, and ordinal/ordinal variables. We train four state-of-the-art disentanglement learning VAEs (BetaVAE (Higgins et al., 2017), FactorVAE (Kim and Mnih, 2018), BetaTCVAE (Chen et al., 2018), DipVAE (Kumar et al., 2018)) from a recent study (Locatello et al., 2019) and apply ICA, PCA, and our DMA and IMA discovery methods on their embedding spaces to post-hoc recover the original components. For DMA and IMA, we use the optimization-based algorithms (Eqn. 1) since they find approximate solutions through aggregation of many noisy sample gradients.

Tab. 2 shows the resulting DCI scores. In line with Träuble et al. (2021), we find that the disentanglement learning VAEs fail to recover the correlated components on their own due to their violated stochastic independence assumption (Fig. 4a). In eleven of the twelve model/correlation pairs, DMA or IMA identify better concepts than the VAE unit axes and the PCA/ICA components with improvements of

Correlated components	floor & background		orientation & background		orientation & size	
BetaVAE	0.497 ± 0.03		0.581 ± 0.04		0.491 ± 0.05	
+PCA	0.263 ± 0.03	-47%	0.310 ± 0.02	-47%	0.324 ± 0.04	-34%
+ICA	0.574 ± 0.04	+16%	0.540 ± 0.08	-7%	0.577 ± 0.04	+17%
+Ours (IMA)	0.617 ± 0.02	+24%	0.602 ± 0.05	+3%	0.579 ± 0.03	+18%
+Ours (DMA)	$\mathbf{0.641 \pm 0.03}$	**+29%**	$\mathbf{0.624 \pm 0.06}$	**+7%**	$\mathbf{0.627 \pm 0.03}$	**+28%**
FactorVAE	0.507 ± 0.11		0.502 ± 0.08		$\mathbf{0.712 \pm 0.01}$	
+PCA	0.358 ± 0.07	-29%	0.474 ± 0.05	-5%	0.556 ± 0.03	-22%
+ICA	0.294 ± 0.07	-42%	0.263 ± 0.05	-48%	0.340 ± 0.03	-52%
+Ours (IMA)	0.551 ± 0.04	+9%	0.498 ± 0.03	-1%	0.595 ± 0.05	-16%
+Ours (DMA)	$\mathbf{0.584 \pm 0.05}$	**+15%**	$\mathbf{0.510 \pm 0.05}$	**+2%**	0.556 ± 0.04	-22%
BetaTCVAE	0.619 ± 0.01		0.613 ± 0.04		0.659 ± 0.01	
+PCA	0.400 ± 0.03	-35%	0.421 ± 0.07	-31%	0.450 ± 0.07	-32%
+ICA	0.540 ± 0.02	-13%	0.497 ± 0.04	-19%	0.627 ± 0.02	-5%
+Ours (IMA)	0.623 ± 0.02	+1%	0.652 ± 0.03	+6%	0.638 ± 0.04	-3%
+Ours (DMA)	$\mathbf{0.666 \pm 0.01}$	**+8%**	$\mathbf{0.664 \pm 0.02}$	**+8%**	$\mathbf{0.748 \pm 0.03}$	**+14%**
DipVAE	0.631 ± 0.02		0.652 ± 0.02		0.548 ± 0.04	
+PCA	0.158 ± 0.01	-75%	0.160 ± 0.02	-75%	0.170 ± 0.02	-69%
+ICA	0.630 ± 0.02	-0%	0.651 ± 0.02	-0%	0.542 ± 0.03	-1%
+Ours (IMA)	0.644 ± 0.02	+2%	0.624 ± 0.01	-4%	0.558 ± 0.05	+2%
+Ours (DMA)	$\mathbf{0.684 \pm 0.01}$	**+8%**	$\mathbf{0.679 \pm 0.01}$	**+4%**	$\mathbf{0.601 \pm 0.05}$	**+10%**

Table 2: DMA recovers the components best in 11 out of 12 cases across different models and correlated components of Shapes3D. Mean $\pm$ std. err. of DCI across all components.



(a) Correlation strength (b) Number pairwise corr.

Figure 5: DMA and IMA recover the components even under strong and multiple correlations between them. ICA and PCA fail to return better components than the unit axes.

up to 29 %. This experiment shows that their concept discovery works regardless of (1) the model type and (2) the type of components correlated. On average, DMA delivers better results than IMA (+0.047), despite the generative process of Shapes3D only being roughly IMA or DMA-compliant. We therefore hypothesize that the DMA criterion might be more robustly optimizable in practice. Fig. 4b visualizes the performance achieved via DMA when traversing the embedding space. It also shows that small DCI differences can mean a significant improvement. This is because (1) the metric is computed across all six components and the strong baselines already identify many concepts and (2) a perfect score of 1.0 is usually not possible due to non-linearly encoded components. We investigate other correlation strengths with similar findings in App. D.3.

4.3 GAUSSIANITY AND MULTIPLE CORRELATIONS

In this section, we increase the distributional challenges to analyze whether our approaches are as distribution-agnostic

Method	$s = 0.1$	$s = 0.15$	$s = 0.2$	$s = \infty$
unit dirs.	0.238 ± 0.01	0.244 ± 0.01	0.247 ± 0.01	0.286 ± 0.02
PCA	0.238 ± 0.01	0.376 ± 0.03	0.373 ± 0.03	0.343 ± 0.03
ICA	0.409 ± 0.02	0.309 ± 0.02	0.311 ± 0.01	$\mathbf{0.652 \pm 0.00}$
(Ours) IMA	0.295 ± 0.01	0.302 ± 0.01	0.333 ± 0.04	0.266 ± 0.12
(Ours) DMA	$\mathbf{0.435 \pm 0.01}$	$\mathbf{0.411 \pm 0.03}$	$\mathbf{0.392 \pm 0.02}$	0.369 ± 0.05

Table 3: Without correlations ($s = \infty$), ICA is able to recover the components of a classification model. Under correlations, DMA works best. Mean $\pm$ std. err. of DCI.

as intended. We sample the components of Shapes3D from a (rotationally symmetric) Gaussian. Additionally, we introduce correlations between multiple components to its covariance matrix. Details on how covariance matrices are constructed are given in App. C.3.

First, we study a single pair of correlated components (floor and background color) with increasing correlation strength ρ. Fig. 5a shows that the BetaVAE handles low correlations well but starts deteriorating from a strength of $\rho > 0.5$, along with ICA. The DCI of our methods is an average constant of $+0.145$ above the BetaVAE's for $\rho \leq 0.85$. After this, it returns to the underlying BetaVAE's DCI, possibly because the two components collapsed in the BetaVAE's embedding space. For Fig. 5b, we gradually add more moderately correlated ($\rho \approx 0.7$) pairs to the Gaussian's covariance matrix until eventually all components are correlated. Again, our models show a constant benefit over the underlying BetaVAE's DCI curve. This experiment highlights that both DMA and IMA perform well with (1) strong and (2) multiple correlations and (3) Gaussian components.

4.4 DISCRIMINATIVE EMBEDDING SPACES

We highlight that our approach is also applicable to classification models that were trained in a purely discriminative manner, e.g., the feature space of a CNN model. To investigate this setting, we set up an 8-class classification problem on the Shapes3D dataset, where the combination of the four binarized components object color, wall color (blue/red vs. yellow/green), shape (cylinder vs. cube) and orientation (left vs. right) determines the class as visualized in App. C.4. To make the setting even more realistic, we artificially add labeling noise close to the decision boundary, correlations as in Sec. 4.2, and a small L2-regularizer on the embeddings to keep them in a reasonable range. We train a discriminative CNN with a K=6-dimensional embedding space.

The discriminative loss leads to a clustered distribution in the embedding space. ICA expectedly works very well in this highly non-Gaussian distribution, when no significant correlations are present which is in line with the result in Thm. 3.2. However, tables turn as we increasingly correlate the floor and background color: Starting at $s = 0.2$, DMA outperforms ICA and the other methods as can be seen in Tab. 3. While IMA leads to better concepts over the unit directions, it does not reach the level of DMA. We note that both ICA and our methods improve again for very strong correlations, where the setup approaches the case of three independent components (the other two components being treated as one) that is easier again. Overall, this demonstrates that our methods are applicable to purely discriminative embedding spaces and are more robust to high levels of correlations than ICA.

4.5 REAL-WORLD CONCEPT DISCOVERY

Last, we go beyond the traditional benchmarks and perform realistic concept discovery: We analyze the embedding space of a ResNet50 classifier (He et al., 2016) trained on the CUB-200-2011 (Wah et al., 2011) dataset consisting of high-resolution images of birds. This amplifies the challenges of the previous sections, i.e., a discriminative space, non-linear component dependencies of varying strengths across multiple components, and a large 512-dimensional embedding space. One restriction of this experiment is that CUB has no data-generating components to compare against, so we cannot report DCI scores. However, we qualitatively show that DMA can deliver interpretable concepts by matching them to annotated attributes of CUB.

We apply DMA and IMA to discover K=30 concepts of which the first two DMA concepts are shown exemplarily in Fig. 6. The images with the highest positive scores on the first component (on the right) consistently show white birds. The other end of the component comprises birds whose primary color is black. This gives a high Spearman rank correlation with the CUB attribute "primary color: white". The second concept is similarly interpretable. To quantify this across all K components, we provide an initial quantitative evaluation based on the Spearman rank correlation between components and attributes in App. D.7. It indicates that ICA and PCA have problems providing such components and the components identified by DMA usually correspond more closely to the attributes. The concepts provided by our method also compare favorably to those identified by ACE (Ghorbani et al., 2019) and ConceptSHAP (Yeh et al., 2019). While the construction of further quantitative evaluation schemes goes beyond the scope of this work, these promising results highlight that DMA also works for high-dimensional, real-world datasets.

5 DISCUSSION

We conclude by discussing the limitations of this work and related approaches and provide constructive guidance on which approach to choose in practice.

Limitations. In order to overcome distributional assumptions, our approach requires other forms of constraints. Most notably, we suppose that the generative processes comply

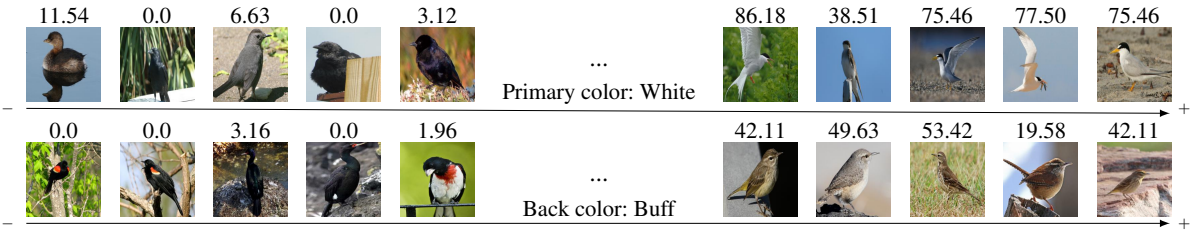


Figure 6: Components discovered by DMA on CUB correlate with interpretable ground truth attributes. Images are ordered by their concept scores $(\boldsymbol{Me})_i$, and the numbers show their ground truth annotated attribute score.

with the functional properties of Disjoint or Independent Mechanisms. While they are intuitive and our empirical results suggest that they are a useful approximation of real-world images, we acknowledge that these requirements are not strictly fulfilled in most practical scenarios and the quality of the results depends on the extent to which these constraints are violated. We investigate the robustness of our methods to violations of the assumptions in App. D.5. Compared to the classical methods such as PCA or IMA, the gradient-based optimization requires additional resources. However, the runtime strongly depends on hyperparameters such as the number of optimization steps. We also show that improved results can still be obtained time budget comparable to that of PCA and IMA in App. D.5.

Choosing the right approach for concept discovery. Overall, our results show that unsupervised conceptual explanations with guarantees are only possible under specific sets of working assumptions. In this paragraph, we would like to briefly summarize them and give constructive suggestions on which approaches are best used when.

- PCA works with uncorrelated components that are orthogonally encoded. We believe that the assumption of an orthogonal encoding is rather unlikely in practice, even if non-correlation was possible.
- ICA works well under independently distributed components but fails under dependent components. We suggest using this method when there is evidence that the ground truth components are independent.
- DMA does not require independence, but instead requires a disjoint mechanisms process and a faithful encoder to this process. This assumption is particularly suitable for image-generating processes.
- IMA does not require independence as well, but requires a faithful encoder to an independent mechanisms process. The class of independent mechanism processes is larger and may also cover non-image processes (Gresele et al., 2021). However, it requires the additional NEMR condition. We further empirically observed that the objective derived from IMA is harder to optimize for with SGD optimizers.
- Other approaches like ConceptSHAP (Yeh et al., 2019) and ACE (Ghorbani et al., 2019) also come with certain restrictions: ACE requires a model that is scale and shift invariant, while ConceptSHAP is specifically designed for computer vision models with spatial feature maps such as ResNet (He et al., 2016). Further, these approaches come without formal guarantees.

6 CONCLUSION

Summary. We proposed identifiability as a minimal requirement for concept discovery algorithms. Furthermore, we suggested the two functional paradigms of disjoint and independent mechanisms and proved that they can recover known components in visual embedding spaces. Extensive experiments confirmed that they offer substantial improvements on various generative and discriminative models and remain unaffected by distributional challenges.

Outlook. We believe our work to be a valuable step towards a rigorous formalization of concept discovery. However, the considered setup can be generalized in the future, for instance to components that are not linearly encoded. This would permit even stronger guarantees. While we have taken a technical perspective here, future work is required to investigate the effect of improved concepts on upstream explanations.

Acknowledgements

The authors thank Frederik Träuble, Luigi Gresele, and Julius von Kügelgen for insightful discussions during early development of this project. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Michael Kirchhof.

References

Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations

via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Arianna Bisazza and Clara Tump. The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2871–2876. Association for Computational Linguistics, 2018.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Rachel Carrington, Karthik Bharath, and Simon Preston. Invariance and identifiability issues for word embeddings. *Advances in Neural Information Processing Systems*, 32, 2019.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020b.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. In *Advances in Neural Information Processing Systems*, 2022.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32, pages 9277–9286, 2019.

Luigi Gresele, Julius von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. Independent mechanism analysis, a new concept? In *Advances in Neural Information Processing Systems*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9(7):1483–1492, 1997.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component Analysis*. John Wiley & Sons, Inc, 2001.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999. ISSN 0893-6080.

Aapo Hyvärinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

Ian T Jolliffe. *Principal component analysis*. Springer, 2nd edition, 2002.

Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (cme): concept-based model extraction. *AIMLAI workshop at the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2020.

Dmitry Kazhdan, Botty Dimanov, Helena Andres Terre, Mateja Jamnik, Pietro Liò, and Adrian Weller. Is disentanglement all you need? comparing concept-based & disentanglement approaches. *RAI, WeaSul, and RobustML workshops at The Ninth International Conference on Learning Representations 2021*, 2021.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Tobias Leemann, Yao Rong, Stefan Kraft, Enkelejda Kasneci, and Gjergji Kasneci. Coherence evaluation of visual concepts with objects and language. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

Gemma Elyse Moran, Dhanya Sridhar, Yixin Wang, and David Blei. Identifiable deep generative models via sparse decoding. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=vd0onGWZbE.

Lukas Muttenthaler, Charles Yang Zheng, Patrick McClure, Robert A. Vandermeulen, Martin N Hebart, and Francisco Pereira. VICE: Variational interpretable concept embeddings. In *Advances in Neural Information Processing Systems*, 2022.

Bjorn Ommer and Joachim M Buhmann. Learning the compositional nature of visual objects. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Matthew Painter, Adam Prugel-Bennett, and Jonathon Hare. Linear disentangled representations and unsupervised action estimation. In *Advances in Neural Information Processing Systems*, volume 33, pages 13297–13307. Curran Associates, Inc., 2020.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *International Conference on Learning Representations*, 2022.

Andrew Ross, Nina Chen, Elisa Zhao Hang, Elena L Glassman, and Finale Doshi-Velez. Evaluating the interpretability of generative models by interactive reconstruction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pages 10401–10412. PMLR, 2021.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, 2011.

Tao Yang, Xuanchi Ren, Yuwang Wang, Wenjun Zeng, and Nanning Zheng. Towards building a group-based unsupervised representation disentanglement framework. In *International Conference on Learning Representations*, 2021.

Chih-Kuan Yeh, Been Kim, Sercan O Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ICA with unconditional priors. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

When are Post-hoc Conceptual Explanations Identifiable? (Supplementary material)

Tobias Leemann[1,2,†] **Michael Kirchhof**[1,†] **Yao Rong**[1,2] **Enkelejda Kasneci**[2] **Gjergji Kasneci**[2]

[1]University of Tübingen, Tübingen, Germany
[2]Technical University of Munich, Munich, Germany
[†]equal contribution

A ADDITIONAL RELATED WORK

Orthogonality constraints and disentanglement for generative models. In the context of generative adversarial networks (GANs) [Goodfellow et al., 2014], the problem of analyzing and discovering interpretable directions has be studied recently by Voynov and Babenko [2020]. Ren et al. [2022] propose a contrastive approach to discover interpretable directions using pretrained generative models. Wei et al. [2021] have proposed an orthogonality regularization of the Jacobian, which resulted in more interpretable generative abilities. Ramesh et al. [2018] constrain the right-singular vectors of a generator Jacobian to be unit directions, which corresponds to column-wise orthogonal generator Jacobians. We go beyond these works by providing rigorous results on identifiability and by extending the scope to a encoder-only models.

B PROOFS

B.1 ROTATIONS DESTROY ORTHOGONALITY LEMMA

We start by first proving an auxiliary lemma. We show that orthogonality of Jacobians, i.e., $\boldsymbol{J}_f \boldsymbol{J}_f^\top = \boldsymbol{S}$ with a diagonal matrix $\boldsymbol{S}$ will be destroyed in the general case when a rotation $\boldsymbol{R}$ is applied, such that $\boldsymbol{J}_{Rf} \boldsymbol{J}_{Rf}^\top = \boldsymbol{R}\boldsymbol{J}_f \boldsymbol{J}_f^\top \boldsymbol{R}^\top = \boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top$ is not a diagonal matrix anymore.

Lemma B.1 (Rotations destroy orthogonality patterns.) *Let $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ be a diagonal matrix, $\boldsymbol{S} = \operatorname{diag}(\boldsymbol{s})$ with diagonal entries $\boldsymbol{s} > 0$ and $s_i \neq s_j, \forall i \neq j$, i.e., all diagonal entries of $\boldsymbol{S}$ are different and positive. Let $\boldsymbol{R} \in \mathbb{R}^{K\times K}$ be any rotation matrix with $\boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}$. If $\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top$ is a diagonal matrix, $\boldsymbol{R}$ must a signed permutation matrix (a permutation matrix where entries can be ± 1).*

Proof. With $\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top = \operatorname{diag}(\lambda_1, \dots \lambda_K)$, we have for each unit vector $\boldsymbol{e}^{(i)}$, $i = 1, \dots, K$, that

$$\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top \boldsymbol{e}^{(i)} = \lambda_i \boldsymbol{e}^{(i)} \ . \tag{2}$$

We can represent $\boldsymbol{R}$ by its rows, $\boldsymbol{R} = [\boldsymbol{r}_1, \dots, \boldsymbol{r}_K]^\top$ where each $\boldsymbol{r}_i \in \mathbb{R}^K$. In this notation, $\boldsymbol{R}^T \boldsymbol{e}^{(i)} = \boldsymbol{r}_i$, i.e., multiplication of the transpose with a unit vector will select the row $\boldsymbol{r}_i$. This results in

$$\boldsymbol{R}\boldsymbol{S}\boldsymbol{r}_i = \lambda_i \boldsymbol{e}^{(i)} \tag{3}$$

Because $\boldsymbol{R}$ is invertible and square, we can left-multiply the equation by $\boldsymbol{R}^\top$. Using $\boldsymbol{R}^T \boldsymbol{e}^{(i)} = \boldsymbol{r}_i$ again, we arrive at

$$\boldsymbol{S}\boldsymbol{r}_i = \lambda_i \boldsymbol{r}_i. \tag{4}$$

This implies that all $\boldsymbol{r}_i$ are eigenvectors of the matrix $\boldsymbol{S}$ with the eigenvalues λ_i. By the initial assumption, $\boldsymbol{S}$ is a diagonal matrix with all-different entries s_i. The eigenvectors of such a matrix are only scaled unit vectors $\boldsymbol{e}^{(j)}$. Thus, each $\boldsymbol{r}_i$ will be

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

a scaled unit-vector. The constraint of $\boldsymbol{R}$ being an orthogonal matrix enforces the $\boldsymbol{r}_i$ to be mutually different unit vectors with length 1. Therefore, $\boldsymbol{R}$ necessarily has the form of a signed permutation. □

Note that the converse is also true. If $\boldsymbol{R}$ is a signed permutation matrix, $\boldsymbol{RSR}^\top$ will be diagonal.

B.2 PCA ENSURES IDENTIFIABILITY (THEOREM 3.1)

Theorem B.1 (PCA identifiability, Theorem 3.1) *Let $z_k, k = 1, \dots, K$, be uncorrelated random variables with non-zero and unequal variancecs. Let $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ is an orthonormal matrix. If an orthonormal post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{K\times K}$ results in mutually uncorrelated components $(z'_1, \dots, z'_K) = \boldsymbol{z}' = \boldsymbol{Me}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a matrix where $|s_{ii}| = 1$ for $i \in 1, \dots K$.*

Proof. Since both $\boldsymbol{M}$ and $\boldsymbol{D}$ are orthogonal, $\boldsymbol{MD} = \boldsymbol{Q}$ is also orthogonal. Our post-hoc transformation resulted in uncorrelated components, i.e., $\mathrm{Cov}(\boldsymbol{Qx}) = \boldsymbol{Q}\,\mathrm{Cov}(\boldsymbol{x})\boldsymbol{Q}^\top\boldsymbol{\Gamma}$ is diagonal, where $\boldsymbol{\Gamma}$ is some diagonal matrix. Thus, $\boldsymbol{Q}\,\mathrm{Cov}(\boldsymbol{x})\boldsymbol{Q}^\top$ is diagonal, too. We also know that our original components are uncorrelated with unequal variances, i.e., $\mathrm{Cov}(\boldsymbol{x}) = \mathrm{diag}(\boldsymbol{s})$ with $\boldsymbol{s} > 0$ and $s_i \neq s_j, \forall i \neq j$. Our helper Lemma B.1 then implies that $\boldsymbol{Q}$ must be a signed permutation. Thus, $\boldsymbol{z}' := \boldsymbol{Me} = \boldsymbol{MDz} = \boldsymbol{Qz} =: \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a matrix where $|s_{ii}| = 1$ for $i \in 1, \dots K$. □

B.3 ICA ENSURES IDENTIFIABILITY (THEOREM 3.2)

Theorem B.2 (ICA identifiability, Theorem 3.2) *Let $z_k, k = 1, \dots, K$, be independent random variables with non-zero variances where at most one component is Gaussian. Let $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ has full rank. If a post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ results in mutually independent components $(z'_1, \dots, z'_K) = \boldsymbol{z}' = \boldsymbol{Me}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a scaling matrix.*

Proof. (1) We know that $\boldsymbol{z}' = MDz =: \boldsymbol{C}'z$. Let us start with an additional assumption that both $\boldsymbol{z}'$ and $\boldsymbol{z}$ have unit variances. Then, by Comon [1994, App. A .1], $\boldsymbol{C}'$ must be orthonormal.

Let us recall the following result

Theorem B.3 (Theorem 11 from Comon [1994]) *Let $\boldsymbol{x}$ be a vector with independent components, of which at most one is Gaussian, and whose densities are not reduced to a point-like mass. Let $\boldsymbol{C}$ be an orthogonal $K \times K$ matrix and $\boldsymbol{z}$ the vector $\boldsymbol{z} = \boldsymbol{Cx}$. Then the following three properties are equivalent:*

1. *The components z_i are pairwise independent.*
2. *The components z_i are mutually independent.*
3. *$\boldsymbol{C} = \boldsymbol{SP}$ where $\boldsymbol{S}$ is diagonal, $\boldsymbol{P}$ is a permutation.*

Since $\boldsymbol{z}$ fulfills the conditions of this this theorem and $\boldsymbol{z}'$ has mutually independent entries, we know that $\boldsymbol{C}' = \boldsymbol{SP}$.

(2) We now allow arbitrary variances, i.e., $\mathrm{Cov}(\boldsymbol{z}') = \boldsymbol{\Lambda}$ and $\mathrm{Cov}(\boldsymbol{z}) = \boldsymbol{\Gamma}$ where both covariance matrices are positive diagonal matrices. $\boldsymbol{z}' = \boldsymbol{MDz} = \boldsymbol{C}'\boldsymbol{z} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{C}'\boldsymbol{\Gamma}^{1/2}\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z} =: \boldsymbol{\Lambda}^{1/2}\boldsymbol{C}''\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z}$. This is equivalent to $(\boldsymbol{\Lambda}^{-1/2}\boldsymbol{z}') = \boldsymbol{C}''(\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z})$. These rescaled random vectors both have unit variances, so (1) implies that $\boldsymbol{C}'' = \boldsymbol{S'P'}$. We can plug this back into the previous equation and see that $\boldsymbol{z}' = \boldsymbol{\Lambda}^{1/2}\boldsymbol{C}''\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{S'P'}\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z} =: \boldsymbol{P'S''z}$. Thus, $\boldsymbol{z}' = \boldsymbol{Me} = \boldsymbol{MDz} = \boldsymbol{P'S''z}$, where $\boldsymbol{P}' \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S''} \in \mathbb{R}^{K\times K}$ is a scaling matrix. □

B.4 TRANSFER LEMMA

DMA and IMA are based on structures in the Jacobian of the generative process. To be able to use them in the encoder and ultimately discover concepts, we first show that if an encoder mirrors the behavior of the generative process, up to a rotation and scale, its Jacobians must also mirror the Jacobians of the generative process.

Lemma B.2 (Transfer lemma) *Let $\boldsymbol{f}$ be a faithful encoder for the generative process $\boldsymbol{g}$ and further $\boldsymbol{f} \circ \boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{PSz}$ $\forall \boldsymbol{z} \in \mathcal{Z}$ where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diagonal matrix. Then $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{P'S'J_g}(\boldsymbol{z})^\top$ where $\boldsymbol{P'} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S'} \in \mathbb{R}^{K\times K}$ is a diagonal matrix.*

Proof. Let $\boldsymbol{z} \in \mathcal{Z}$ be arbitrary. $(\boldsymbol{f} \circ \boldsymbol{g})(\boldsymbol{z}) = \boldsymbol{PSz}$ implies $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J_g}(\boldsymbol{z}) = \boldsymbol{PS}$. Since $\boldsymbol{f}$ is faithful to $\boldsymbol{g}$, $\boldsymbol{S}$ has full rank, i.e., $\boldsymbol{S} = \text{diag}(\alpha_1, \ldots, \alpha_K)$ with $\alpha_k \in \mathbb{R}_{\neq 0}, k = 1, \ldots, K$.

Now, let us write $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})) = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^\top$ with $\boldsymbol{v}_i \in \mathbb{R}^L$. Similarly, we can write $\boldsymbol{J_g}(\boldsymbol{z}) = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ with $\boldsymbol{w}_i \in \mathbb{R}^L, i = 1, \ldots, K$.

Let us focus on an individual row of $\boldsymbol{J_f}$, i.e., let $k \in \{1, \ldots, K\}$ be a fixed index of a row. Since $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J_g}(\boldsymbol{z}) = \boldsymbol{PS}$ and $\boldsymbol{P}$ is a permutation matrix with exactly one 1 per row, there is precisely one column index k' such that the k-th row and k'-th column of $\boldsymbol{PS}$ is non-zero. This setup allows drawing certain conclusions about the vector $\boldsymbol{v}_k$. Let $j = 1, \ldots, K$ denote an arbitrary column of $\boldsymbol{PS}$. Then,

(i) if $j = k'$, then $\boldsymbol{v}_k^\top \boldsymbol{w}_{k'} = \alpha_{k'} \neq 0$. In consequence, $\boldsymbol{v}_k \neq 0$, $\boldsymbol{w}_{k'} \neq 0$ and so we can decompose $\boldsymbol{v}_k = \boldsymbol{a}_k + \boldsymbol{b}_k$, where $\boldsymbol{a}_k \in \text{span}(\{\boldsymbol{w}_{k'}\}) \setminus \{0\}$ and $\boldsymbol{b}_k \in \text{span}(\{\boldsymbol{w}_{k'}\})^\perp$, where $^\perp$ denotes the orthogonal complement. Because $\text{span}(\{\boldsymbol{w}_{k'}\}) = \{\mu \boldsymbol{w}_{k'} | \mu \in \mathbb{R}\}$, we know that $\boldsymbol{a}_k = \frac{\alpha_{k'}}{\|\boldsymbol{w}_{k'}\|_2^2} \boldsymbol{w}_{k'}$.

(ii) if $j \neq k'$, then $\boldsymbol{v}_k^\top \boldsymbol{w}_j = 0$. With (i), it follows that $\boldsymbol{b}_k \in \text{span}\left(\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K\}\right)^\perp = \text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp$.

Since $\boldsymbol{f}$ is faithful to $\boldsymbol{g}$, we know that for each $\mathbf{c} \in \text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp$, $\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{z}))\mathbf{c} = \mathbf{0}$ ant therefore $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{b}_k = \mathbf{0}$ This demands that the k-th component of the product is also 0, i.e., $\mathbf{v}_k \boldsymbol{b}_k = (\boldsymbol{a}_k + \boldsymbol{b}_k)^\top \boldsymbol{b}_k = \boldsymbol{a}_k^\top \boldsymbol{b}_k + \boldsymbol{b}_k^\top \boldsymbol{b}_k = 0$. By design $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$ are orthogonal such that immediately follows $\boldsymbol{b}_k = \mathbf{0}$ Hence, $\boldsymbol{v}_k = \boldsymbol{a}_k + \mathbf{0} = \frac{\alpha_{k'}}{\|\boldsymbol{w}_{k'}\|_2^2} \boldsymbol{w}_{k'} + \mathbf{0}$ for our selected row k. Globally, this means $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{P'S'J_g}(\boldsymbol{z})^\top$, with some scaling matrix $\boldsymbol{S'}$ and permutation matrix $\boldsymbol{P'}$. □

B.5 DISJOINT MECHANISMS ENSURE IDENTIFIABILITY (THEOREM 3.3)

Theorem B.4 (Identifiability under DMA, Theorem 3.3) *Let $\boldsymbol{g}$ have disjoint mechanisms and $\boldsymbol{f}$ be a faithful encoder to $\boldsymbol{g}$. If a full-rank post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ results in disjoint rows in the Jacobian $\boldsymbol{MJ_f}(\boldsymbol{g}(\boldsymbol{z}))$ for some $\boldsymbol{z} \in \mathcal{Z}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K \times K}$ is a scaling matrix.*

Proof. We know that $\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{D}$ and $\boldsymbol{D}$ has full rank. Since $\boldsymbol{M}$ also has full rank, there exists a non-singular matrix $\boldsymbol{E'}$ such that $\boldsymbol{M} = \boldsymbol{E'D}^{-1}$. We can rewrite $\boldsymbol{E'} = \boldsymbol{SE}$, where $\boldsymbol{E}$ has normalized rows and $\boldsymbol{S}$ is a diagonal matrix.

Since $\boldsymbol{D}^{-1}\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{I}$ and $\boldsymbol{g}$ is DMA, we can apply the transfer lemma (Lemma B.2). It implies that $\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z))$ has orthogonal rows.

Suppose now for contradiction that $\boldsymbol{E}$ was not a permutation matrix. This means that without loss of generality the first row must contain at least two columns whose entries are not equal to zero. Since $\boldsymbol{E}$ has full rank, there must be a second row with a non-zero entry in at least one of these columns. Since $\boldsymbol{D}^{-1}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}_a))$ has disjoint rows, $\boldsymbol{SED}^{-1}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}_a)) = \boldsymbol{MJ_f}(\boldsymbol{g}(\boldsymbol{z}_a))$ can no longer have disjoint rows. This contradicts the assumption. Hence, $\boldsymbol{E}$ must be a permutation matrix $\boldsymbol{P}$. This give $\boldsymbol{z'} = \boldsymbol{Me} = \boldsymbol{PSD}^{-1}\boldsymbol{Dz} = \boldsymbol{PSz}$. □

B.6 INDEPENDENT MECHANISMS ENSURE IDENTIFIABILITY (THEOREM 3.4)

Theorem B.5 (Identifiability under IMA, Theorem 3.4) *Let $\boldsymbol{g}$ adhere to IMA. Let $\boldsymbol{f}$ be a faithful encoder to $\boldsymbol{g}$. Suppose we have obtained an $\boldsymbol{f'} = \boldsymbol{Mf}$ with a full-rank $\boldsymbol{M} \in \mathbb{R}^{K \times K}$ and orthogonal rows in its Jacobian $\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))$, i.e, $\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))^\top = \boldsymbol{\Sigma}(\boldsymbol{z})$ where $\boldsymbol{\Sigma}(\boldsymbol{z})$ is diagonal. If additionally for two points $\boldsymbol{z}_a, \boldsymbol{z}_b \in \mathcal{Z}$ and $\gamma_i \coloneqq \frac{\Sigma_{ii}(\boldsymbol{z}_b)}{\Sigma_{ii}(\boldsymbol{z}_a)}$ and $\forall i, j = 1 \ldots K, i \neq j : \gamma_i \neq \gamma_j$ (NEMR condition), then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K \times K}$ is a scaling matrix.*

Proof. We know that $\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{D}$ and $\boldsymbol{D}$ has full rank. Since $\boldsymbol{M}$ also has full rank, there exists a non-singular matrix $\boldsymbol{E}$ such that $\boldsymbol{M} = \boldsymbol{ED}^{-1}$. We will now show that the solution set of $\boldsymbol{E}$ can be constrained to be a permutation and scaling operation in three steps.

(1) $\boldsymbol{J_{f'}}$ is orthogonal, i.e., $\boldsymbol{\Sigma}(\boldsymbol{z}_a) = (\boldsymbol{MJ}_f(\boldsymbol{g}(\boldsymbol{z}_a)))(\boldsymbol{MJ}_f(\boldsymbol{g}(\boldsymbol{z}_a))^\top = (\boldsymbol{ED}^{-1}\boldsymbol{J}_f(g(z_a)))(\boldsymbol{ED}^{-1}\boldsymbol{J}_f(g(z_a)))^\top = \boldsymbol{E}(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))^\top\boldsymbol{E}^\top$. Since $\boldsymbol{D}^{-1}\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{I}$ and $\boldsymbol{g}$ is DMA, we can apply the transfer lemma (Lemma B.2) and know that $\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a))$ must have orthogonal rows, i.e., $(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))^\top = \boldsymbol{\Gamma}_a$, where $\boldsymbol{\Gamma}_a$ is some diagonal matrix with full rank. Substituting this back into the previous term, $\boldsymbol{\Sigma}(\boldsymbol{z}_a) = \boldsymbol{E}\boldsymbol{\Gamma}_a\boldsymbol{E}^\top$. The same holds for $\boldsymbol{z}_b$, i.e., $\boldsymbol{\Sigma}(\boldsymbol{z}_b) = \boldsymbol{E}\boldsymbol{\Gamma}_b\boldsymbol{E}^\top$.

(2) We've seen in (1) that both $\boldsymbol{\Sigma}(\boldsymbol{z}_a)$ and $\boldsymbol{\Gamma}_a$ are the results of quadratic forms. Hence, their entries are all positive, and strictly positive because they have full rank. Thus we can define $\boldsymbol{Q} := \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{E}\boldsymbol{\Gamma}_a^{1/2}$. Due to (1), $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}$, i.e., $\boldsymbol{Q}$ is orthogonal. It is easy to see that $\boldsymbol{E} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{1/2}$. In other words, $\boldsymbol{E}$ must be a (twice) scaled orthogonal matrix.

(3) From (1) we get that

$$\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1} = \boldsymbol{E}\boldsymbol{\Gamma}_a\boldsymbol{E}^\top(\boldsymbol{E}\boldsymbol{\Gamma}_b\boldsymbol{E}^\top)^{-1} \tag{5}$$

$$\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1} = \boldsymbol{E}\boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1}\boldsymbol{E}^{-1} \tag{6}$$

$$\boldsymbol{E}^{-1}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{E} = \boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1} \tag{7}$$

Now we can insert the result from (2)

$$\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{Q}^\top\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{1/2}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{1/2} = \boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1} \tag{8}$$

$$\boldsymbol{Q}^\top\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{1/2}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q} = \boldsymbol{\Gamma}_a^{1/2}\boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1}\boldsymbol{\Gamma}_a^{-1/2} \tag{9}$$

$$\boldsymbol{Q}^\top\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{Q} = \boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1} \tag{10}$$

$$\tag{11}$$

Due to the NEMR condition, $\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}$ is a diagonal matrix with unequal positive entries. We can thus apply Lemma B.1 which implies that $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{S}$ where $\boldsymbol{P}$ is a permutation and $\boldsymbol{S}$ a diagonal matrix. Inserting this back into (2) gives $\boldsymbol{E} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{1/2} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{P}\boldsymbol{S}\boldsymbol{\Gamma}_a^{1/2} = \boldsymbol{P}\boldsymbol{S}'$, where $\boldsymbol{S}'$ is a diagonal matrix. Hence, $\boldsymbol{z}' = \boldsymbol{M}\boldsymbol{e} = \boldsymbol{P}\boldsymbol{S}'\boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{z} = \boldsymbol{P}\boldsymbol{S}'\boldsymbol{z}$. □

In the next section, we discuss how the proofs can be turned into analytical solutions to discover the ground truth components.

B.7 ANALYTICAL SOLUTIONS TO CONCEPT DISCOVERY

B.7.1 Disjoint Mechanisms

Under a perfect DMA process $\boldsymbol{g}$ and a noiseless faithful encoder $\boldsymbol{f}$ to $\boldsymbol{g}$, we can compute an analytical solution for $\boldsymbol{M}$ that will result in an encoder $\boldsymbol{f}' = \boldsymbol{M}\boldsymbol{f}$ that is compliant with the *DMA criterion*, i.e., disjoint rows in its Jacobian. Suppose we are provided with a gradient matrix of $\boldsymbol{f}$, $\boldsymbol{J}_f(\boldsymbol{x}_a) \in \mathbb{R}^{K\times L}$. We propose the following steps:

1. Select a submatrix $\boldsymbol{J}_{reg} \in \mathbb{R}^{K\times K}$ of K linearly independent columns in $\boldsymbol{J}_f(\boldsymbol{x}_a)$, such that $\det(\boldsymbol{J}_{reg}) \neq 0$.
2. Compute and return $\boldsymbol{M} = \boldsymbol{J}_{reg}^{-1}$
3. This will result in $\boldsymbol{f}' = \boldsymbol{M}\boldsymbol{f}$ having disjoint rows in its Jacobian.

Proof. $\boldsymbol{J}_f(\boldsymbol{x}_a)$ must be of the form $\boldsymbol{J}_f(\boldsymbol{x}_a) = \boldsymbol{H}^{-1}\boldsymbol{J}_{f^*}(\boldsymbol{x}_a)$ for such an $\boldsymbol{M}$ to exist, where $\boldsymbol{J}_{f^*}$ is the Jacobian of an encoder $\boldsymbol{f}^*$ with disjoint rows and $\boldsymbol{H}$ has full rank. $\boldsymbol{J}_{reg}$ can be written as $\boldsymbol{J}_{reg} = \boldsymbol{H}^{-1}\boldsymbol{J}_{f^*,reg}$, where $\boldsymbol{J}_{f^*,reg}$ is a square submatrix of $\boldsymbol{J}_{f^*}$ with the same selected selected columns. The submatrix $\boldsymbol{J}_{f^*,reg}$ also will be of to be of full rank because it can be written as $\boldsymbol{H}\boldsymbol{J}_{reg}$, which are both full rank. Because of the DMA principle, $\boldsymbol{J}_{f^*,reg}$ again needs to be of the form $\boldsymbol{P}\boldsymbol{S}$ with one component active in each column. Furthermore, $\boldsymbol{M} = \boldsymbol{J}_{reg}^{-1} = (\boldsymbol{H}^{-1}\boldsymbol{P}\boldsymbol{S})^{-1} = \boldsymbol{S}^{-1}\boldsymbol{P}^{-1}\boldsymbol{H}$. As the inverses of scaling and permutation matrices have the same respective form again, $\boldsymbol{M}\boldsymbol{H}^{-1} = \boldsymbol{S}'\boldsymbol{P}'$. Therefore, $\boldsymbol{f}' = \boldsymbol{S}'\boldsymbol{P}'\boldsymbol{f}^*$, maintaining its disjoint Jacobians.

B.7.2 Independent Mechanisms

Suppose we are given matrices $\boldsymbol{\Sigma}(\boldsymbol{z}_a) = \boldsymbol{J}_f(\boldsymbol{x}_a)\boldsymbol{J}_f(\boldsymbol{x}_a)^\top = \boldsymbol{D}^{-1}\boldsymbol{\Gamma}_a(\boldsymbol{D}^{-1})^\top$ and $\boldsymbol{\Sigma}(\boldsymbol{z}_b) = \boldsymbol{J}_f(\boldsymbol{x}_b)\boldsymbol{J}_f(\boldsymbol{x}_b)^\top$. We then apply the following steps

1. $\boldsymbol{U} = \text{inverse}(\text{cholesky}(\boldsymbol{\Sigma}(\boldsymbol{z}_a))$
2. $\boldsymbol{V} = \text{eigenvectors}(\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_b)\boldsymbol{U}^\top)$
3. return $\boldsymbol{H} = \boldsymbol{V}^\top\boldsymbol{U}$

Algorithm 1: DMA concept discovery with SGD.

Input: encoder $\boldsymbol{f}$, images $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$
Jacobians $\leftarrow$ Gradient($\boldsymbol{f}$, $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$).detach()
$M \leftarrow K$-dim identity matrix
for L *epochs,* $\boldsymbol{J_f}(\boldsymbol{x}) \in$ *Jacobians* **do**
  $\boldsymbol{U} \leftarrow |\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{x})|$ `// No absolute value operation here for IMA`
  $\boldsymbol{U} \leftarrow$ row-normalize $\boldsymbol{U}$
  loss $\leftarrow ||\boldsymbol{U}\,\boldsymbol{U}^\top - \boldsymbol{I}_K||_F$
  loss.backwards() `// Optimize` M
end
return M

Algorithm 2: DMA concept discovery with SGD (determinant loss).

Input: encoder $\boldsymbol{f}$, images $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$
Jacobians $\leftarrow$ Gradient($\boldsymbol{f}$, $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$).detach()
$M \leftarrow K$-dim identity matrix
for L *epochs,* $\boldsymbol{J_f}(\boldsymbol{x}) \in$ *Jacobians* **do**
  $\boldsymbol{U} \leftarrow |\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{x})|$ `// No absolute value operation here for IMA`
  $\boldsymbol{V} \leftarrow \boldsymbol{U}\boldsymbol{U}^\top$
  loss $\leftarrow \log\left(\prod_i V_{ii}\right) - \log\det\left(\boldsymbol{V}\right)$
  loss.backwards() `// Optimize` M
end
return M

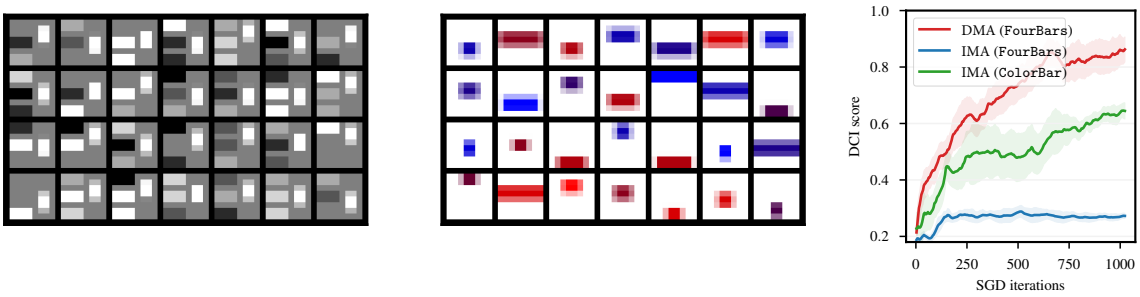
The first step implies that $\boldsymbol{U}^{-1}\boldsymbol{U}^{-\top} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)$ and that $\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{U}^\top = \boldsymbol{I}$. We have thus identified the matrix $\boldsymbol{E}$ from step (2) of the identifiability proof, which has the form $\boldsymbol{U} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{M}$. In step two we compute $\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_b)\boldsymbol{U}^\top = \boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{\Gamma}_b\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{Q}^\top\boldsymbol{\Lambda}^{1/2} = \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^\top$, where $\boldsymbol{R}$ holds the eigenvalues. Accordingly, by left and right multiplying with $\boldsymbol{V}$, we observe that $(\boldsymbol{V}^\top\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{z}_b)(\boldsymbol{V}^\top\boldsymbol{U})^\top = \boldsymbol{R}$, i.e., $(\boldsymbol{V}^\top\boldsymbol{U})$ solves the orthogonality problem for $\boldsymbol{\Sigma}(\boldsymbol{z}_b)$. We can easily verify that $\boldsymbol{H} = \boldsymbol{V}^\top\boldsymbol{U}$ is also a solution for $\boldsymbol{\Sigma}(\boldsymbol{z}_a)$ by computing $\boldsymbol{V}^\top\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{U}^\top\boldsymbol{V} = \boldsymbol{I}$. By the identifiability result, $\boldsymbol{H} = \boldsymbol{V}^\top\boldsymbol{U} = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{M}$, a scaled and permuted version of $\boldsymbol{D}^{-1}$, if the additional gradient ratio condition is fulfilled with $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$.

B.8 ALGORITHMS

We present the SGD optimization for DMA in Algorithm 1. Note that the algorithm for IMA optimization via SGD can be obtained by just omitting the absolute value operation in the line indicated by the comment. For the smaller toy datasets, we experiment with a version of the algorithm that uses the determinant (see Algorithm 2), similar to the objective put forward by Gresele et al. [2021]. As the determinant operation is hard to backpropagate through and might be unstable, we recommend Algorithm 1 for real-world applications and observed no significant performance differences on the datasets studied in this work.

B.9 EXTENDING GRADIENTS TO GENERAL ATTRIBUTIONS

We make an initial attempt to generalize our method, considering gradients as a simple form of attribution method. Intuitively, $\boldsymbol{J_f} = \nabla_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}))$ contains input gradients (termed grad in the remainder) which can be thought of as a simple form of attribution for each component [Simonyan et al., 2013, Shah et al., 2021]. Thus, on a more general level, our proposed approach optimizes for the disjointness of attributions. Thus, we may use other forms of *homogeneous attributions* in place of $\boldsymbol{J_f}$. These are local attribution methods $A_{\boldsymbol{f}} : \mathbb{R}^L \to \mathbb{R}^{K\times L}$ for the encoder $\boldsymbol{f}$ with $A_{\boldsymbol{M}\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{M}A_{\boldsymbol{f}}(\boldsymbol{x})$ that map an instance $\boldsymbol{x}$ to a matrix of attributions for each latent dimension. Besides the above input gradients, this class contains other popular methods such as integrated gradients (IG) [Sundararajan et al., 2017] and smoothed gradients (SG) [Smilkov et al.,



(a) Random samples in the `FourBars` dataset. (b) Random samples in the `ColorBar` dataset. (c) Disentangling gradients of synthetic datasets with SGD.

Figure 7: Random samples drawn from the synthetic datasets (a,b). On the `FourBars` dataset, IMA fails to iterate towards a disentangled solution, because the non-equal magnitudes condition is violated. However, IMA converges on the `ColorBar` dataset, although at a slower rate (c)

2017] (because these methods are linear in $\boldsymbol{f}$). Thus, we can formulate a generalized *disjoint attributions objective*:

$$\min_{\boldsymbol{M}} \sum_{n=1}^{N} \left|\left| \left|\overline{\boldsymbol{M}A_f(x)}\right| \left|\overline{\boldsymbol{M}A_f(x)}\right|^\top - I_K \right|\right|_F^2 . \tag{12}$$

We indicate the row-normalization operation by the overbar, and denote by $|\cdot|$ the element-wise absolute values operation. Without the absolute value operation this results in the *independent attributions objective*.

C EXPERIMENTAL DETAILS

We report the most important implementation details for our experiments in this section. Please confer the actual implementation available online[1] for full information.

C.1 SYNTHETIC DATASETS

We show random samples from both datasets in Figure 7. We provide an additional graphics with the behavior on the synthetic datasets in Figure 7c. They show that SGD exhibits a convergence behavior as predicted by our theory and comparable to the analytical solutions (shown in the main paper).

C.2 ARCHITECTURES

For the disentanglement models, we use the implementations provided by the open source library `disentanglement-pytorch`[2]. For the evaluation measures, we use the implementation of `disentanglement_lib`[3] with their respective default parameters. We use a simple encoder and decoder architecture, that consists of five and six feed-forward convolutional layers respectively and relies on the ReLU activation function.

C.3 CORRELATED SAMPLING

In this paper, we use two methods to introduce correlations between the ground truth components. Both methods rely on proportional resampling: We first draw a batch that has multiple times the final batch size (we use factors from 3-6

[1] https://github.com/tleemann/identifiable_concepts
[2] https://github.com/amir-abdi/disentanglement-pytorch
[3] https://github.com/google-research/disentanglement_lib

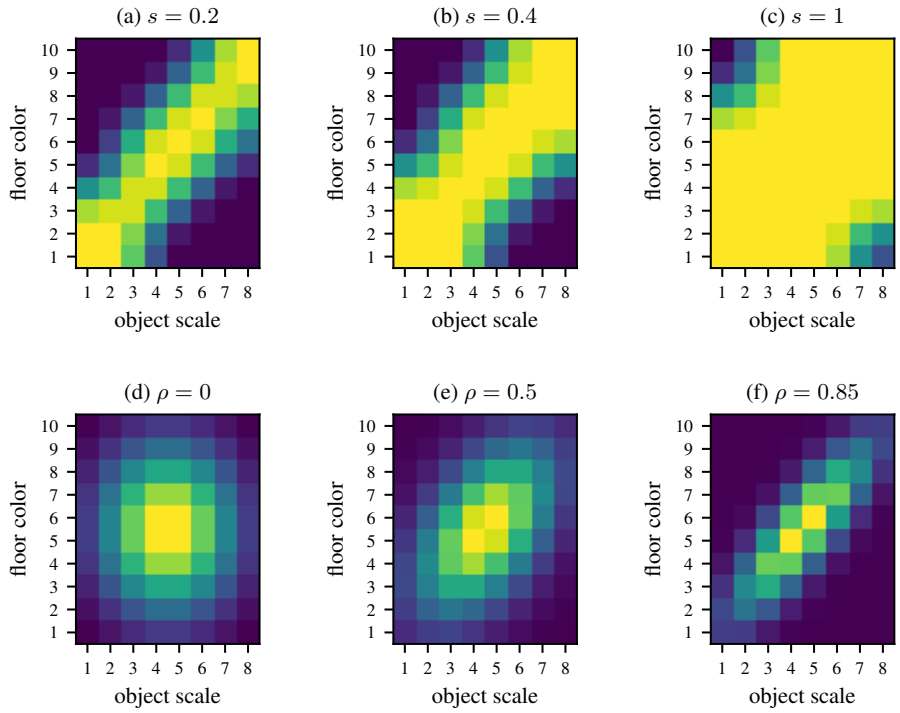


Figure 8: Exemplary correlated densities of the components floor color and object scale under the correlated sampling setup of Gresele et al. [2021] (a – c) and with our Gaussian sampling (d – f). The correlation strength is indicated on top. Purple denotes a low and yellow a high density.

depending on the non-uniformity of the distribution), then compute the (non-normalized) probability of each sample under a given distribution over the component values, and then resample a final batch (with replacement) proportional to these probabilities.

The two methods differ in the probability distribution assigned to the component values. The first setting (used in Sec. 4.2) uses the approach of Träuble et al. [2021]: As visualized in Fig. 8(a) to (c), we pick two components z_1 and z_2, create the grid of possible values, and then lay a diagonal line over this grid. Along this line, we set a normal distribution with a standard deviation s. A higher s means that the distribution gives a higher probability to more component combinations of the grid, whereas a smaller s is more restrictive. Mathematically, it is defined by Träuble et al. [2021] as:

$$p(z_1, z_2) \propto \exp\left(-\frac{(z_1 - \alpha z_2)^2}{2s^2}\right), \tag{13}$$

where $\alpha = z_1^{\max}/z_2^{\max}$ brings the components to a same scale and s is similarly normalized to the maximum values that z_1 and z_2 can take. The remaining components $z_i, i > 2$, are marginalized out of this distribution and thus continue to be sampled uniformly at random.

This setting is limited to one pair of components and also introduces a non-Gaussian distribution over all components. To tackle these limitations and thus to make the distributional challenge harder, we use a different probability distribution in Sec. 4.3. Here, we lay a normal distribution over *all* components, i.e., $z \sim \mathcal{N}(\mu, \Sigma)$, where μ is centered in the middle of the possible values, i.e., $\mu = \frac{z^{\max} - z^{\min}}{2}$. Σ is similarly normalized, since we decompose it into $\Sigma = \text{diag}(\sigma^2)\Gamma$. The vector $\sigma \in \mathbb{R}^K_{>0}$ gives standard deviations for each component via $\sigma^2 = \left(\frac{\mu+0.5}{2}\right)^2$ such that the distribution stretches across the grid of possible values. Note that the $+0.5$ is because the values are assumed to be zero-indexed. Γ is a correlation matrix with 1 on its diagonal. In the first experiment in Sec. 4.3, we correlate only one pair of variables and set their corresponding off-diagonal entries in Γ to ρ. Fig. 8 (d) to (f) show the corresponding marginal distributions of these components. In the

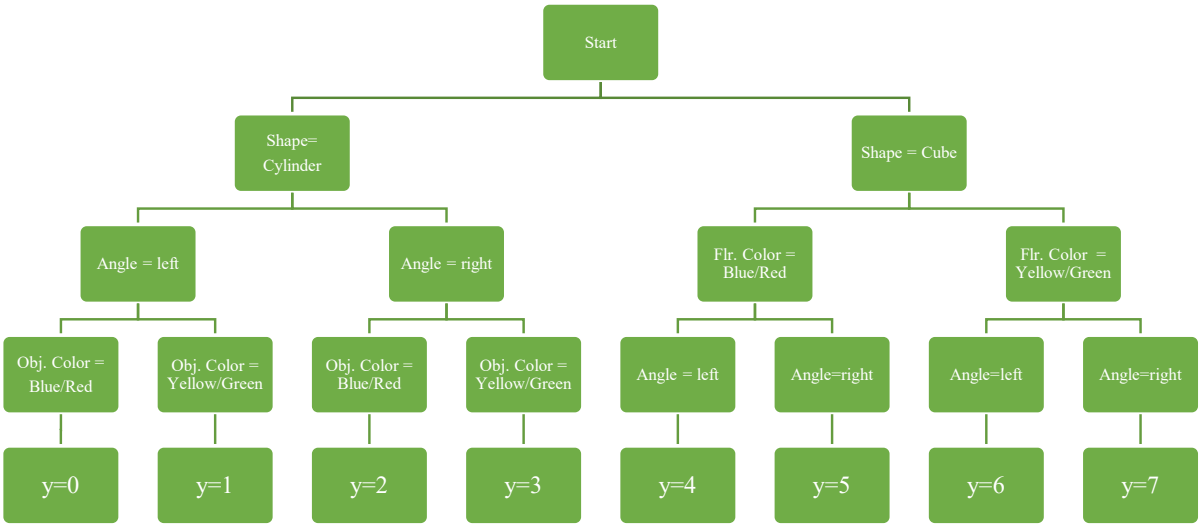


Figure 9: The decision tree setup that we use for the discriminative classification problem. Each image is assigned one out of eight class labels y according to the following decision tree.

second experiment, we fill Γ with several correlations in the following order:

$$\begin{array}{c} \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{array} \begin{pmatrix} 1 & 4 & 12 & 14 & 9 \\ & 11 & 5 & 10 & 6 \\ & & 3 & 8 & 15 \\ & & & 13 & 7 \\ & & & & 2 \\ & & & & \end{pmatrix} \tag{14}$$

where the component order of the rows and columns is $z_1 = \texttt{floor\_color}$, $z_2 = \texttt{background\_color}$, $z_3 = \texttt{object\_color}$, $z_4 = \texttt{object\_scale}$, $z_5 = \texttt{object\_shape}$, $z_6 = \texttt{orientation}$. Here, it is important to ascertain that the covariance matrix stays positive definite. Thus, we start with $\rho = 0.7$, check if the lowest eigenvalue of Σ is at least 0.2, and if not, reduce ρ by a factor of 0.9 until the eigenvalue fulfills this property. While technically it would be enough to have the smallest eigenvalue anywhere above 0, we found that 0.2 helps in numerical stability, for instance when inverting the covariance matrix to compute the multivariate normal distribution density.

C.4 DISCRIMINATIVE SETUP

The decision tree that is used to generate the class distribution is shown in Figure 9. It relies on 4 (binarized) components. We trained a simple CNN classifier for this problem using the cross-entropy loss. In addition to the classification loss terms, we add a regularizer $\|\boldsymbol{z}\|_2^2$, which constrains the latent codes to not grow arbitrarily large, during training. To create a realistic setup, we subsample the dataset to follow a normal distribution as shown in Fig. 8d. We also add label noise near the decision boundary: For objects which have an orientation that is nearly centered, we follow each branch (left/right) with a probability of 50 %. With increasing left-orientedness, the probability of following the left branch increases to almost 100 % in form of a sigmoid function over the actual orientation. We follow the same procedure for the remaining features. We train the classifier for 10k iterations at a batch size of 24 and verify that it reaches an accuracy close to the best-possible one taking the mislabeled samples into account. We add correlations by increasing the chance of the the factors *obj. color* and *floor color* taking the same binary value. We use our disjoint attributions approach to find a $H \in \mathbb{R}^{4\times 6}$ matrix that should map the 6-dimensional latent space of the model to the four binary concepts that are used in the classification task. For the unit directions, we take the first four unit directions of the latent space, for PCA and ICA, we take the most prominent four components discovered for the evaluation with the four annotated ground truth concepts.

C.5 EVALUATION SCORES

Several scores to quantify disentanglement have been proposed in the literature and often emphasize a different aspect of disentanglement [Sepliarskaia et al., 2019]. Among the most common scores is the Disentanglement-Completeness-Informativenss score (DCI) by Eastwood and Williams [2018]. In their work, they propose a metric to measure Disentanglement, that relies on training predictors $\hat{z}_j = f_j(e)$ to predict each individual ground truth component z_j from the learned latent representation e. Furthermore, they compute normalized importance weights P_{ik} that quantify how important learned component e_i is for predicting the ground component z_k. The disentanglement metric computes a row-wise entropy over the P-matrix, which assigns a score of 1, if the learned component e_i is useful for predicting only a single factor and as score of 0, if it is equally useful for predicting all factors. Other commonly used metrics include the Mutual Information Gap (MIG) [Chen et al., 2018], Separated Attribute Predictability (SAP) [Kumar et al., 2018] and the FactorVAE metric [Kim and Mnih, 2018]. However, it is unclear which of these metrics (or if any) also provide useful results in the correlated setting Träuble et al. [2021]. Therefore, to compute the reliable evaluations, we train the model (and the post-processing methods such as PCA, ICA, IMA, DMA) on the correlated dataset, but compute the metrics on samples from the full, *uncorrelated* datasets to avoid distortion in our scores. Träuble et al. noted that the DCI scores were able to discover entanglement between 2 variables [Träuble et al., 2021, Figure 11, Appendix], whereas most other metrics failed even in this case. Therefore, we mainly rely on this score for our experiments but also report results corresponding to Sec. 4.2 for the other scores that show a similar picture in this appendix (Appendix D.4).

C.6 CUB EXPERIMENTS

CUB-200-2011 is a fine-grained dataset containing a total of 11,788 images of 200 bird species (5994 for training and 5794 for testing). We trained a ResNet-50 with two fully-connected (fc) layers (the second fc layer served as a bottleneck layer and took 2048-dim feature vectors as input and output 512-dim ones) on CUB for 100 epochs using a SGD optimizer with an initial learning rate of 0.001. The input images were center cropped to 224×224 pixels. Trained on a standard cross-entropy loss, the ResNet achieved a classification accuracy of on average 77.47% on five random seeds, indicating proper training. After training the classifier, we applied our proposed method to discover components in the embedding space.

CUB provides no ground-truth components since it is a real-world dataset. It does, however, contain 312 attributes semantically describing the bird classes, e.g., wing color or beak shape. These attributes have no guarantee to be complete, but they offer 312 interpretable components. This allows for an attempt to quantify whether our discovered components are interpretable and meaningful by comparing whether they match some of these interpretable ones.

Formally, we are given a set of image feature embeddings $\{\boldsymbol{e}_n\}_{n=1,\dots,N}$, $\boldsymbol{e}_n \in \mathbb{R}^L$ and a matrix $\boldsymbol{H} = (\boldsymbol{h}_1, \dots, \boldsymbol{h}_K) \in \mathbb{R}^{L \times K}$ that contains the directions of discovered components ($L = 512$, $K = 30$). A score s_n^k of n-th image for the k-th discovered component can be calculated by projecting the feature embeddings on that component direction, i.e., $s_n^k = \langle \boldsymbol{e}_n, \boldsymbol{h}_k \rangle$. One pitfall is that s_n^k can be negative, indicating, e.g., a non-black bird for the component "primary color: black", but this opposite attribute is usually encoded in a separate attribute in CUB, e.g., "primary color: white". Thus, we separate the negative and positive values into two components (where we set values of the opposite sign to 0), resulting in $2 \cdot K$ positive scores for each image.

To compare these component scores with the attributes, we make use of the numerical attribute values provided in CUB. First, we average the $2 \cdot K$ component values of all images of a class, to be comparable with the class-wise attributes provided by CUB. This gives us a numerical $2 \cdot K$ dimensional component description and a 312 dimensional attribute description per class. Now, we match the discovered components to the attributes. We compare each discovered component to each attributes via the Spearman's rank correlation coefficient and consider the attribute with the highest score to match the component. These are the matches used in Sec. 4.5. We further use the (average) Spearman's rank correlation across all components to their best-matching attributes to quantify how well the components match to interpretable attributes in Appendix D.7.

C.7 HYPERPARAMETERS FOR THE DISENTANGLEMENT MODELS

We orient our hyperparameter ranges by the works of Träuble et al. [2021], Locatello et al. [2019]. The exact ranges are provided in Tab. 4. We find the best hyperparameters in the ranges for each correlation strength/dataset/model triple separately. Then we train five models from independent seeds to run our experiments. We use the Adam optimizer for all model with a learning rate of 10^{-4}, batch size of 64 and train for 300k iterations (equiv. to 40 epochs on Shapes3D).

Model	Ranges
BetaVAE	$\beta \in \{1, 2, 4, 6, 8, 16\}$
FactorVAE	$\gamma \in \{5, 8, 10, 20, 30, 40, 50, 100\}$
BetaTCVAE	$\beta \in \{1, 2, 4, 6, 8, 10\}$
DIPVAEI	$\lambda_{od} \in \{1, 2, 5, 10, 20, 50\}$

Table 4: The hyperparameter ranges considered in this work.

Dataset	MPI3D-real		
Correlated components	background & object color	background & robot arm dof-1	robot arm dof-1 & robot arm dof-2
BetaVAE	0.340 ± 0.027	0.277 ± 0.026	0.300 ± 0.046
+PCA	0.116 ± 0.008	0.174 ± 0.021	0.154 ± 0.015
+ICA	0.237 ± 0.042	0.205 ± 0.023	0.180 ± 0.021
+Ours (IMA)	$\mathbf{0.355 \pm 0.033}$	$\mathbf{0.349 \pm 0.015}$	$\mathbf{0.337 \pm 0.038}$
+Ours (DMA)	0.334 ± 0.025	0.317 ± 0.028	0.278 ± 0.030
FactorVAE	$\mathbf{0.205 \pm 0.022}$	$\mathbf{0.239 \pm 0.017}$	0.171 ± 0.005
+PCA	0.179 ± 0.010	0.234 ± 0.012	0.171 ± 0.006
+ICA	0.066 ± 0.009	0.090 ± 0.006	0.073 ± 0.011
+Ours (IMA)	0.201 ± 0.019	0.226 ± 0.010	$\mathbf{0.191 \pm 0.011}$
+Ours (DMA)	0.184 ± 0.013	0.218 ± 0.016	0.180 ± 0.013
BetaTCVAE	$\mathbf{0.383 \pm 0.022}$	$\mathbf{0.359 \pm 0.026}$	$\mathbf{0.309 \pm 0.036}$
+PCA	0.356 ± 0.022	0.328 ± 0.017	0.295 ± 0.038
+ICA	0.245 ± 0.041	0.260 ± 0.024	0.170 ± 0.045
+Ours (IMA)	0.323 ± 0.025	0.316 ± 0.029	0.271 ± 0.033
+Ours (DMA)	0.327 ± 0.027	0.325 ± 0.025	0.272 ± 0.033
DipVAE	0.235 ± 0.019	0.181 ± 0.049	0.232 ± 0.040
+PCA	0.090 ± 0.005	0.088 ± 0.028	0.091 ± 0.011
+ICA	0.234 ± 0.019	0.180 ± 0.048	0.232 ± 0.041
+Ours (IMA)	0.230 ± 0.022	0.182 ± 0.048	0.230 ± 0.042
Ours (DMA)	$\mathbf{0.249 \pm 0.026}$	$\mathbf{0.188 \pm 0.049}$	$\mathbf{0.253 \pm 0.051}$

Table 5: MPI-3D dataset: Mean $\pm$ std. err. of the DCI scores (across all components of the dataset) of several models and post-hoc methods applied to their embeddings. Columns show which pair of components was correlated during training.

For the optimization of the post-hoc disentanglement problem, we use slightly different hyperparameters. We use the RMSProp optimizer with learning rate of 10^{-3} and a batch size of 48.

C.8 DETAILS ON THE INTRODUCTORY EXAMPLE

The introductory example is inspired by a real explanation generated for a missclassification of the ResNet50 model pretrained on the ImageNet [Russakovsky et al., 2015] dataset delivered with the popular `pytorch` [Paszke et al., 2017] package. Using the approach devised by Leemann et al. [2022], we use the individual neurons of the classifier's last-layer as concepts and describe them by words. We obtain the conceptual explanation shown in Figure 10. We simplify the explanation for the motivational figure and give the concepts relatable names. However, the gist of the example stays the same.

D ADDITIONAL RESULTS

D.1 RECONSTRUCTION QUALITY

As a check, we investigate the reconstruction quality of the disentanglement models. For the 3D shapes, the reconstruction is very high, but we observe some more serious reconstruction errors on the MPI-3d dataset (see Appendix D.2). Figures 11 and 12 show the original images on the left and the reconstructions of a randomly chosen BetaVAE on the right. On Shapes3D, the BetaVAE is able to reconstruct the image from its embedding representation. On MPI3D-real, it is able to reconstruct the big image parts shared across many pictures (ground, background stripe and background), but becomes blurry in the smaller and more nuanced robot arm and object shapes. This indicates that the information on these components might not be stored

Prediction: **jack-o-lantern**,
True class: **traffic light**

No.	Top Concepts (**jack-o-lantern**)	Weight	Act.
2	candles, burning, flame, fire, fires (rating: 2.28)	0.24	1.92
4	batting helmet, patella, spinal column, hindquarter, passenger car (rating: 2.09)	0.14	1.68
8	birds, flamingo, new world blackbird, duck, october (rating: 2.04)	0.22	0.74

No.	Top Concepts (**traffic light**)	Weight	Act.
1	stoplight, traffic light, traffic, boulevard, transportation system (rating: 2.04)	0.28	1.04
2	candles, burning, flame, fire, fires (rating: 2.28)	0.15	1.92
4	towers, buildings, condos, lamppost, building (rating: 1.93)	0.19	1.18

candles, burning, flame, fire, fires (rating: 2.28)

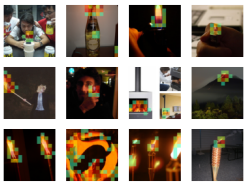


Figure 10: Original local conceptual explanation of the missclassification. We find that the most activating concept "candles, burning, flame..." activates for very dark images. This concept is also highly activated for the traffic light example. We cleared up the description of the concepts for the motivational figure.

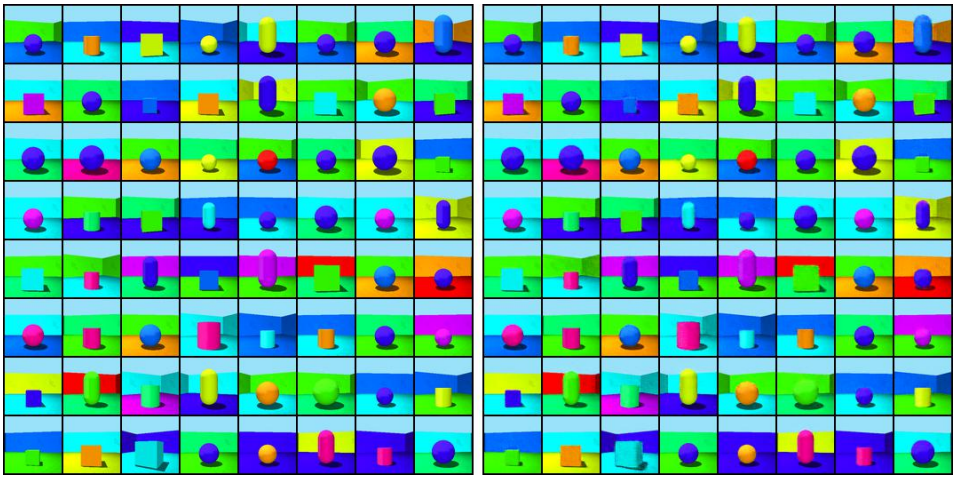


Figure 11: Random example images (left) and their reconstructions (right) of a BetaVAE on Shapes3D.

in the embedding space and is thus hardly disentanglable. A longer training (800k instead of 300k iterations) did not resolve the issue. The issue might arise, following Gondal et al. [2019], because the input images were scaled down to 64x64 pixels making the detailed objects hard to perceive, and because the same architecture as in the Shapes3D experiments was used, which might not be expressive enough.

D.2 RESULTS FOR THE MPI-3D DATASET

In addition to 3Dshapes, we use the challenging MPI3D-real dataset [Gondal et al., 2019], which consists of realistic images of a moving robot arm. It is by far more challenging, as the component is only present in a small portion of the images, and the data consists of real photographs. We report the results on this dataset in Table 5. We saw low disentanglement scores of both the base and post-hoc models on MPI3D-real compared to the performance on Shapes3D. This implies that the embedding spaces of the VAEs was not trained well. In fact, this is supported by the reconstruction quality considerations on both Shapes3D and MPI3D-real. Because our approaches are based on the given embeddings, they also struggle when they incorrectly reflect the sample.

D.3 CORRELATION STRENGTHS AND ATTRIBUTION METHODS IN FIRST EXPERIMENT

In this section we provide additional ablations for the rectification experiment in Sec. 4.2. We investigate the impact of the choice of attribution method (Appendix B.9) and the correlation strength s. The values (DCI scores) are shown in Tab. 8. As expected, our approach offers the highest gains over the baseline when the correlation is higher. Starting at $s = 0.4$, our runs start to reliably outperform the baselines. Regarding the attributions, there is no clear picture, but Grad and SG seem to yield good results more stably across runs. DMA usually outperforms IMA, which supports our theoretical results on

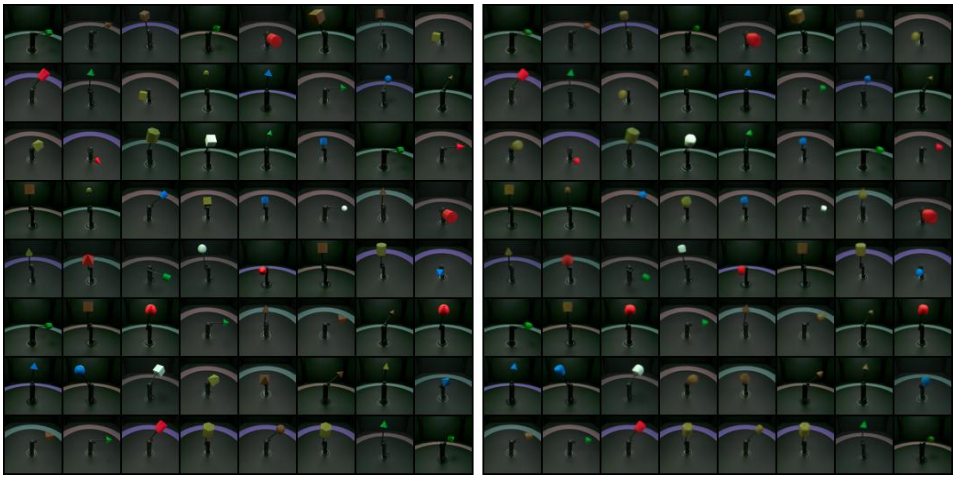


Figure 12: Random example images (left) and their reconstructions (right) of a BetaVAE on MPI3D-real.

identifiability.

D.4 FURTHER DISENTANGLEMENT METRICS

Tables 9 – 11 show the results of the experiment in Sec. 4.2 measured in the alternative metrics MIG, FactorVAE and SAP score. For MIG, we see similar results as for DCI in Table 2 and in Table 5. The results in FactorVAE and SAP score are slightly inferior but our approach still improves over the baseline in many setups. We also compute the disentanglement only on the two correlated components for the first pair of factors in Table 7. This emphasized the improvement introduced by our IMA and DMA approaches.

D.5 RUNTIMES AND FURTHER ABLATION STUDIES

Runtime. Runtime can be an important concern for algorithms in explainable AI, for instance when they are to be deployed on embedded devices. We therefore report the runtimes required to obtain the results shown in Table 2 here:

Algorithm	PCA	ICA	Ours-DMA	Ours-IMA
Runtime (sec)	316 ± 38	320 ± 44	1140 ± 97	1017 ± 121

For our SGD-based optimization, we note that the user can choose how many optimization steps are executed. In the present work, we chose 20000 steps to make sure that the optimization has converged. Using these settings, the runtime of our algorithms is approximately 3 times as high as that of the baseline. We think that this is not prohibitively more expensive. However, convergence of the optimization is usually achieved much quicker.

Effect of less SGD iterations. To ablate the behavior of our approach with a smaller runtime budget, we rerun all the approaches in Table 2 using only 8000 iterations, making the runtime approximately equal across methods. We report the DCI scores as in the original table in Table 6 and see that our DMA approach still outperforms all the baselines in 10 of 12 settings. Thus, even when runtime is an important concern in the evaluation, our approach can still yield competitive results.

Robustness with respect to noise. While IMA covers a more general class of functions, we empirically observed superior performance for DMA in most experiments. We therefore hypothesize that the performance difference stems from the behavior of IMA and DMA under noisy gradients and from the approximate optimizers that we use. We conduct an ablation study to obtain further evidence for these hypotheses. We modify the `FourBars` dataset to fulfill NEMR by adding varying magnitudes of the component gradients in the rows of $\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{x}))$. This dataset is now solvable by both IMA and DMA. We then add noise to the analytical gradients. We perform a fixed number of 500 SGD steps of Algorithm 2 and otherwise use the same optimizer parameters as in the main paper. We obtain the DCI curves across different noise levels shown in Figure 13. Without noise, both algorithms find disentangled solutions with DCI scores >0.9 (practically perfect disentanglement when

Correlated components	floor & background		orientation & background		orientation & size	
BetaVAE	0.497 ± 0.03		0.581 ± 0.04		0.491 ± 0.05	
+PCA	0.263 ± 0.03	-47%	0.310 ± 0.02	-47%	0.324 ± 0.04	-34%
+ICA	$\mathbf{0.574 \pm 0.04}$	+16%	0.540 ± 0.08	-7%	0.577 ± 0.04	+17%
+Ours (OA)	0.533 ± 0.11	+7%	0.594 ± 0.04	+2%	0.576 ± 0.03	+17%
+Ours (DA)	0.472 ± 0.14	-5%	$\mathbf{0.633 \pm 0.05}$	+9%	$\mathbf{0.617 \pm 0.03}$	+26%
FactorVAE	0.507 ± 0.11		0.502 ± 0.08		$\mathbf{0.712 \pm 0.01}$	
+PCA	0.358 ± 0.07	-29%	0.474 ± 0.05	-5%	0.556 ± 0.03	-22%
+ICA	0.294 ± 0.07	-42%	0.263 ± 0.05	-48%	0.340 ± 0.03	-52%
+Ours (OA)	0.539 ± 0.04	+6%	0.498 ± 0.03	-1%	0.568 ± 0.06	-20%
+Ours (DA)	$\mathbf{0.567 \pm 0.07}$	+12%	$\mathbf{0.531 \pm 0.04}$	+6%	0.571 ± 0.02	-20%
BetaTCVAE	0.619 ± 0.01		0.613 ± 0.04		0.659 ± 0.01	
+PCA	0.400 ± 0.03	-35%	0.421 ± 0.07	-31%	0.450 ± 0.07	-32%
+ICA	0.540 ± 0.02	-13%	0.497 ± 0.04	-19%	0.627 ± 0.02	-5%
+Ours (OA)	0.635 ± 0.04	+3%	0.648 ± 0.03	+6%	0.682 ± 0.02	+4%
+Ours (DA)	$\mathbf{0.644 \pm 0.01}$	+4%	$\mathbf{0.659 \pm 0.02}$	+8%	$\mathbf{0.724 \pm 0.02}$	+10%
DipVAE	0.631 ± 0.02		0.652 ± 0.02		0.548 ± 0.04	
+PCA	0.158 ± 0.01	-75%	0.160 ± 0.02	-75%	0.170 ± 0.02	-69%
+ICA	0.630 ± 0.02	-0%	0.651 ± 0.02	-0%	0.542 ± 0.03	-1%
+Ours (OA)	0.640 ± 0.01	+1%	0.621 ± 0.02	-5%	0.545 ± 0.05	-1%
+Ours (DA)	$\mathbf{0.683 \pm 0.01}$	+8%	$\mathbf{0.676 \pm 0.01}$	+4%	$\mathbf{0.591 \pm 0.06}$	+8%

Table 6: Using 8000 instead of 20000 SGD iterations: Mean $\pm$ std. err. of the DCI scores of post-hoc methods applied to the embedding spaces of four disentanglement architectures with different pairs of correlated variables. Our DMA method still yields competitive results even with fewer SGD steps.

Dataset	Shapes3D
Correlated factors	floor vs. background
BetaVAE	0.579 ± 0.089
+PCA	0.291 ± 0.033
+ICA	0.435 ± 0.076
+IMA-SGD	$\underline{0.738 \pm 0.072}$
+DMA-SGD	$\mathbf{0.868 \pm 0.025}$
FactorVAE	0.684 ± 0.163
+PCA	0.526 ± 0.136
+ICA	0.363 ± 0.097
+IMA-SGD	$\underline{0.779 \pm 0.063}$
+DMA-SGD	$\mathbf{0.847 \pm 0.072}$
BetaTCVAE	0.589 ± 0.005
+PCA	0.388 ± 0.046
+ICA	0.609 ± 0.065
+IMA-SGD	$\mathbf{0.876 \pm 0.027}$
+DMA-SGD	$\underline{0.754 \pm 0.127}$
DipVAE	0.615 ± 0.114
+PCA	0.429 ± 0.169
+ICA	0.585 ± 0.024
+IMA-SGD	$\mathbf{0.798 \pm 0.099}$
+DMA-SGD	$\underline{0.782 \pm 0.009}$

Table 7: Mean $\pm$ std. err. of the DCI scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The DCI is computed across **the two correlated components** of the dataset.

evaluated on traversals). When we add noise, the disentanglement scores decrease as the working assumptions now only hold approximately. At a noise level of 0.1, the actual gradients shown in Figure 3a are hard to see already with bare eyes. At each point there is a small but consistent gap between the performance of IMA and DMA, indicating that the DMA objective often finds better solutions with the standard SGD optimizer pipeline. This matches our empirical findings of the real data experiments.

D.6 QUALITATIVE RESULTS ON SHAPES3D

In this section, we want to show another traversal plot like the one in Fig. 4 and more thoroughly analyze its latent space. We chose another architecture (BetaTCVAE) and $s = 0.2$ with the usual correlated factors *floor color* and *background color*. Out of the 5 independent runs, we selected the one with the highest DCI score (of the base model) for the analysis.

Linear entanglement matrix. To study which factors are encoded in which latent dimension, we compute a matrix of linear entanglement. By our linear entanglement hypothesis, $\boldsymbol{z}' = \boldsymbol{D}\boldsymbol{z}$, where the matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_K] \in \mathbb{R}^{K \times K}$ contains the directions $\boldsymbol{d}_i \in \mathbb{R}^K$, in which the ground truth concepts are encoded. Changing the component i (entry $\boldsymbol{z}_i$) by one unit will change the resulting embedding by $\boldsymbol{d}_i$. To find these $\boldsymbol{d}_i$, we take the factors at the origin of the traversal plot and alter only a single component i. We then encode the image corresponding to that change, and measure the change in embeddings to find the linear direction $\boldsymbol{d}_i$ that the corresponding component is encoded in (to be precise, we sample several changes and take the largest eigenvector of the embedding changes covariance). Thus, we can estimate the matrix $\boldsymbol{D}$. An example is shown in Fig. 14a and provides evidence that linear entanglement is possible when training autoencoder models from correlated data.

To estimate which factors are changing when a unit direction of the (plain or post-processed) embedding space is followed (a change in $\boldsymbol{z}_i'$), we can invert the equation to $\boldsymbol{z} = \boldsymbol{D}^{-1}\boldsymbol{z}'$. The columns in $\boldsymbol{D}^{-1}$ correspond to the change in ground truth components that going one unit in the latent space coordinate i will entail. We refer to this matrix $\boldsymbol{D}^{-1}$, that shows which ground truth components will be altered by moving along one latent dimension as *linear entanglement matrix*.

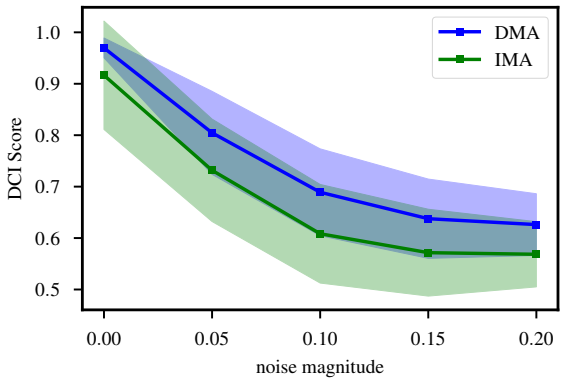


Figure 13: Robustness of optimization to noisy gradients. We use a variant of the `FourBars` dataset that can be identified both by IMA and DMA (the NEMR condition holds) and add noise of increasing magnitude to the analytical gradients. While the disentanglement scores (DCI) decrease for both methods, we observe that the performance of IMA under noise is slightly worse than that of DMA. This may be one factor contributing to the weaker overall performance of IMA as compared to DMA.

Figure 15 shows the traversals along with the corresponding linear entanglement matrices that correspond well to the changes observed. For the plain method, the components that were correlated are deeply entangled (upper line). However, our method (DMA, SG, lower line) is able to separate them well, which is testified both by the traversal and the linear disentanglement matrix.

D.7 FURTHER RESULTS ON CUB

For a quantitative evaluation, we match the discovered concepts on CUB with the annotated ground truth attributes. we report results for the quantitative comparison on CUB introduced in Appendix C.6 of our methods with PCA, ICA, and a baseline of randomly sampled directions. Furthermore, we implement ConceptSHAP [Yeh et al., 2019] and ACE [Ghorbani et al., 2019] and use them to discover concepts on CUB (using their default settings otherwise). The results of this metric are presented in Table 12.

ICA failed to discover meaningful components, while PCA was only capable of discovering very few high-variance ones in the beginning, but begins to fail for $K > 10$. This is possibly because in PCA, the directions are required to be orthogonal. Surprisingly, both PCA and ICA were not much better than the random baseline. Regarding ConceptSHAP and ACE, we find that ACE often focused on the background concepts and ConceptSHAP discovered concepts that are usually more focused on the birds but hard to localize in a fine-grained manner. Our method constantly discovered components and surpassed all three baselines. In particular, our method (DMA) lead to good performance. This leads us to the hypotheses that for high-dimensional data, the disjointness principle is required to identify solutions. Figure 16 illustrates the correlation between the ground-truth attribute representation (scores) and predicted representation by using our model (using plain gradients) for the top discovered component. The two components are clearly correlated, but more in a block-sense: Classes with low scores on the attribute received low scores on the discovered component. The same holds for high scores, but within these, we observe stronger noise, which explains why the Spearman's correlation values were imperfect. This can be due to a certain degree of arbitrage in the ground-truth attribute values of each class. Here, Fig. 17, just like Fig. 6 in the main paper, shows qualitative examples, including the ground-truth values which appear to fluctuate. We emphasize that this analysis should be viewed as an initial take on quantifying the quality of interpretable components, but that a refined benchmark is material for future work.

Model	BetaVAE			FactorVAE			BetaTCVAE			DIPVAEI		
Correlation	$s = 0.2$	$s = 0.4$	$s = \infty$	$s = 0.2$	$s = 0.4$	$s = \infty$	$s = 0.2$	$s = 0.4$	$s = \infty$	$s = 0.2$	$s = 0.4$	$s = \infty$
unit dirs.	0.666 ±0.030	0.497 ±0.028	0.650 ±0.049	0.441 ±0.065	0.507 ±0.105	0.651 ±0.087	0.580 ±0.022	0.619 ±0.008	0.504 ±0.056	0.686 ±0.072	0.631 ±0.018	0.868 ±0.052
PCA	0.287 ±0.010	0.263 ±0.028	0.357 ±0.024	0.312 ±0.048	0.358 ±0.075	0.484 ±0.064	0.341 ±0.018	0.400 ±0.030	0.396 ±0.061	0.266 ±0.029	0.158 ±0.013	0.215 ±0.037
ICA	0.394 ±0.099	0.574 ±0.040	0.674 ±0.012	0.193 ±0.052	0.294 ±0.070	0.390 ±0.109	0.516 ±0.019	0.540 ±0.023	**0.642 ±0.007**	0.672 ±0.073	0.630 ±0.018	**0.870 ±0.049**
Grad (IMA)	0.638 ±0.067	0.617 ±0.018	0.556 ±0.109	0.478 ±0.046	0.551 ±0.040	**0.666 ±0.041**	0.548 ±0.035	0.623 ±0.021	0.551 ±0.038	0.705 ±0.062	0.644 ±0.019	0.794 ±0.043
IG (IMA)	0.702 ±0.035	0.460 ±0.128	0.578 ±0.117	0.470 ±0.035	0.511 ±0.042	0.581 ±0.066	0.619 ±0.024	0.533 ±0.006	0.612 ±0.024	0.650 ±0.072	0.605 ±0.006	0.701 ±0.045
SG (IMA)	0.677 ±0.037	0.438 ±0.127	0.609 ±0.131	0.475 ±0.042	0.561 ±0.040	0.644 ±0.055	0.533 ±0.028	0.620 ±0.021	0.559 ±0.040	0.698 ±0.060	0.642 ±0.017	0.785 ±0.046
Grad (DMA)	0.645 ±0.067	**0.641 ±0.031**	**0.690 ±0.062**	0.547 ±0.056	0.584 ±0.047	0.385 ±0.169	**0.629 ±0.033**	0.666 ±0.010	0.598 ±0.057	**0.717 ±0.059**	**0.684 ±0.009**	0.857 ±0.037
IG (DMA)	0.645 ±0.076	0.530 ±0.106	0.548 ±0.114	**0.573 ±0.046**	**0.615 ±0.045**	0.631 ±0.128	0.607 ±0.028	0.624 ±0.021	0.584 ±0.039	0.703 ±0.073	0.659 ±0.008	0.771 ±0.029
SG (DMA)	**0.711 ±0.040**	0.593 ±0.094	0.633 ±0.062	0.506 ±0.057	0.600 ±0.027	0.644 ±0.066	0.628 ±0.033	**0.670 ±0.014**	0.595 ±0.059	0.716 ±0.059	0.682 ±0.010	0.851 ±0.036

Table 8: Mean ± std. err. of the DCI score of the experiments in Sec. 4.2 for the first correlated component pair (*floor* vs *background* color) in Shapes3D, as an ablation study with further correlations strengths and attribution methods (see Appendix B.9). We observe only small differences between attribution methods, with plain Grad and SG performing best in the DMA setting.

Dataset	Shapes3D			MPI3D-real		
Correlated factors	floor vs. background	orientation vs. background	orientation vs. size	background vs. object color	background vs. robot arm dof-1	robot arm dof-1 vs. robot arm dof-2
BetaVAE	0.309 ± 0.031	0.426 ± 0.043	0.335 ± 0.059	0.232 ± 0.022	0.185 ± 0.031	$\mathbf{0.196 \pm 0.034}$
+PCA	0.111 ± 0.031	0.101 ± 0.009	0.092 ± 0.031	0.095 ± 0.010	0.105 ± 0.023	0.123 ± 0.033
+ICA	0.360 ± 0.040	0.324 ± 0.054	0.277 ± 0.036	0.155 ± 0.025	0.163 ± 0.014	0.071 ± 0.014
+Ours (IMA)	0.511 ± 0.029	0.437 ± 0.044	0.502 ± 0.030	$\mathbf{0.239 \pm 0.021}$	$\mathbf{0.229 \pm 0.022}$	0.187 ± 0.039
+Ours (DMA)	$\mathbf{0.594 \pm 0.023}$	$\mathbf{0.485 \pm 0.057}$	$\mathbf{0.545 \pm 0.034}$	0.193 ± 0.036	0.092 ± 0.038	0.080 ± 0.015
FactorVAE	0.297 ± 0.084	0.319 ± 0.076	$\mathbf{0.423 \pm 0.018}$	0.079 ± 0.001	0.103 ± 0.020	0.080 ± 0.010
+PCA	0.202 ± 0.057	0.135 ± 0.028	0.235 ± 0.036	$\mathbf{0.111 \pm 0.006}$	$\mathbf{0.122 \pm 0.011}$	$\mathbf{0.107 \pm 0.009}$
+ICA	0.199 ± 0.061	0.106 ± 0.025	0.078 ± 0.021	0.018 ± 0.008	0.061 ± 0.015	0.069 ± 0.015
+Ours (IMA)	$\mathbf{0.337 \pm 0.033}$	$\mathbf{0.322 \pm 0.056}$	0.288 ± 0.092	0.070 ± 0.014	0.086 ± 0.018	0.039 ± 0.014
+Ours (DMA)	0.276 ± 0.036	0.217 ± 0.064	0.213 ± 0.036	0.046 ± 0.021	0.045 ± 0.016	0.048 ± 0.015
BetaTCVAE	0.333 ± 0.008	0.400 ± 0.046	0.402 ± 0.017	$\mathbf{0.279 \pm 0.025}$	$\mathbf{0.223 \pm 0.030}$	0.201 ± 0.039
+PCA	0.249 ± 0.033	0.145 ± 0.039	0.184 ± 0.062	0.265 ± 0.019	0.203 ± 0.028	$\mathbf{0.213 \pm 0.035}$
+ICA	0.390 ± 0.031	0.276 ± 0.043	0.346 ± 0.072	0.199 ± 0.040	0.158 ± 0.038	0.170 ± 0.033
+Ours (IMA)	0.484 ± 0.025	0.490 ± 0.033	0.526 ± 0.036	0.092 ± 0.029	0.071 ± 0.029	0.041 ± 0.014
+Ours (DMA)	$\mathbf{0.525 \pm 0.014}$	$\mathbf{0.540 \pm 0.021}$	$\mathbf{0.620 \pm 0.024}$	0.120 ± 0.037	0.122 ± 0.044	0.075 ± 0.028
DipVAE	0.493 ± 0.032	0.481 ± 0.020	0.433 ± 0.044	0.138 ± 0.020	0.099 ± 0.040	$\mathbf{0.143 \pm 0.045}$
+PCA	0.063 ± 0.006	0.086 ± 0.027	0.108 ± 0.014	0.054 ± 0.016	0.042 ± 0.011	0.064 ± 0.010
+ICA	0.495 ± 0.032	0.438 ± 0.053	0.224 ± 0.026	0.138 ± 0.023	0.096 ± 0.040	0.139 ± 0.047
+Ours (IMA)	0.512 ± 0.042	0.425 ± 0.036	0.465 ± 0.049	$\mathbf{0.146 \pm 0.019}$	$\mathbf{0.105 \pm 0.033}$	0.136 ± 0.049
+Ours (DMA)	$\mathbf{0.591 \pm 0.028}$	$\mathbf{0.546 \pm 0.017}$	$\mathbf{0.497 \pm 0.060}$	0.133 ± 0.029	0.094 ± 0.036	0.125 ± 0.045

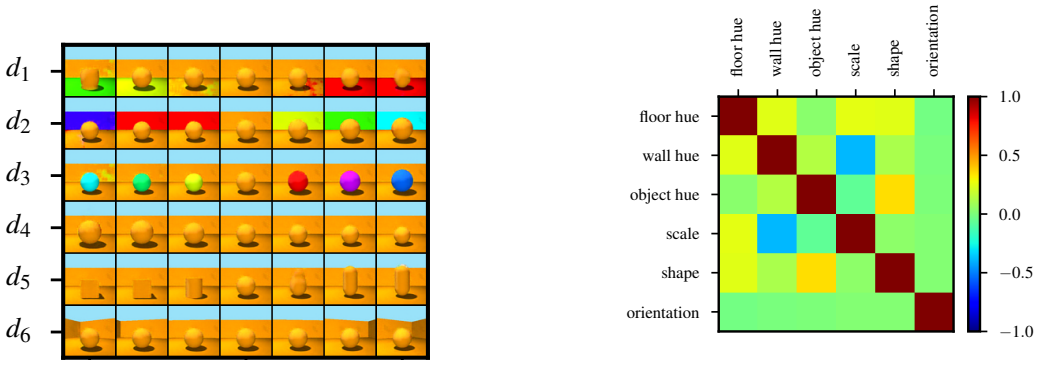
Table 9: Mean ± std. err. of the Mutual-Information Gap (MIG) scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The MIG is computed across all components of the dataset.

Dataset	Shapes3D			MPI3D-real		
Correlated factors	floor vs. background	orientation vs. background	orientation vs. size	background vs. object color	background vs. robot arm dof-1	robot arm dof-1 vs. robot arm dof-2
BetaVAE	$\mathbf{0.834 \pm 0.022}$	$\mathbf{0.839 \pm 0.053}$	0.828 ± 0.011	0.557 ± 0.032	0.490 ± 0.044	0.412 ± 0.022
+PCA	0.722 ± 0.060	0.689 ± 0.047	0.716 ± 0.035	0.393 ± 0.037	0.452 ± 0.031	0.398 ± 0.031
+ICA	0.797 ± 0.036	0.775 ± 0.083	0.794 ± 0.022	0.385 ± 0.100	0.262 ± 0.061	0.251 ± 0.031
+Ours (IMA)	0.767 ± 0.108	0.808 ± 0.060	$\mathbf{0.832 \pm 0.022}$	0.565 ± 0.022	0.504 ± 0.036	0.443 ± 0.027
+Ours (DMA)	0.813 ± 0.087	0.829 ± 0.068	0.826 ± 0.029	$\mathbf{0.567 \pm 0.024}$	$\mathbf{0.525 \pm 0.042}$	$\mathbf{0.444 \pm 0.027}$
FactorVAE	0.636 ± 0.045	0.622 ± 0.064	0.595 ± 0.050	$\mathbf{0.354 \pm 0.016}$	$\mathbf{0.389 \pm 0.015}$	0.342 ± 0.006
+PCA	0.627 ± 0.071	$\mathbf{0.680 \pm 0.027}$	$\mathbf{0.652 \pm 0.024}$	0.330 ± 0.018	0.388 ± 0.022	$\mathbf{0.353 \pm 0.016}$
+ICA	0.619 ± 0.059	0.446 ± 0.146	0.200 ± 0.148	0.277 ± 0.013	0.242 ± 0.082	0.304 ± 0.017
+Ours (IMA)	$\mathbf{0.663 \pm 0.022}$	0.661 ± 0.028	0.644 ± 0.051	0.347 ± 0.007	0.386 ± 0.020	0.337 ± 0.013
+Ours (DMA)	0.646 ± 0.026	0.637 ± 0.023	0.619 ± 0.026	0.330 ± 0.015	0.375 ± 0.016	0.335 ± 0.013
BetaTCVAE	0.676 ± 0.012	0.814 ± 0.052	0.877 ± 0.015	0.445 ± 0.044	0.379 ± 0.021	0.346 ± 0.020
+PCA	0.761 ± 0.035	0.738 ± 0.063	0.794 ± 0.037	$\mathbf{0.505 \pm 0.040}$	$\mathbf{0.425 \pm 0.012}$	0.389 ± 0.008
+ICA	0.834 ± 0.004	0.761 ± 0.051	0.806 ± 0.051	0.149 ± 0.099	0.168 ± 0.053	0.057 ± 0.035
+Ours (IMA)	0.837 ± 0.004	0.849 ± 0.015	$\mathbf{0.879 \pm 0.013}$	0.463 ± 0.048	0.401 ± 0.018	0.399 ± 0.019
+Ours (DMA)	$\mathbf{0.842 \pm 0.000}$	$\mathbf{0.854 \pm 0.017}$	0.878 ± 0.013	0.460 ± 0.046	0.399 ± 0.018	$\mathbf{0.399 \pm 0.014}$
DipVAE	$\mathbf{0.826 \pm 0.006}$	0.839 ± 0.006	0.785 ± 0.033	$\mathbf{0.517 \pm 0.046}$	$\mathbf{0.473 \pm 0.046}$	0.430 ± 0.013
+PCA	0.671 ± 0.019	0.603 ± 0.064	0.653 ± 0.039	0.431 ± 0.028	0.373 ± 0.027	0.344 ± 0.021
+ICA	0.826 ± 0.006	0.831 ± 0.007	0.749 ± 0.027	0.434 ± 0.042	0.423 ± 0.027	0.424 ± 0.012
+Ours (IMA)	0.824 ± 0.007	0.812 ± 0.018	0.785 ± 0.029	0.503 ± 0.044	0.471 ± 0.035	0.436 ± 0.021
+Ours (DMA)	0.822 ± 0.006	$\mathbf{0.850 \pm 0.012}$	$\mathbf{0.809 \pm 0.045}$	0.505 ± 0.040	0.459 ± 0.040	$\mathbf{0.448 \pm 0.026}$

Table 10: Mean $\pm$ std. err. of the FactorVAE scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The FactorVAE score is computed across all components of the dataset.

Dataset	Shapes3D			MPI3D-real		
Correlated factors	floor vs. background	orientation vs. background	orientation vs. size	background vs. object color	background vs. robot arm dof-1	robot arm dof-1 vs. robot arm dof-2
BetaVAE	0.086 ± 0.003	0.119 ± 0.004	0.100 ± 0.005	0.127 ± 0.014	0.098 ± 0.015	$\mathbf{0.092 \pm 0.025}$
+PCA	0.047 ± 0.005	0.062 ± 0.006	0.066 ± 0.006	0.027 ± 0.005	0.055 ± 0.008	0.037 ± 0.006
+ICA	0.007 ± 0.001	0.013 ± 0.001	0.019 ± 0.004	0.017 ± 0.006	0.007 ± 0.002	0.004 ± 0.001
+Ours (IMA)	$\mathbf{0.099 \pm 0.026}$	0.114 ± 0.008	0.112 ± 0.007	$\mathbf{0.131 \pm 0.011}$	$\mathbf{0.113 \pm 0.005}$	0.082 ± 0.024
+Ours (DMA)	0.094 ± 0.020	$\mathbf{0.127 \pm 0.012}$	$\mathbf{0.114 \pm 0.013}$	0.107 ± 0.025	0.059 ± 0.024	0.037 ± 0.013
FactorVAE	0.072 ± 0.006	0.059 ± 0.006	$\mathbf{0.064 \pm 0.001}$	0.059 ± 0.004	0.066 ± 0.008	0.054 ± 0.003
+PCA	0.060 ± 0.006	$\mathbf{0.066 \pm 0.004}$	0.057 ± 0.004	$\mathbf{0.065 \pm 0.008}$	$\mathbf{0.076 \pm 0.004}$	$\mathbf{0.071 \pm 0.003}$
+ICA	0.013 ± 0.002	0.008 ± 0.001	0.006 ± 0.002	0.002 ± 0.000	0.002 ± 0.001	0.001 ± 0.000
+Ours (IMA)	$\mathbf{0.077 \pm 0.012}$	0.052 ± 0.005	0.054 ± 0.017	0.054 ± 0.006	0.059 ± 0.006	$\mathbf{0.036 \pm 0.015}$
+Ours (DMA)	0.071 ± 0.014	0.053 ± 0.012	0.040 ± 0.010	0.041 ± 0.017	0.043 ± 0.015	0.044 ± 0.013
BetaTCVAE	0.052 ± 0.002	0.107 ± 0.013	0.096 ± 0.016	$\mathbf{0.151 \pm 0.017}$	$\mathbf{0.133 \pm 0.007}$	$\mathbf{0.117 \pm 0.011}$
+PCA	0.073 ± 0.004	0.075 ± 0.011	0.107 ± 0.015	0.148 ± 0.018	0.125 ± 0.009	0.109 ± 0.007
+ICA	0.015 ± 0.000	0.010 ± 0.001	0.011 ± 0.002	0.011 ± 0.004	0.005 ± 0.002	0.004 ± 0.002
+Ours (IMA)	0.105 ± 0.003	0.119 ± 0.012	$\mathbf{0.130 \pm 0.023}$	0.055 ± 0.017	0.059 ± 0.016	0.056 ± 0.003
+Ours (DMA)	$\mathbf{0.108 \pm 0.005}$	$\mathbf{0.127 \pm 0.013}$	0.109 ± 0.017	0.071 ± 0.020	0.072 ± 0.010	0.051 ± 0.015
DipVAE	0.083 ± 0.004	0.084 ± 0.003	0.070 ± 0.002	0.056 ± 0.011	0.039 ± 0.013	0.057 ± 0.016
+PCA	0.027 ± 0.003	0.034 ± 0.006	0.043 ± 0.004	0.023 ± 0.004	0.030 ± 0.008	0.022 ± 0.005
+ICA	0.006 ± 0.001	0.003 ± 0.002	0.030 ± 0.002	0.011 ± 0.005	0.005 ± 0.003	0.005 ± 0.002
+Ours (IMA)	0.089 ± 0.012	0.082 ± 0.005	0.077 ± 0.002	$\mathbf{0.060 \pm 0.008}$	$\mathbf{0.047 \pm 0.010}$	$\mathbf{0.061 \pm 0.016}$
+Ours (DMA)	$\mathbf{0.114 \pm 0.003}$	$\mathbf{0.105 \pm 0.008}$	$\mathbf{0.084 \pm 0.007}$	0.051 ± 0.008	0.043 ± 0.012	0.054 ± 0.016

Table 11: Mean $\pm$ std. err. of the SAP scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The SAP score is computed across all components of the dataset.



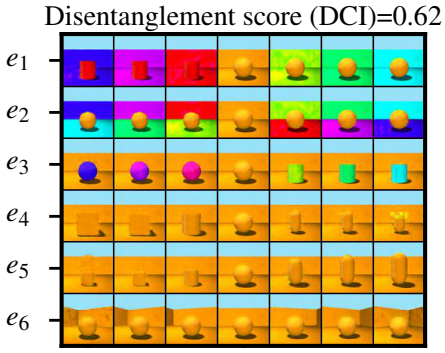
(a) optimal linear directions (traversal plot)

(b) Cosines of the linear directions

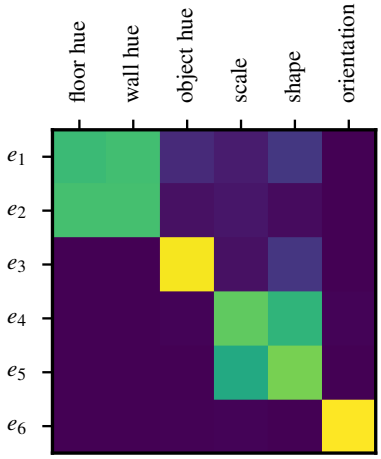
Figure 14: Empirical results for linear entanglement. For the model shown in Fig. 4 (trained on correlated data), we observe almost perfect linear entanglement, i.e., that $f \circ g = D$: (a) There exist linear directions d_1 to d_6 in f's embedding space that encode the individual components. (b) However, these directions are not necessarily orthogonal; they can be entangled as testified by non-zero cosine distances between them. See Fig. 15 for additional results.

Num. components	K=1	K=10	K=20	K=30
PCA	**0.789** ± 0.024	0.602 ± 0.007	0.497 ± 0.005	0.440 ± 0.006
ICA	0.515 ± 0.028	0.442 ± 0.005	0.412 ± 0.006	0.390 ± 0.007
ACE [Ghorbani et al., 2019]	0.623 ± 0.012	0.579 ± 0.010	0.550 ± 0.008	0.527 ± 0.007
ConceptSHAP [Yeh et al., 2019]	0.655 ± 0.014	0.596 ± 0.006	0.568 ± 0.008	0.545 ± 0.006
Ours-IMA,Grad	0.657 ± 0.025	0.601 ± 0.009	0.564 ± 0.009	0.535 ± 0.008
Ours-DMA,Grad	0.701 ± 0.045	**0.626** ± 0.029	**0.585** ± 0.028	**0.559** ± 0.011

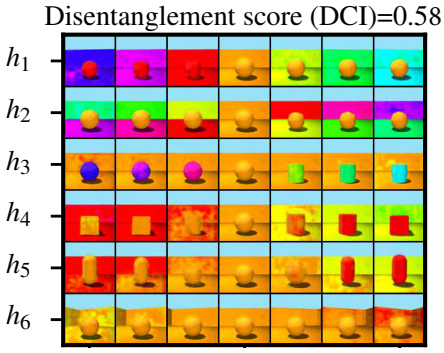
Table 12: Quantitative comparison of discovered components using our methods, PCA, ICA and a random baseline. Mean correlation score of top-K (K in column) discovered components are shown in (mean ± std.) for five runs.



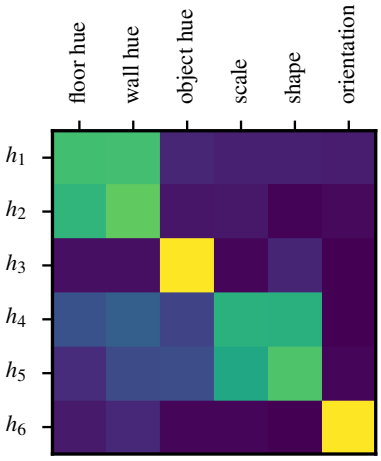
(a) BetaTCVAE



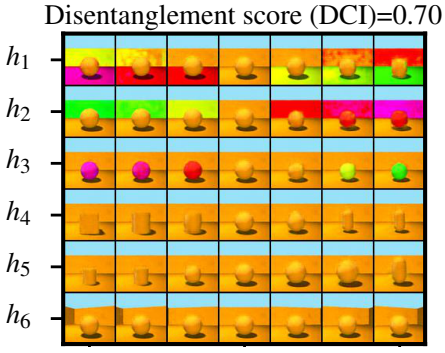
(b) Corresponding linear entanglement matrix



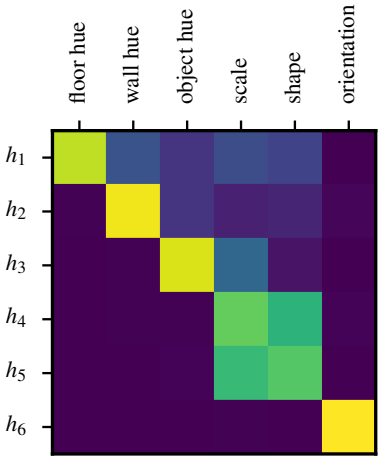
(c) BetaTCVAE + ICA



(d) Corresponding linear entanglement matrix



(e) BetaTCVAE + Ours (DMA, SG)



(f) Corresponding linear entanglement matrix

Figure 15: Traversal plots from another model (BetaTCVAE) trained on the correlated dataset. As for all traversal plots in this paper, we manually permuted the dimensions to match across plots. In addition, we compute a matrix of linear entanglement that shows which ground truth factors is changed when moving into a certain direction (brightness corresponds to maginitude of change). While none of the post-hoc methods manages to disentangle shape and size (most likely due to their non-linear encoding), our model resolves the linearly entangled factors *floor hue* and *wall hue* fairly well, which can also be seen from the entanglement matrix.

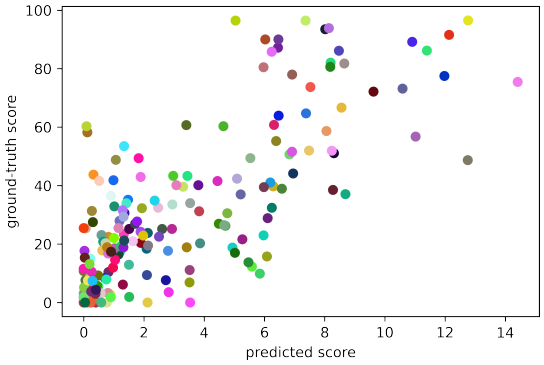


Figure 16: Correlation between ground-truth attribute scores and our predicted scores for the best matched component. Each dot represents a class.

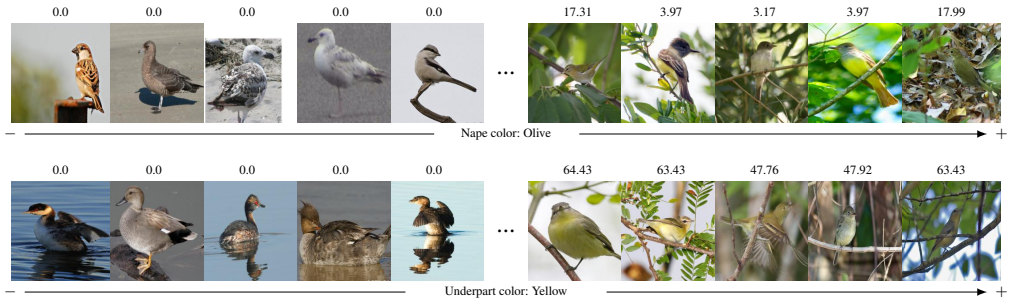


Figure 17: Examples of discovered components on CUB. The corresponding ground-truth attribute is shown under images and the ground-truth value of each image is depicted above the image. "+/−" indicate the positive/negative direction along the discovered concept.

References

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32, pages 9277–9286, 2019.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems*, 32, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Luigi Gresele, Julius von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. Independent mechanism analysis, a new concept? In *Advances in Neural Information Processing Systems*, 2021.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Tobias Leemann, Yao Rong, Stefan Kraft, Enkelejda Kasneci, and Gjergji Kasneci. Coherence evaluation of visual concepts with objects and language. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Aditya Ramesh, Youngduck Choi, and Yann LeCun. A spectral regularizer for unsupervised disentanglement. *arXiv preprint arXiv:1812.01161*, 2018.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *International Conference on Learning Representations*, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Anna Sepliarskaia, Julia Kiseleva, Maarten de Rijke, et al. Evaluating disentangled representations. *arXiv preprint arXiv:1910.05587*, 2019.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2046–2059, 2021. URL https://proceedings.neurips.cc/paper/2021/file/0fe6a94848e5c68a54010b61b3e94b0e-Paper.pdf.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, ICML*, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pages 10401–10412. PMLR, 2021.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020.

Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6721–6730, 2021.

Chih-Kuan Yeh, Been Kim, Sercan O Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

3.2 Counterfactual Explanations for Decisions with Human Oversight

Publication 2

Tobias Leemann, Martin Pawelczyk, Bardh Prenkaj, and Gjergji Kasneci: Towards Non-Adversarial Algorithmic Recourse. *World Conference on Explainable Artificial Intelligence (XAI)*, 2024. Reproduced with permission from Springer Nature.

Author Contributions. Gjergji Kasneci suggested the initial problem of differentiating between counterfactual and adversarial examples to Martin Pawelczyk and me. I developed the real-world use case to highlight the relevance of the discussion to practitioners. I led the experimentation, which was supported by Martin Pawelczyk, who contributed the synthetic data experiment and Bardh Prenkaj, who contributed an experiment on adversarial training. Gjergji Kasneci and Martin Pawelczyk contributed significantly to the final write-up and through discussions on the problem's practical relevance through the GDPR's oversight requirements.

Summary. Human oversight is a fundamental principle in EU regulation. It is required in Art. 22 of the GDPR (European Parliament, 2016), which asserts the right of the data subject *"not to be subject to a decision based solely on automated processing which produces legal effects concerning him or her"*. A slightly weaker statement can be found in Art. 14(4d) of the AIA. Nevertheless, in the literature on counterfactual explanations, the more realistic scenario where automated decisions have to be approved by a human supervisory board has not been deeply studied.

This contribution connects this case to the academic debate on the difference between adversarial examples (Goodfellow et al., 2014) and counterfactual explanations. Adversarial examples and counterfactuals are computationally similar as they are both concerned with finding a neighboring instance that is assigned a different class label. It has been argued that the unique characteristic of adversarial examples – as opposed to counterfactual explanations – is that they lead to a misclassification compared to the ground truth. This difference is essential when we consider machine-learned decisions that the human supervisory board can override: Supposing that the human supervisors make the final decision, a user who is assigned and implements an adversarial example will be approved by the ML model, but rejected by the human oversight board in the end. This highlights that in high-stakes situations, it is imperative to obtain counterfac-

tual explanations that do not exhibit adversarial characteristics, referred to as *non-adversarial algorithmic recourse*. We investigate how different components in the setup, e.g., the machine learning model or cost function used to measure distance determine whether the outcome can be considered adversarial or not. Surprisingly, our experiments indicate that these design choices are often more critical in deciding whether recourse is non-adversarial, than switching between recourse or attack algorithms. Most prominently, using a robust and accurate machine learning model results in less adversarial recourse in practice.

3.2.1 Discussion

Despite the surprisingly clear mandate that high-stakes decisions require human oversight in the GDPR, the practical case where experts can overturn decisions is seldom considered in the technical XAI literature. Our work is one of the first to explicitly model the human supervisors and the human recipient of the counterfactuals in this realistic scenario. One of our main findings is that counterfactual explanation algorithms often produce adversarial examples, while adversarial attack algorithms often produce valid, non-adversarial counterfactuals. We confirm that many examples generated by current counterfactual explanation methods will flip the model decision but not change the ground truth assigned by human experts. This issue drastically reduces the value of counterfactuals for end users, who might not get approved by the human oversight board even after implementing the assigned recourse correctly. Our work goes beyond similar findings, e.g., by Pawelczyk et al. (2022), by providing practical technical guidance for how counterfactuals should be computed in GDPR-compliant decision-making, for instance by using robust and performant models. While our work presents the problem and valuable first steps towards a solution, we think further work is needed on aligning the predictive models and human experts to systematically reduce the risk of adversarial recourse. It can, therefore, also be seen as a contribution towards making explanations robust in adversarial settings (Bordt et al., 2022).

Towards Non-adversarial Algorithmic Recourse

Tobias Leemann[1,2( )], Martin Pawelczyk[3], Bardh Prenkaj[2], and Gjergji Kasneci[2]

[1] University of Tübingen, Tübingen, Germany
tobias.leemann@uni-tuebingen.de
[2] Technical University of Munich, Munich, Germany
[3] Harvard University, Cambridge, MA, USA

Abstract. The streams of research on adversarial examples and counterfactual explanations have largely been growing independently. This has led to several recent works trying to elucidate their similarities and differences. Most prominently, it has been argued that adversarial examples, as opposed to counterfactual explanations, have a unique characteristic in that they lead to a misclassification compared to the ground truth. However, the computational goals and methodologies employed in existing counterfactual explanation and adversarial example generation methods often lack alignment with this requirement. Using formal definitions of adversarial examples and counterfactual explanations, we introduce non-adversarial algorithmic recourse and outline why in high-stakes situations, it is imperative to obtain counterfactual explanations that do not exhibit adversarial characteristics. We subsequently investigate how different components in the objective functions, e.g., the machine learning model or cost function used to measure distance, determine whether the outcome can be considered an adversarial example or not. Our experiments on common datasets highlight that these design choices are often more critical in deciding whether recourse is non-adversarial than whether recourse or attack algorithms are used. Furthermore, we show that choosing a robust and accurate machine learning model results in less adversarial recourse desired in practice.

Keywords: Counterfactuals · Adversarials · Algorithmic Recourse

1 Introduction

A continuous stream of predominantly independent research in the fields of adversarial examples [26,58] and counterfactual explanations [47,61,63,65] has sparked an ongoing scholarly discourse on their similarities and differences [23,46]. While adversarial examples originate from the security literature, characterizing instances capable of deceiving machine-learned classifiers into erroneous decisions, algorithmic recourse has its roots in the trustworthy machine-learning

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 395–419, 2024.
https://doi.org/10.1007/978-3-031-63800-8_20

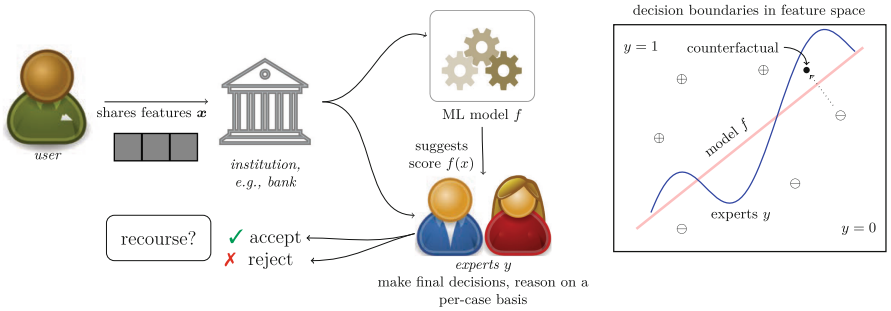


Fig. 1. Overview of the realistic decision-making scenario considered in this work. We consider the relevant case where an institution, e.g., a bank, deploys a machine learning model to support decision-making overseen by human experts that make final, case-based decisions based on the model's score (left). In such a setting, constructing recourse only based on the scoring model f may lead to unreliable recourse because the experts' final y decision is based on further restrictions, thereby representing a shifted decision boundary (right).

literature. Algorithmic recourse is primarily concerned with providing actionable recommendations for changes that would lead to a different, more favorable outcome for the end user (e.g., changing a loan decision from rejection to acceptance). Despite the apparent differences in goals and associated semantics between adversarial examples and recourse, scholars have observed a strikingly similar algorithmic foundation underpinning these two domains [23,46].

The current debate surrounding the potential distinctions between these two concepts remains somewhat ambiguous. To provide greater context and significance to this discourse, we establish a tangible connection to a real-world application where the differentiation between counterfactual and adversarial examples becomes intuitive and indispensable. To this end, we slightly modify the established recourse problem in the context of loan assignments [62]. Unlike previous work, which assumes that a machine learning system solely determines loan assignments, we argue that this perspective oversimplifies the real world. Article 22 of the European Union's General Data Protection Regulation (GDPR) [25], which asserts the right of the data subject *"not to be subject to a decision based solely on automated processing which produces legal effects concerning him or her"*, thereby suggesting that automated models alone cannot make legally binding decisions. Consequently, we consider a more practical scenario where algorithmic decisions are subject to scrutiny by a human expert panel. This expert panel holds the authority to issue a final, case-specific decision and can override the model's recommendation. This refined setup is illustrated in Fig. 1.

Complementarily, the GDPR grants individuals who receive an adverse decision the right to receive *"meaningful information about the logic involved"* which, in a broader context, can be interpreted as the right to "recourse" [64]. When the model exclusively determines decisions, it is evident that recourse can be directly computed from the model itself. However, in the more realistic scenario

considered in this work, where human experts play a role in the final decisions, the model's output does not fully encapsulate the ultimate decision. This raises the question of appropriate recourse design in such a scenario and how to reconcile these two GDPR principles – the right to receive meaningful recourse and the prohibition of fully automated decision-making.

Under the premise that the model has been mainly distilled from past decisions of the experts, we consider the experts as an imperfect oracle providing ground truth labels,[1] whereas the model returns an imperfect approximation of these labels. While the humans decide on a per-case basis, it is hard to directly ask them for specific thresholds, as the interplay of the features quickly makes the task intractable. Therefore, we are interested in computing counterfactual explanations that do not only change the model's prediction but also flip the *true* labels. This perspective aligns with the argument made by Freiesleben [23] that a distinctive feature of adversarial examples, as opposed to counterfactual explanations, is their tendency to be misclassified regarding their true labels. Since counterfactual explanations should also change the true label in this case, this gives rise to the term "non-adversarial algorithmic recourse", i.e., *counterfactual explanations that come with both a change in the model's prediction and a changed ground-truth label*.

Unlike prior work taking a merely definitional view, this work additionally contributes to implementing non-adversarial algorithmic recourse in practical scenarios. In summary, we propose the following contributions:

- **Introduction of a novel recourse problem:** We introduce a novel recourse problem that addresses real-world decision systems wherein human experts play a pivotal role in making case-based decisions, while also considering input from a machine learning model.
- **Proposing non-adversarial recourse as a solution to the realistic recourse problem:** We consider prior work's [23] distinction of adversarial examples and counterfactual explanations and suggest a novel formal definition of *non-adversarial algorithmic recourse*, proving a conceptual bridge between the academic discourse on distinguishing adversarial examples from counterfactual explanations and practical decision-making.
- **Promoting non-adversarial recourse theoretically:** After a theoretical analysis of the problem, we derive optimal cost functions that encourage non-adversarial recourse. Our analysis underscores how feature attributions can be leveraged to identify task-relevant features contributing to less "adversarial" recourse.
- **Empirical Insights:** We are the first to consider several other key components practitioners can manipulate to foster non-adversarial algorithmic recourse. These include improving the robustness and accuracy of the machine learning model and the recourse algorithms. In contrast to parts of the literature which argue that cost functions are central, we empirically find that changes in the model are often more significant than the cost function.

[1] The oracle is imperfect as some labels are generated from "defaults", i.e., false positives of expert decisions.

2 Related Work

Human-Assisted Decisions. In crucial situations, societies rely on human experts for decisions. However, delays and quality issues due to a shortage of experts and a high volume of decisions, e.g., long waits for medical diagnoses, have sparked a debate on when automated or human decision-making should be deployed. A stream of prior works [10,51,59] argue that ML models should make decisions in high-stake domains where they have matched or surpassed the average of human performance. Nevertheless, their decisions can still be worse than those of human experts [53] in some cases. In this direction, works such as [13,14,43] propose to optimize ML models to operate under different automation levels: i.e., take decisions on a fraction of the given instances and leave the rest to human experts. In line with other works [21], we argue that the human factor in the loop in a human-AI partnership cannot be neglected when considering the application of AI on real-world problems [1,27]. This position is also cemented in common data protection laws such as the EU's GDPR [25], which grants individuals a right to object fully automated decision-making. For GDPR-compliant decision-making, human oversight can thus be considered essential. Unlike previous works, we explicitly model a human expert panel in the decision-making setup as depicted in Fig. 1, which makes the generation of reliable recourse much more challenging.

Counterfactual Explanations. There is an established literature on the computation of counterfactual explanations [2,8,34,37,41,50,54,61,65] in variegated domains. According to Guidotti et al. [28], given a classifier f that outputs a decision $f(\boldsymbol{x}) = y$ for an instance $\boldsymbol{x}$, a counterfactual explanation of $\boldsymbol{x}'$ is an instance $\boldsymbol{x}'$ such that $f(\boldsymbol{x}') \neq y$, and the difference between $\boldsymbol{x}$ and $\boldsymbol{x}'$ is minimal. Current research streams include the robustness of counterfactual explanations [18,48,60] and the compatibility with other GPDR principles [49]. We briefly review this research field in the following but point the reader to recent surveys [28,52,63] for a comprehensive overview. Mothilal et al. [41] solve an optimization problem with various constraints, among which user-specified ones for (im)mutable features, to ensure feasibility and diversity when producing a set of counterfactuals for a given input. Carreira-Perpiñán and Hada [8] propose CEODT specifically designed for classification trees, including Oblique Decision Trees (ODTs) [29]. Because the counterfactual optimization problem for ODTs is non-convex, nonlinear, and non-differentiable, CEODT computes an exact solution via the optimization problem within the region represented by each leaf and finally picks the leaf with the best solution. Lastly, Ustun et al. [61] were among the first authors to address the problem of actionability in counterfactual explanations (i.e., recourse). Their method constrains the generated counterfactuals such that manipulations do not change immutable features. Overall, we note that previous literature relies on the common assumption that an automated model acts as a sole decision-maker, which might not be realistic in practical scenarios.

Adversarial Examples. Following Szegedy et al. [58], adversarial examples are instances that contain subtle perturbations – usually via adding small amounts of noise – to instances in the training set. These "new" instances, when fed to an underlying ML model, produce an erroneous output with high confidence. It is possible to build an adversarial example $\boldsymbol{x}'$ which is indistinguishable[2] from $\boldsymbol{x}$ but is classified incorrectly, i.e., $f(\boldsymbol{x}') \neq y$. Successfully generating such examples gives rise to *adversarial attacks* [5,26,39], which can have potentially lethal consequences (e.g., in biosecurity and biotechnology [45], autonomous driving [20,66], and power grid blackouts [24]). Several methods have been proposed in the literature to generate adversarial examples assuming varying degrees of knowledge/access of the model, training data, and methods for injecting perturbations. Goodfellow et al. [26], Kurakin et al. [35], and Moosavi et al., [40] propose methods with gradient and data access to find the minimum ℓ_∞-norm (and ℓ_2-norm respectively) perturbations. With only assuming query access to the target classifier, the authors in [11,44,57] design adversarial examples to tightly control sparsity. We refer the reader to a well-established survey for a comprehensive overview of adversarial examples [3].

Linking Counterfactuals and Adversarial Examples. Strikingly, counterfactual explanations and adversarial examples have conceptual connections and a similar formulation [6,23,65] (see also Sect. 3). Freiesleben [22] highlights conceptual differences in aims, formulation, and use-cases between both sub-fields and suggests that the distinctive formal feature of adversarial examples lies in their misclassification concerning the ground truth. Concurrently, there have been proposals to align recourse with a ground truth. König et al. [34] proposes improvement-focused causal recourse, designed to change the true targets instead of the predictions but relies on causal information. Laugel et al. [36] proposes the notion of "justified recourse" that should be close to a correctly classified instance. On the other hand, Browne et al. [6] focus on deep networks and attribute conceptual differences to the interpretation of semantics in the hidden layers of deep networks. Pawelczyk et al. [46] formalize the similarities between popular counterfactual explanations and adversarial example generation methods identifying conditions when they are equivalent. Trying to disentangle and reconcile the various distinctions made in prior works, we provide formal definitions in the next section. Besides König et al. [34], who rely on causal information, there have been few attempts to implement recourse that follows the ground truth. In this work, we provide valuable insights on how to implement non-adversarial recourse in practical decision-making.

3 Preliminaries

We first formalize the general problem considered in this work, before we provide the relevant distinctions between adversarial examples and counterfactual explanations.

[2] We invite the reader to think about images in this context, as described in [26]. Additionally, some works analyze perturbations – e.g., adversarial patches – that are perceptually distinguishable by humans but fool the classifier f [16,19,67].

3.1 The General Problem

Both recourse and adversarial methods consider a fixed machine learning model $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^k$. We usually consider the binary classification problem, where the label is binary, i.e., $\mathcal{Y} = \{0, 1\}$ or a numerical score, $\mathcal{Y} = \mathbb{R}$.

We suppose there is another function $y : \mathcal{X} \to \mathcal{Y}$ that assigns the true labels and represents the human experts in our introductory example. In practice, it is impossible to perfectly learn this function with a model, for instance due to insufficient data coverage or additional circumstances that can be taken into considerations only by the human experts. However, it is possible to query y a limited number of times, as it is possible to present the experts with an example and ask for their decision. We model the expert predictions y in the scenario outlined as

$$y(\boldsymbol{x}) = g(\boldsymbol{x}, f(\boldsymbol{x})), \tag{1}$$

where g models the human expert committee that can recalibrate the score in light of the features in their entirety. However, we suppose that we usually have $y(\boldsymbol{x}) \approx f(\boldsymbol{x})$, i.e., the original score is only lightly adapted through g. In practice, models are fitted on a limited number of potential observations of the experts' decisions.

As noted before [46], the classical optimization problem solved by both practical adversarial and counterfactual methods for a model $f : \mathcal{X} \to \mathcal{Y}$ a factual input $\boldsymbol{x} \in \mathcal{X}$, and a target label $y_t \in \mathcal{Y}$ is mathematically similar and can usually be formalized as a special case of the following general optimization problem [23]:

$$\underset{\boldsymbol{x}' \in \mathcal{X}}{\operatorname{argmin}}\ d_1(\boldsymbol{x}, \boldsymbol{x}') + \lambda d_2(f(\boldsymbol{x}'), y_t), \tag{2}$$

where $d_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a distance metric defined on the input space, $d_2 : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a metric on the output space and $\lambda \in \mathbb{R}_{\geq 0}$ is a non-negative trade-off parameter. Intutively, the solution to this problem returns instances, that are close to the factual $\boldsymbol{x}$ and have a label that is close (or corresponds exactly) to the target label y_t.

3.2 Algorithms for Computing Counterfactual Explanations and Adversarial Examples

We briefly review the most common strategies to compute counterfactuals and adversarial examples in practice.

Score CounterFactual Explanations (SCFE). For a given classifier $f(h(\boldsymbol{x}))$ that relies on logit scores $h(\boldsymbol{x})$ and a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, Wachter et al. [65] formulate the problem of finding a counterfactual $\boldsymbol{x}'$ for $\boldsymbol{x}$ as:

$$\operatorname{argmin}_{\boldsymbol{x}'} \left(h(\boldsymbol{x}') - s\right)^2 + \lambda\, d(\boldsymbol{x}, \boldsymbol{x}'), \tag{3}$$

where s is the target score for $\boldsymbol{x}$. The problem is solved for different values of λ until $f(\boldsymbol{x}') = s$. More specifically, to arrive at a counterfactual probability of

0.5, the target score for $h(\boldsymbol{x})$ for a sigmoid function is $s = 0$. Using the inverse logit transform $h(\boldsymbol{x}) = \text{invlogit}(f(\boldsymbol{x}))$, the first part of the objective can be interpreted as a particular instantiation of d_2 in Eq. (2) when $\mathcal{Y}$ is taken to be the interval $[0, 1]$.

Diverse Counterfactual Explanations (DiCE). As different users may have different preferences (i.e., it might be easier for them to change one feature or another), DiCE [42] generates multiple counterfactuals. An additional loss term is added to the objective in Eq. (3) to encourage diversity. As users will only choose one counterfactual in practice, we usually consider a randomly selected instance of the discovered recourse candidate for evaluation as in [49].

Actionable Recourse (AR). The actionable recourse (AR) method by Ustun et al. [61] sets up the following optimization problem:

$$\min \ \text{cost}(\boldsymbol{\delta}; \boldsymbol{x}) \tag{4}$$

$$\text{s.t.} f(\boldsymbol{x} + \boldsymbol{\delta}) = +1, \delta \in \mathcal{A}(\boldsymbol{x}), \tag{5}$$

where $+1$ corresponds to the positive outcome and a is an action set $\mathcal{A}(\boldsymbol{x})$. This problem corresponds to Eq. (2) when using a distance function d_1 that returns ∞ once $\delta \neq \mathcal{A}(\boldsymbol{x})$ and the cost function otherwise. The distance d_2 can be interpreted as the Dirac-distance, that is ∞ once $f(\boldsymbol{x} + \boldsymbol{\delta}) \neq 1$. They solve the problem using mixed integer linear programming (MIP) for linear models.

Like counterfactual explanations, most adversarial example methods also solve a constrained optimization problem to find perturbations in the input manifold that cause models to misclassify.

C&W Attack. For a given input $\boldsymbol{x}$ and classifier f, Carlini and Wagner [7] formulate the problem of finding an adversarial example $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ such that $f(\boldsymbol{x}') \neq f(\boldsymbol{x})$ as:

$$\underset{\boldsymbol{x}' \in \mathcal{X}}{\text{argmin}} \ c \cdot \ell(\boldsymbol{x}') + d(\boldsymbol{x}, \boldsymbol{x}') \ \ \text{s.t.} \ \ \boldsymbol{x}' \in [0, 1]^d, \tag{6}$$

where $c > 0$ is a suitably chosen hyperparameter, and $\ell(\cdot)$ is an objective function on the adversarial $\boldsymbol{x}'$ s.t. $f(\boldsymbol{x}') = y_t$ iff $\ell(\boldsymbol{x}') \leq 0$ with y_t being a target class. The authors choose $d(\boldsymbol{x}, \boldsymbol{x}')$ to be the l_p norm of $\boldsymbol{\delta}$, i.e., minimizing the p-norm of $\boldsymbol{\delta}$ is equivalent to minimizing $d(\boldsymbol{x}, \boldsymbol{x}')$.

DeepFool Attack. For a given instance $\boldsymbol{x}$, DeepFool [40] perturbs it by adding small perturbation $\boldsymbol{\delta}_{\text{DF}}$ at each iteration. The minimal perturbation to change the classification model's prediction is the solution to the following objective:

$$\boldsymbol{\delta}^*_{\text{DF}}(\mathbf{x}) \in \underset{\boldsymbol{\delta} \ \text{s.t.} \ \boldsymbol{x}+\boldsymbol{\delta} \in \mathcal{X}}{\text{argmin}} ||\boldsymbol{\delta}||_2 \ \text{s.t.} \ \text{sign}(f(\mathbf{x} + \boldsymbol{\delta})) \neq \text{sign}(f(\mathbf{x})) \tag{7}$$

PGD Attack. PGD [38] is a first-order optimization technique. In the context of adversarial examples, it is usually used to maximize[3], the objective for a specific factual $\boldsymbol{x}$. This is because the objective is typically chosen to be the cross-entropy loss $\mathcal{L}$:

$$\underset{\boldsymbol{\delta} \text{ s.t. } \boldsymbol{x}+\boldsymbol{\delta} \in \mathcal{C}}{\operatorname{argmax}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}), f(\boldsymbol{x})) \tag{8}$$

where $\boldsymbol{\delta}$ is the adversarial perturbation to be added to the factual $\boldsymbol{x}$. PGD maximizes the objective by taking steps along the gradient's direction. After each update, the current perturbation $\boldsymbol{\delta}^t$ is projected onto a set of constraints $\mathcal{C}$. For instance, the adversarial examples are all constrained to a ball of size ϵ around $\boldsymbol{x}$. We argue that the projection of the adversarials $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ into an ϵ-ball could be interpreted as a d_1 distance function in Eq. (2), that returns an infinite cost value for actions outside the ϵ-ball. Meanwhile, the cross-entropy loss subsumes the role of the d_2-cost function. Therefore, Eq. (8) can be considered as a special case of Eq. (2) transformed into a maximization problem.

We invite the reader to notice that the approaches presented above – whether adversarial attacks or counterfactual explanation methods – solve the same objective. In fact, they can be interpreted as heuristics to optimizing an instance of the formulation in Eq. (2), although pertaining to different "semantics" as argued in [55]. However, a precise distinction between counterfactual explanations and adversarial attack algorithms cannot be derived from their implementations. To this end, we investigate precise definitions for both problems in the next section.

4 Definitions

4.1 Formalizing Adversarials and Counterfactuals

We take the definition of an adversarial example by Freiesleben [23] as a starting point. It intuitively describes the properties that such instances should have. In other words, they should be close to the original instance, change the model's predictions and be misclassified. Most notably and in contrast to other works, Freiesleben argues that the misclassification is a distinctive property of adversarial examples. This distinctive property has also previously been mentioned in other works on adversarial examples more or less directly [56], giving rise to the following definition:

Definition 1 (Adversarial Example [23]). *An instance $\boldsymbol{x}' \in \mathcal{X}$ is an **adversarial example** for a factual $\boldsymbol{x} \in \mathcal{X}$ and a classifier $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ if the following conditions hold:*

(1) $\boldsymbol{x}'$ is close to $\boldsymbol{x}$, i.e., $d_1(\boldsymbol{x}, \boldsymbol{x}') < \epsilon$;
(2) the classifier output is changed, i.e., $f(\boldsymbol{x}) \neq f(\boldsymbol{x}')$;
(3) $\boldsymbol{x}'$ is misclassified, i.e., $y(\boldsymbol{x}') \neq f(\boldsymbol{x}')$.

[3] Thus, projected gradient ascent is often the more appropriate description for this attack. However, we will follow common practice and refer to the algorithm as PGD.

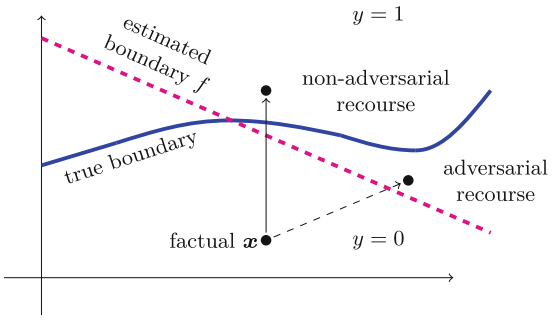


Fig. 2. Visualizing our definitions. The space of valid recourse for a factual $\boldsymbol{x}$ changes crosses the classifier f's estimated decision-boundary (pink). The experts combine it with their expertise and restrictions into a latent decision boundary (blue). However, some recourse might not change the true label and is therefore considered adversarial (dashed arrow). The challenge is to obtain recourse that convinces the human experts. To this end, we are interested in finding the directions that lead to *non-adversarial recourse* (solid arrow). (Color figure online)

We also consider the definition of recourse (or equivalently, counterfactual examples) by Freiesleben [23], which states that recourse $\boldsymbol{x}'$ changes the classification label and is the closest point to the factual that does so. We propose a slight relaxation. In particular, we argue that even points that are not closest to the factual are still valid (though possibly suboptimal) recourse.

Definition 2 (Recourse). *An instance $\boldsymbol{x}' \in \mathcal{X}$ is **recourse** for a factual $\boldsymbol{x} \in \mathcal{X}$ with $f(\boldsymbol{x}) \neq y_t$, a classifier $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$, and a target label $y_t \in \mathcal{Y}$ if the following conditions hold:*

(1) $\boldsymbol{x}'$ is close to $\boldsymbol{x}$, i.e., $d_1(\boldsymbol{x}, \boldsymbol{x}') < \epsilon$;
(2) the classifier output is changed to the target, i.e., $f(\boldsymbol{x}') = y_t \neq f(\boldsymbol{x})$.

These general definitions cover most definitions explicitly or implicitly used in the literature (see [23] for details). We immediately see that our definition of recourse abandons the final constraint in the definition of adversarial examples, that $\boldsymbol{x}'$ should be misclassified. For the two-class problem where y_t is just the opposite class of $f(\boldsymbol{x})$, according to these definitions, (a) all adversarial examples are recourse[4], and (b) there is a distinct (though potentially empty) subset of examples, that are recourse, but are not adversarials, as visualized in Fig. 2.

4.2 Non-adversarial Algorithmic Recourse

In this work, we place our attention on the examples present in this subset, that are recourse but not adversarial examples. We thus refer to them as *non-adversarial recourse* and introduce a novel definition for this class of instances:

[4] For multi-class problems, all adversarials which are classified as y_t are recourse.

Definition 3 (Non-adversarial Recourse). *An instance $\boldsymbol{x}' \in \mathcal{X}$ is **non-adversarial recourse** for a factual $\boldsymbol{x} \in \mathcal{X}$ with $f(\boldsymbol{x}) \neq y_t$, target label $y \in \mathcal{Y}$, and a classifier $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ if the following conditions hold:*

(1) $\boldsymbol{x}'$ is close to $\boldsymbol{x}$, i.e., $d_1(\boldsymbol{x}, \boldsymbol{x}') < \epsilon$;
(2) the classifier output is changed, i.e., $f(\boldsymbol{x}') = y_t \neq f(\boldsymbol{x})$;
(3) $\boldsymbol{x}'$ is not misclassified, $f(\boldsymbol{x}') = y(\boldsymbol{x}')$.

We observe that in the considered realistic decision-making scenario, we desire recourse that convinces the human experts, i.e., also changes the true label y. These correspond exactly to the instances described in the definition of non-adversarial recourse.

5 Theoretical Analysis

As outlined in Fig. 2, we are interested in finding changes, or at least directions of change, that lead to non-adversarial recourse efficiently. As it is impossible to precisely model the ground truth y in our setup (otherwise, there would be no need for an additional human expert), this is challenging in practice. However, we can use some guidance from the model, which approximates the ground truth, to find non-adversarial recourse.

5.1 Summarizing Influential Factors for Less Adversarial Recourse

We first take a step back and consider the general formulation of the problem given in Eq. (2). We observe that the problem formulation features three potential factors of influence (the model f, the distance functions d_1 and d_2) and a hyperparameter (the choice of optimization algorithm) that can be changed in practice to arrive at less adversarial recourse. If we follow the usual binary classification setup where we chose $\lambda > 0$ and d_2 to be the Dirac distance that amounts to infinity if the target label is not met, i.e., $d_2(f(\boldsymbol{x}'), y_t) = \delta_{f(\boldsymbol{x}')=y_t}$, there are three remaining factors of influence, that we tackle in this study with different outcomes (discussed in more detail in Sect. 6):

Machine Learning Model. Considering the model f first, we note that there is a simple theoretical solution to non-adversarial recourse: If the model would exactly match the theoretical ground truth, i.e., $f \equiv y$, there would be no adversarial recourse as every instance that leads to a different model prediction also changes the ground truth. However, in the setup we consider, it is impossible to perfectly learn y. Nevertheless, using the best possible model as close to the ground truth as possible should be fruitful. Another way to improve the model's alignment with the ground truth – in case the truth is known to be smooth in some measure – could be to potentially leverage regularization techniques such as adversarial training [38] to rule out many adversarial instances in the first place. **We empirically find that more accurate and robust models lead to less adversarial recourse.**

Input Space Distance Function. The distance function d_1 has been attributed a crucial role when computing recourse or adversarial examples. For instance, Wachter et al. [65] have claimed that, unlike recourse, *none of the standard works on adversarial perturbations use appropriate distance functions*. In this work, we follow the perspective of [6,23], who argue that the distance metric is not definitional but may still play an essential role in making recourse non-adversarial. Besides standard cost functions like p-norms such as the l_1, l_2, and l_∞, we are interested in how feature weightings may potentially impact recourse. We follow the intuition that some features are discriminative in the ground truth problem, e.g., income determines creditworthiness. However, ML models may rely on many more features, as the model designers cannot precisely specify a priori which features will be relevant for the task. When non-discriminative features are used in the task, they may open the door to adversarial changes as they can be picked up by an ML model regardless of their irrelevance w.r.t. the ground truth. In the next section, we will present an attempt to down-weigh the influence of such features by individually assigning a cost to each of them. In particular, we will consider distance functions of the form

$$d_{1,S}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \boldsymbol{\delta}^\top \boldsymbol{S} \boldsymbol{\delta}, \boldsymbol{S} = \operatorname{diag}(\boldsymbol{s}), \tag{9}$$

where $\boldsymbol{S} \in \mathbb{R}^{k \times k}$ is some diagonal matrix with diagonal $\boldsymbol{s} = [s_1, \ldots, s_k]^\top \in \mathbb{R}^k_{>0}$ and $\boldsymbol{\delta} \coloneqq \boldsymbol{x}' - \boldsymbol{x}$. For simple models analytical solutions of algorithmic recourse exist [46]. This allows to set up a nested optimization problem, where besides optimizing the recourse for a specific cost function, we find the cost function such that the resulting optimal recourse remains most non-adversarial. We will introduce the specific objective in the next section. We will see that the problem of finding optimal values for $\boldsymbol{s}$ can be solved analytically based on the gradients for linear models. **Surprisingly, we empirically find that the cost function does not play a key role in obtaining non-adversarial recourse.**

Optimization Routine. As the general problem is highly non-linear for complex models, it is hard to discover an optimal solution. As a result, algorithms to compute recourse or adversarial examples include different heuristic optimization routines such as stochastic gradient descent (deployed in SCFE, DICE, and C&W), gradient projection (deployed in PGD), or discretization (deployed in AR). The optimization procedure may thus also play a non-negligible role in determining whether the nature of the resulting recourse is adversarial and whether approaches designed for recourse yield fewer adversarial examples than their adversarial counterparts. **In this regard, we find that adversarial methods succeed to compute non-adversarial recourse, but also incur higher costs.**

5.2 Optimal Cost Functions Under Linear Models with Noisy Labels

In this section, we will restrict ourselves to the input space distance function d_1 and study its influence on the recourse from a theoretical standpoint.

We first introduce a measure to quantify the extent to which recourse is non-adversarial. To be able to do so, we consider the simplified setup where we have a feature set $\mathcal{F}$ and a subset of discriminative features $\mathcal{F}_{\text{disc}} \subset \mathcal{F}$ that contains relevant information affecting the ground truth. The remainder of the features are noise variables. Such features exist for many tasks; however, they may require a change of representation to be axis-aligned. For instance, in image generation models such as StyleGAN [32], the first latent variables control high-level concepts in the generation, whereas the later variables merely add noise that is unimportant for the classification output. Successes with dimensionality reduction techniques through autoencoding [30] also show that important information occupies only a subspace of tabular data. As outlined in Fig. 3, following the discriminative features is essential for obtaining non-adversarial recourse. We can quantify the share of the recourse that lies in the discriminative directions over the entire length of the recourse vector through the following measure.

Definition 4 (NADV measure). *Let $p \in \mathbb{N} \cup \{\infty\}$. The non-adversarialness measure $NADV_p$ is defined as*

$$NADV_p(\boldsymbol{\delta}) = \frac{\sum_{i \in \mathcal{F}_{disc}} |\boldsymbol{\delta}_i|}{\|\boldsymbol{\delta}\|_p}. \tag{10}$$

We consider linear models in our initial analysis, as they are the standard in many industries (e.g., in financial applications such as credit scoring [12]) and are commonly studied in the literature on algorithmic recourse [49,60]. They model a generative process of the form

$$y(\boldsymbol{x}) = \boldsymbol{\beta}^\top \boldsymbol{x} + \epsilon, \tag{11}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise of variance σ^2 and $\beta \in \mathbb{R}^k$ denotes the true linear parameter vector. Such a model can be easily adapted to a classification task by introducing a decision threshold, e.g., $y(\boldsymbol{x}) > 0$ indicates a positive outcome. As motivated in the introduction, the noise may represent uncertainty and variance in the human labels. We are interested in weightings s_i that minimize this measure, potentially leveraging the empirical coefficients $\hat{\boldsymbol{\beta}}$ obtained when fitting a linear model to the noisy data.

Theorem 1 (Optimal feature weights for recourse in linear models). *Suppose the data-generating process in Eq. (11) and that for $i \notin \mathcal{F}_{disc}$, we have $\beta_i = 0$, and for $i \in \mathcal{F}_{disc}$, $|\beta_i| > \alpha \in \mathbb{R}$. We can maximize the expected $NADV_p$ measure for $p \in \{1, 2, \infty\}$ when using the empirical coefficients $\hat{\beta}_i$ of the fitted model by setting the weights to*

$$s_i \sim \begin{cases} \left\{1, \ \textit{if } i = \arg\max_j \boldsymbol{p}_{disc}(\hat{\beta}_j), \ \textit{else } \infty\right\} & \textit{if } p = 1 \\ \frac{|\hat{\beta}_i|}{\boldsymbol{p}_{disc}(\hat{\beta}_i)} & \textit{if } p = 2 \\ |\hat{\beta}_i| & \textit{if } p = \infty \end{cases},$$

where $\boldsymbol{p}_{disc}(\hat{\beta}_i)$ is a probability of the feature being discriminative dependent on its empirical coefficient, which has a tractable sigmoidal form given in the Appendix.

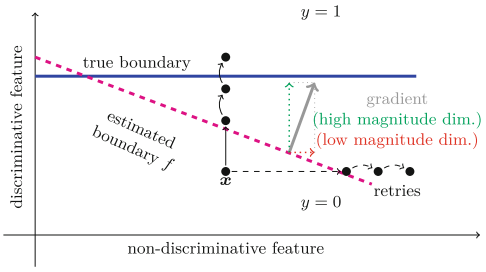


Fig. 3. Role of discriminative features in providing non-adversarial recourse. When features can be discriminative, (i.e., class-relevant) or non-discriminative (i.e., noise features), exploiting the discriminative ones will eventually lead to non-adversarial recourse, whereas solely relying on the non-discriminative ones will result in an adversarial. Nevertheless, even when selecting the correct features, several retry steps in the recourse direction may be required to cross the true decision boundary. To align recourse with discriminative features, the gradients of the model may serve as guidance, as we expect the discriminative dimensions to exhibit a higher gradient magnitude.

We provide a proof of this result in Appendix A. This finding highlights that in the case of discriminative and non-discriminative features in the data (even if they are not known), different loss functions affect which share of the recourse is awarded to the discriminative features. It also highlights the effect of the different norms. Optimizing the NADV_1 measure assigns infinite costs to all but the dimension that is most likely to be discriminative (with the highest absolute coefficient). On the other hand, the NADV_∞ measure is maximized if the discriminative features exhibit the maximum change of all features, disregarding changes in non-discriminative features. Therefore, the solution attempts to change all dimensions equally through assigning more discriminative dimensions a proportionally higher cost. This ensures that the less discriminative dimensions are altered as well. We observe that $p = 2$ seems to constitute a suitable trade-off, where dimensions with low probabilities of being discriminative ($\boldsymbol{p}_{\text{disc}}(\hat{\beta}_i) \approx 0$) are penalized by high costs, but the changes will otherwise be distributed evenly among the remaining dimensions.

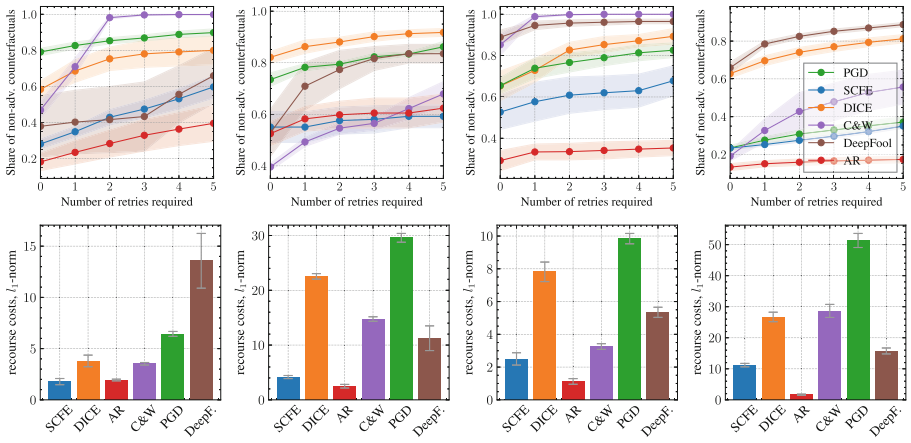
6 Experimental Evaluation

6.1 Experimental Setup

Datasets and Preprocessing. To link to the scenario considered in the introduction, we consider four tabular datasets concerned with high-stakes decision-making scenarios where human oversight may be required.

The Law School Admission data set[5] ("admission") contains information on students from law schools across the United States. Features are collected

[5] https://github.com/mkusner/counterfactual-fairness.



(a) Admission (b) German Credit (c) COMPAS (d) HELOC

Fig. 4. Both adversarial and recourse methods can succeed in producing non-adversarial recourse for ANNs. As it might not always be possible to change the ground truth immediately, we study the share of non-adversarial recourse instances after taking a certain number of retries r (a higher share is better). We experiment with three recourse methods (SCFE, DICE, AR) and three adversarial methods (C&W, PGD, DeepFool). Our results indicate that DICE and PGD usually perform best in identifying non-adversarial counterfactuals. The other adversarial methods, C&W and DeepFool, often outperform the standard recourse method SCFE regarding non-adversarial recourse. Note that recourse methods strictly optimize for the lowest costs and are therefore less robust than adversarial methods, which incur higher costs.

prior to their entry to law school and include race, sex, entrance exam scores (LSAT), grade-point average (GPA), and regional group. The predicted variable is the z-score of the first-year average grade (ZFYA). The German Credit dataset ("german") is taken from the UCI machine learning repository[6] and is concerned with credit scoring. It contains the personal data of 1000 individuals with a binary indicator named "credit risk" that serves as a prediction target. The Home Equity Line of Credit ("HELOC") data set[7] is a large collection of HELOC applications from anonymized homeowners collected by the financial services provider FICO. The target variable RiskPerformance is "Bad" if the applicant was at least 90 days past due within the two years after opening the credit account. The COMPAS data set[8] was initially collected by ProPublica and contains features describing criminal defendants in Broward County, Florida. It also contains their respective recidivism score provided by the COMPAS algorithm and whether or not they reoffended within the following two years. For our

[6] http://archive.ics.uci.edu/dataset/144/statlog+german+credit+data.
[7] https://community.fico.com/s/explainable-machine-learning-challenge.
[8] https://www.kaggle.com/s/danofer/compass.

analysis, we only kept features relevant for predicting recidivism within the next two years and dropped irrelevant features such as name, date, sex, and race.

For all datasets, continuous features are standardized. Datasets with continuous labels are used in a binary classification fashion where we only predict if the z-score exceeds the population's median.

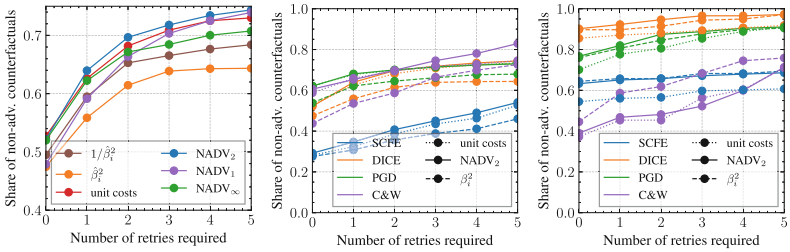
Machine Learning Models. We use standard Artificial Neural Networks (ANN) that reflect the implementation of the `sklearn` library but are implemented in the `PyTorch` library to leverage automated differentiation capabilities. We train the ANN model (two fully connected hidden layers of width 30) using stochastic gradient descent with the ADAM optimizer. An overview over implementation parameters is provided in the Appendix.

Adversarial Attacks and Recourse Algorithms. We implement three powerful adversarial attacks and three recourse methods to study the problem from a practical perspective. We stick to the methods introduced earlier, which include `SCFE` [65], which uses a gradient-based objective to find recourses, `DICE` [42] with an extra diversity constraint, and `AR` [61], which uses a Mixed-Integer-Program on a discretized action set. Regarding the adversarial attacks, we use `C&W` [7] that finds the minimum perturbation on the factual instance to make it change class, `PGD` [38] that uses projected gradients to engender adversarials, and `DeepFool` [40] that perturbs the input iteratively until the class changes. We adapt the cost function of each optimization algorithm to reflect Eq. (2) and plug in the different cost functions.

Ground Truth. Unfortunately, the number of instances with labels on real-world data sets is limited, such that the ground truth function y is not explicitly available. We, therefore, rely on a simulated ground truth, which uses a subset of the training data that will not be used for model training or testing. We use this data to construct a k nearest neighbor classifier (with $k = 5$) that uses a subset of features to simulate an expert committee relying on discriminative features and deciding by majority vote. We manually select features that can be considered directly discriminative for the task, which are listed in Table 2. For instance on the COMPAS dataset, we use features such as the number of priors, and whether recividism has occurred in the last two year. By doing so, we can guarantee that we have discriminative and non-discriminative features. We then use this ground truth y to predict the remaining instances of the train set. Subsequently, the actual ML model is trained on the remainder of the data and their predictions, making up tuples of the form $(\boldsymbol{x}, y(\boldsymbol{x}))$.

Evaluation Measures. Many recourse (and adversarial) methods are implemented to stop right after the model's boundary is crossed. However, this might not initially lead to the non-adversarial recourse desired in practice, even if the correct discriminative features are manipulated (see Fig. 3 for an illustration). We argue that in the practical use case, an individual would query the oracle (e.g., submit their application to the bank again) after obtaining recourse. If the recourse was ineffective in changing the loan decision, an individual could continue to move in the given direction (e.g., further increase their savings amount)

410 T. Leemann et al.



(a) all cost functions (b) Admission (c) German Credit

Fig. 5. Cost functions can play a role in generating non-adversarial recourse. (a) "admission" dataset with ANN model, DICE results shown. (b,c): Our NADV$_2$ cost function helps in making recourse slightly less adversarial for several and thereby reduces the number of retries required. However, analysing the standard deviations does *not* confirm statistical significance.

until the loan is eventually awarded. We mimic this setup, by increasing the magnitude of $\boldsymbol{\delta} = \boldsymbol{x}' - \boldsymbol{x}$ by 10% in each step, thus considering $\boldsymbol{x}'_r = \boldsymbol{x} + (1.1^r)\boldsymbol{\delta}$ after $r \geq 0$ retries. We additionally consider the canonical recourse costs in the l_1 and l_2 norm.

6.2 Choice of Optimization Algorithm

We first put all six implemented methods to the test and check the adversarialness of their outputs. The results are visualized in Fig. 4. We consider the initial recourse and up to 5 more steps in the initial direction. We observe that DICE and PGD usually perform best in identifying non-adversarial counterfactuals. However, the other adversarial methods, C&W and DeepFool, also often outperform the classical recourse method SCFE regarding non-adversarial recourse. This underlines that, for tabular data, the methods do not reliably produce adversarials. Indeed, they could be considered as recourse methods as well. However, we observe that the adversarial techniques usually result in higher costs, because returning an optimal solution is not their main concern (it just needs to be "close" to the input). In contrast, many recourse methods are designed to provide cost-optimal solutions. Non-adversarial recourse is associated with higher cost, leading us to believe that classical recourse methods may be overly cost sensitive for this purpose. We obtained similar results using L2-costs.

6.3 Choice of Cost Function

We now study the different cost functions derived in Sect. 5.2 to actual implementations of both recourse and adversarial methods on real data. In particular, we compute the gradients of the model and use the cost weightings derived earlier as well as the default l_2-costs, squared gradient costs (β_i^2, should assign low cost to non-discriminative features) and inverse squared costs ($1/\beta_i^2$) as baselines. DeepFool and AR do not allow for the simple, straightforward inclusion

of arbitrary cost functions, so we only consider the four remaining approaches for this experiment and modify their cost-function. The results are shown in Fig. 5. They show that cost weighting can steer the recourses towards the non-adversarial features and align them better with the ground truth. However, in Fig. 5a, the differences remain statistically insignificant. We observe that the NADV_2 optimal weighting scores best among all costs. Inversely weighting the features (e.g., $s_i = \hat{\beta}_i^2$, which assigns low costs to features with almost zero gradients and high costs to features with high gradients), preventing them from being changed, results in the most adversarial recourse. Even though the gap is small, the improvement seems stable across methods (see Fig. 5b, c) with one exception (C&W on German Credit). In conclusion, while the cost function can help to make recourse less adversarial, its effect seems to be rather subtle.

6.4 Choice of Machine Learning Model

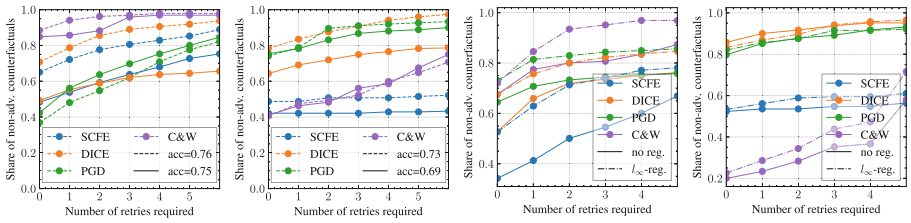
In our analysis section, we outlined how the machine learning (ML) model may be crucial in determining whether the outcomes can be considered adversarial. We first study the role of the goodness of the model fit. To this end, we train a model on a version of the dataset, where a random sample of 25% of the data points have flipped labels, which could reflect a realistic use case with noisy human annotations. To rule out other confounding effects to the convexity or smoothness of the model's decision boundary (models trained on noisy labels may have very sharp and more non-convex decision boundaries), we study logistic regression models in this experiment and report the results in Fig. 6 (a, b). Surprisingly, the drop in accuracy is not very high (it remains in a range of 1.5% to 5%), which we attribute to the datasets being already very noisy previously. Nevertheless, we observe a clear tendency for recourses to be less adversarial for the more accurate models. This trend is stable across datasets and methods.

Adversarial training was proposed by Madry et al. [38] to make models more robust against adversarial attacks. Therefore, it might also offer a suitable way of mitigating adversarial examples in the recourse setup. We study the effect of this form of regularization in an l_∞-ball of radius $\epsilon = 0.2$ in Fig. 6 (c, d). We observe that substantial improvements are possible on the Admission dataset. They are not as pronounced for the remaining datasets but remain visible for most methods. We observe comparable results for the remaining two datasets. Our results highlight that maintaining robust and accurate models is one of the most promising strategies towards non-adversarial recourse.

7 Discussion

Adversarial Methods Compute Recourse on Tabular Data. Intriguingly, we observe that despite their purpose, many adversarial attacks succeed in computing non-adversarial recourse on tabular data. While many of the methods were arguably designed with other data modalities, e.g., images, in mind, our finding raises the question of how transferable existing attacks are to variants of the canonical attack scenario. This observation is one in a series of recent claims

412 T. Leemann et al.



(a) Admission (b) German Credit (c) Admission (d) German Credit

Fig. 6. (a, b): More accurate models lead to less adversarial recourse. We plot the number of retries required to obtain a valid, non-adversarial recourse that changes the ground truth. Logistic Regression Model shown. Results on the remaining datasets can be found in the extended Appendix (https://arxiv.org/abs/2403.10330). **(c, d): Regularization through Adversarial Training may improve non-adversarialness**. We robustify models through adversarial training, which improves the share of non-adversarial recourses.

suggesting that current adversarial attacks may not be realistic in the majority of practical use cases [4] or require a fundamental paradigm shift away from norms as cost functions towards realistic measures of detector evasion [15].

An Implicit Pursuit Towards Non-adversarial Recourse. The recourse literature suggests several strategies for improving the quality of recourse. Kommiya et al. [33] discovered that feature attributions and feature modifications in recourses only partially agree, raising the question of how they can potentially be used as guidance. Recent takes on robustifying recourse by going further than mandated by the actual decision boundary [48,60] can be interpreted as another take to reduce the possibility of ending up with an adversarial. Therefore, we conclude that these works seem to have implicitly followed the goal of obtaining non-adversarial recourse and can be interpreted as orthogonal attempts to reach this common goal. We hope that our precise definition of non-adversarial recourse allows for these efforts to be bundled and unified in the future.

Non-adversarial Recourse via Distributional Constraints. Another avenue we have not followed in this work considers the feasible set. The feasible set $\mathcal{X}$ many works have claimed that recourse should be actionable, leading to realistic instances [50,61]. A fairly general way to arrive at this goal is to constrain the recourse to be in-distribution [17,31,47], which can be seen as another strategy towards non-adversarial recourse: For in-distribution examples, every model that is a suitable approximation of the ground truth should result in an above-chance-level agreement between the model and the ground truth. We leave an investigation of this connection to future work.

8 Conclusion

In this work, we explored the nuanced differences between adversarial examples and counterfactual explanations, focusing on real-world high-stakes decision-

making processes. For such scenarios, we introduced the desirable concept of non-adversarial recourse, emphasizing that useful counterfactual explanations should not only change the model's prediction but also align with the ground truth in contrast to adversarial examples.

Our theoretical and experimental analyses on multiple real-world datasets illuminate different ways the model parameters can shape the generation of non-adversarial recourse. Our findings suggest that choosing a suitable model that is highly accurate and robust has more impact on whether recourse can be considered adversarial than the choice of the cost function. For tabular data, adversarial methods also succeed in computing suitable recourse. In summary, we provided valuable insights into generating counterfactuals of reduced adversarialness. Hence, this work lays a foundation for developing resilient recourse models and their deployment in realistic decision-making scenarios.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A Derivation of Theorem V.I

This section presents the proof of Theorem 1 proof. First, we show how the probability of a relevant feature can be easily estimated in linear models. Suppose we have obtained a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times k}$. Then, we can obtain the analytical least-squares solution $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$. We can estimate the variance of $\hat{\boldsymbol{\beta}}$ to be $\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1}$. Simplifying through assuming the features in $\boldsymbol{x}$ to be independent and of zero-mean, $\boldsymbol{X}^\top \boldsymbol{X}$ is diagonal and we obtain

$$\mathrm{Var}[\hat{\beta}_i] = \frac{\sigma^2}{\sum_{j=1\ldots n} (\boldsymbol{x}_j)_i^2}. \tag{12}$$

This allows to use of the estimated coefficients to estimate the probability of a feature being relevant, $\boldsymbol{p}_{\text{disc}}$ through the following derivation:

$$\begin{aligned}
\boldsymbol{p}_{\text{disc}}(\hat{\beta}_i) &= \boldsymbol{p}(i \in \mathcal{F}_{\text{disc}} | \hat{\beta}_i) && (13)\\
&= \frac{\boldsymbol{p}(\hat{\beta}_i, i \in \mathcal{F}_{\text{disc}})}{\boldsymbol{p}(\hat{\beta}_i, i \in \mathcal{F}_{\text{disc}}) + \boldsymbol{p}(\hat{\beta}_i, i \notin \mathcal{F}_{\text{disc}})} && (14)\\
&= \frac{\boldsymbol{p}(\hat{\beta}_i | i \in \mathcal{F}_{\text{disc}})}{\boldsymbol{p}(\hat{\beta}_i | i \in \mathcal{F}_{\text{disc}}) + \boldsymbol{p}(\hat{\beta}_i | i \notin \mathcal{F}_{\text{disc}}) \underbrace{\frac{\boldsymbol{p}(i \notin \mathcal{F}_{\text{disc}})}{\boldsymbol{p}(i \in \mathcal{F}_{\text{disc}})}}_{q}} && (15)\\
&= \frac{1}{1 + \frac{\boldsymbol{p}(\hat{\beta}_i | i \in \mathcal{F}_{\text{disc}})}{q \cdot \boldsymbol{p}(\hat{\beta}_i | i \notin \mathcal{F}_{\text{disc}})}} \geq \frac{1}{1 + \exp\left(\alpha^2 - 2\alpha |\hat{\beta}_i| - \log q\right)} && (16)\\
&= \text{sigmoid}\left(2\alpha |\hat{\beta}_i| - \alpha^2 + \log q\right). && (17)
\end{aligned}$$

The above calculation highlights that it is possible to use the coefficients $\hat{\boldsymbol{\beta}}$ in the linear model as noisy estimates for assessing whether a feature is discriminative.

We combine this insight with the optimal recourse found using a specific cost matrix $\boldsymbol{S}$. To this end, we leverage the analytical solution to this problem [9, Lemma 4, Appendix]:

$$\boldsymbol{\delta}(\boldsymbol{S}) = \underbrace{\frac{f(\boldsymbol{x}) - y_t}{\hat{\boldsymbol{\beta}}^\top \boldsymbol{S}^{-1} \hat{\boldsymbol{\beta}}}}_{c} \boldsymbol{S}^{-1} \hat{\boldsymbol{\beta}}. \tag{18}$$

We can then compute the expected value of the measure of non-adversarialness for the recourse that will be found with the corresponding cost function:

$$\mathbb{E}_{\hat{\boldsymbol{\beta}}}\left[\mathrm{NADV}_p(\boldsymbol{S})\right] = \mathbb{E}_{\hat{\boldsymbol{\beta}}}\left[\frac{\sum_{i \in \mathcal{F}_{\text{disc}}} |\boldsymbol{\delta}_i|}{\|\boldsymbol{\delta}\|_p}\right] = \mathbb{E}_{\hat{\boldsymbol{\beta}}}\left[\frac{\sum_{i \in \mathcal{F}_{\text{disc}}} |\frac{\hat{\beta}_i}{s_i}|}{\|\boldsymbol{S}^{-1}\hat{\boldsymbol{\beta}}\|_p}\right] \tag{19}$$

$$= \frac{\sum_i p_{\text{disc},i}(\hat{\boldsymbol{\beta}}) |\frac{\hat{\beta}_i}{s_i}|}{\|\boldsymbol{S}^{-1}\hat{\boldsymbol{\beta}}\|_p} = \frac{\boldsymbol{p}_{\text{disc}}^\top(\hat{\boldsymbol{\beta}})(\boldsymbol{S}^{-1}|\hat{\boldsymbol{\beta}}|)}{\|\boldsymbol{S}^{-1}\hat{\boldsymbol{\beta}}\|_p} \tag{20}$$

$$= \frac{\boldsymbol{p}_{\text{disc}}^\top(\hat{\boldsymbol{\beta}})(\boldsymbol{S}^{-1}|\hat{\boldsymbol{\beta}}|)}{\|\boldsymbol{S}^{-1}|\hat{\boldsymbol{\beta}}|\|_p} \tag{21}$$

Taking the above expression, we can obtain optimal costs for different values of p by solving

$$\arg\max \mathbb{E}_{\hat{\boldsymbol{\beta}}}\left[\mathrm{NADV}_p(\boldsymbol{S})\right]. \tag{22}$$

Continuing the calculation separately for the most common values $p \in \{1, 2, \infty\}$, we obtain the following cost weights s_i that depend on the estimated $\hat{\beta}_i$:

$p = 1$	$p = 2$	$p = \infty$
implicit $\boldsymbol{S}^{-1}\lvert\hat{\boldsymbol{\beta}}\rvert = \kappa \boldsymbol{e}_{\arg\max_i \ \boldsymbol{p}_{\text{disc}}(\hat{\beta}_i)}$	$\boldsymbol{S}^{-1}\lvert\hat{\boldsymbol{\beta}}\rvert = \kappa \frac{\boldsymbol{p}_{\text{disc}}(\hat{\boldsymbol{\beta}})}{\|\boldsymbol{p}_{\text{disc}}(\hat{\boldsymbol{\beta}})\|_2}$	$\boldsymbol{S}^{-1}\lvert\hat{\boldsymbol{\beta}}\rvert = \kappa \mathbf{1}$
explicit $s_i \sim \left\{1, \text{ if } i = \arg\max_j \ \boldsymbol{p}_{\text{disc}}(\hat{\beta}_j), \text{ else } \infty\right\}$	$s_i \sim \frac{\lvert\hat{\beta}_i\rvert}{\boldsymbol{p}_{\text{disc}}(\hat{\beta}_i)}$	$s_i \sim \lvert\hat{\beta}_i\rvert$

B Experimental Details

We use the following experimental parameters (Table 1):

Table 1. Implementation parameters

		Artificial Neural Network	Logistic Regression
Config.	Units	[Input dim., 30, 30, 2]	[Input dim., 1]
	Intermediate activations	ReLU	N/A
	Last layer activations	None	Sigmoid
Training	Learning rate	10^{-3}	N/A
	Regularization	None	l_2 with pen = 1
	Batch size	32	N/A
	Epochs	10^3	5×10^3

Method	Optimizer	lr	Iterations	λ	Additional Comments
SCFE	Adam	10^{-1}	100	0.1	step = 0
DiCE	RMSProp	10^{-1}	100	-	Two counterfactuals, one is randomly sampled for evaluation
AR	Default as in [61]	-	-	-	Squared loss in cost function
C&W	Gradient-based as in [7]	10^{-2}	1000	-	Constant factor $c = 1$
DeepFool	-	-	50	2×10^{-2}	Target label for attack directionality [40]
PGD	-	10^{-1}	10	10^{-1}	$\alpha = 10^{-1}$, $\varepsilon = 2$

Table 2. Features that are used by the experts (GT) and total number of features available to adversarial methods and recourse methods on each dataset.

Dataset	GT Features	Tot. features
Admission	ugpa, first_pf	4
German Credit	status, credit-history, employment-duration, housing, number-credits	19
COMPAS	age, two_year_recid, priors_count	5
HELOC	MSinceMostRecentTradeOpen, NumTrades60Ever2DerogPubRec, NumTrades90Ever2DerogPubRec, NumTradesOpeninLast12M, NumInqLast6M, NumInqLast6Mexcl7days, NumRevolvingTradesWBalance, NumInstallTradesWBalance, NumBank2NatlTradesWHighUtilization	22

References

1. Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.: Trends and trajectories for explainable, accountable and intelligible systems: an HCI research agenda. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–18 (2018)
2. Abrate, C., Bonchi, F.: Counterfactual graphs for explainable classification of brain networks. In: KDD (2021)
3. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access **6**, 14410–14430 (2018)
4. Apruzzese, G., Anderson, H.S., Dambra, S., Freeman, D., Pierazzi, F., Roundy, K.: "real attackers don't compute gradients": Bridging the gap between adversarial ml research and practice. In: 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), pp. 339–364. IEEE (2023)
5. Baluja, S., Fischer, I.: Adversarial transformation networks: learning to generate adversarial examples. arXiv preprint arXiv:1703.09387 (2017)
6. Browne, K., Swift, B.: Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. arXiv preprint arXiv:2012.10076 (2020)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)
8. Carreira-Perpiñán, M.Á., Hada, S.S.: Counterfactual explanations for oblique decision trees: exact, efficient algorithms. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 6903–6911 (2021)
9. Chen, Y., Wang, J., Liu, Y.: Strategic recourse in linear classification. arXiv preprint arXiv:2011.00355 **236** (2020)
10. Cheng, J., Danescu-Niculescu-Mizil, C., Leskovec, J.: Antisocial behavior in online discussion communities. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 9, pp. 61–70 (2015)
11. Croce, F., Hein, M.: Sparse and imperceivable adversarial attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4724–4732 (2019)
12. Dastile, X., Celik, T., Potsane, M.: Statistical and machine learning models in credit scoring: a systematic literature survey. Appl. Soft Comput. **91**, 106263 (2020)
13. De, A., Koley, P., Ganguly, N., Gomez-Rodriguez, M.: Regression under human assistance. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2611–2620 (2020)
14. De, A., Okati, N., Zarezade, A., Rodriguez, M.G.: Classification under human assistance. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 5905–5913 (2021)
15. Debenedetti, E., Carlini, N., Tramèr, F.: Evading black-box classifiers without breaking eggs. arXiv preprint arXiv:2306.02895 (2023)
16. Demir, U., Ünal, G.B.: Patch-based image inpainting with generative adversarial networks. CoRR abs/1803.07422 (2018). http://arxiv.org/abs/1803.07422
17. Dhurandhar, A., et al.: Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
18. Dominguez-Olmedo, R., Karimi, A.H., Schölkopf, B.: On the adversarial robustness of causal algorithmic recourse. In: Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 162, pp. 5324–5342. PMLR (2022)

19. Du, A., et al.: Physical adversarial attacks on an aerial imagery object detector. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1796–1806 (2022)
20. Duan, R., et al.: Adversarial laser beam: effective physical-world attack to DNNs in a blink. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16062–16071 (2021)
21. Ferreira, J.J., de Souza Monteiro, M.: The human-AI relationship in decision-making: AI explanation to support people on justifying their decisions. In: Joint Proceedings of the ACM IUI 2021 Workshops, vol. 2903 (2021)
22. Freiesleben, T.: Counterfactual explanations & adversarial examples–common grounds, essential differences, and potential transfers. arXiv preprint arXiv:2009.05487 (2020)
23. Freiesleben, T.: The intriguing relation between counterfactual explanations and adversarial examples. Mind. Mach. **32**(1), 77–109 (2022)
24. Garcia, L., Brasser, F., Cintuglu, M.H., Sadeghi, A.R., Mohammed, O.A., Zonouz, S.A.: Hey, my malware knows physics! attacking PLCs with physical model aware rootkit. In: NDSS, pp. 1–15 (2017)
25. GDPR: Regulation (EU) 2016/679 of the European parliament and of the council. Off. J. Eur. Union (2016)
26. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
27. Grudin, J.: AI and HCI: two fields divided by a common focus. AI Mag. **30**(4), 48 (2009). https://doi.org/10.1609/aimag.v30i4.2271, https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2271
28. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Min. Knowl. Discov. 1–55 (2022)
29. Heath, D., Kasif, S., Salzberg, S.: Induction of oblique decision trees. In: IJCAI, vol. 1993, pp. 1002–1007. Citeseer (1993)
30. Ilkhechi, A., et al.: DeepSqueeze: deep semantic compression for tabular data. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 1733–1746 (2020)
31. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint arXiv:1907.09615 (2019)
32. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of styleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
33. Kommiya Mothilal, R., Mahajan, D., Tan, C., Sharma, A.: Towards unifying feature attribution and counterfactual explanations: different means to the same end. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pp. 652–663 (2021)
34. König, G., Freiesleben, T., Grosse-Wentrup, M.: Improvement-focused causal recourse (ICR). In: AAAI Conference on Artificial Intelligence (2023)
35. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world (2016)
36. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: unjustified counterfactual explanations. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19) (2019)

37. Ma, J., Guo, R., Mishra, S., Zhang, A., Li, J.: Clear: generative counterfactual explanations on graphs. In: Advances in Neural Information Processing Systems, vol. 35, pp. 25895–25907 (2022)
38. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
39. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1765–1773 (2017)
40. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
41. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 607–617 (2020)
42. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*) (2020)
43. Mozannar, H., Sontag, D.: Consistent estimators for learning to defer to an expert. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 7076–7087 (2020)
44. Narodytska, N., Kasiviswanathan, S.P.: Simple black-box adversarial perturbations for deep networks. arXiv preprint arXiv:1612.06299 (2016)
45. Pauwels, E.: How to protect biotechnology and biosecurity from adversarial AI attacks? A global governance perspective. In: Greenbaum, D. (ed.) Cyberbiosecurity, pp. 173–184. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-26034-6_11
46. Pawelczyk, M., Agarwal, C., Joshi, S., Upadhyay, S., Lakkaraju, H.: Exploring counterfactual explanations through the lens of adversarial examples: a theoretical and empirical analysis. In: International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 4574–4594. PMLR (2022)
47. Pawelczyk, M., Broelemann, K., Kasneci, G.: Learning model-agnostic counterfactual explanations for tabular data. In: Proceedings of The Web Conference 2020 (WWW). ACM (2020)
48. Pawelczyk, M., Datta, T., den Heuvel, J.V., Kasneci, G., Lakkaraju, H.: Probabilistically robust recourse: navigating the trade-offs between costs and robustness in algorithmic recourse. In: The Eleventh International Conference on Learning Representations (ICLR) (2023)
49. Pawelczyk, M., Leemann, T., Biega, A., Kasneci, G.: On the trade-off between actionable explanations and the right to be forgotten. In: The Eleventh International Conference on Learning Representations (ICLR) (2023)
50. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 344–350 (2020)
51. Pradel, M., Sen, K.: DeepBugs: a learning approach to name-based bug detection. Proc. ACM Program. Lang. **2**(OOPSLA), 1–25 (2018)
52. Prado-Romero, M.A., Prenkaj, B., Stilo, G., Giannotti, F.: A survey on graph counterfactual explanations: definitions, methods, evaluation, and research challenges. ACM Comput. Surv. (2023)
53. Raghu, M., Blumer, K., Corrado, G., Kleinberg, J., Obermeyer, Z., Mullainathan, S.: The algorithmic automation problem: prediction, triage, and human effort. arXiv preprint arXiv:1903.12220 (2019)

54. Rawal, K., Lakkaraju, H.: Beyond individualized recourse: interpretable and interactive summaries of actionable recourses. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12187–12198 (2020)
55. Sahil, V., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: a review (2010)
56. Stutz, D., Hein, M., Schiele, B.: Disentangling adversarial robustness and generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6976–6987 (2019)
57. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. **23**(5), 828–841 (2019)
58. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
59. Topol, E.J.: High-performance medicine: the convergence of human and artificial intelligence. Nat. Med. **25**(1), 44–56 (2019)
60. Upadhyay, S., Joshi, S., Lakkaraju, H.: Towards robust and reliable algorithmic recourse. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 34 (2021)
61. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*) (2019)
62. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: Proceedings of the Conference on Fairness, Accountability, and Transparency (2019). https://doi.org/10.1145/3287560.3287566
63. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: a review. arXiv:2010.10596 (2020)
64. Voigt, P., Von dem Bussche, A.: The EU general data protection regulation (GDPR). In: A Practical Guide, 1st edn. Springer, Cham (2017). **10**, 3152676
65. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. J. Law Technol. **31**(2) (2018)
66. Zhang, J., Lou, Y., Wang, J., Wu, K., Lu, K., Jia, X.: Evaluating adversarial attacks on driving safety in vision-based autonomous vehicles. IEEE Internet Things J. **9**(5), 3443–3456 (2022). https://doi.org/10.1109/JIOT.2021.3099164
67. Zhao, G., Zhang, M., Liu, J., Li, Y., Wen, J.R.: AP-GAN: adversarial patch attack on content-based image retrieval systems. GeoInformatica, 1–31 (2022)

3.3 High-Fidelity Explanations for Transformers

Publication 3

Tobias Leemann, Alina Fastowski, Felix Pfeiffer, and Gjergji Kasneci: Attention Mechanisms Don't Learn Additive Models: Rethinking Feature Importance for Transformers. *Transactions on Machine Learning Research (TMLR)*, 2024.

Author Contributions. I developed the initial idea for this project together with Gjergji Kasneci. Initial experimentation was done by Felix Pfeiffer, who created the main part of the initial code base. Alina Fastowski contributed an experiment on the Yelp-HAT dataset to assess the alignment with human explanations. I wrote the final manuscript with significant help from Gjergji Kasneci, who contributed valuable improvements to the presentation of the results. In particular, he helped improve the clarity of the theoretical results by dissecting them into smaller corollaries with nuanced discussion.

Summary. The transformer architecture (Vaswani et al., 2017) has set the standard in many domains and become the foundation of modern Large Language Models (LLMs). The inputs for these models are usually represented as a token sequence. This makes it increasingly important to better understand the relation between input tokens and the output for these models. In this work, we both theoretically and empirically investigate the effect of applying classical model-agnostic surrogate models such as LIME (Ribeiro et al., 2016) to transformer models. Surrogate model explanations offer the advantage that they directly describe the predictive model's behavior around a specific instance. They should have high fidelity, i.e., approximate the behavior of the predictive model well in a local neighborhood. We observe that linear surrogate models are popular and most attribution methods, e.g., SHAP and LRP, give rise to an implicit linear model. This canonical interpretation states that when a feature i is assigned an attribution of ϕ_i, removing the feature should roughly decrease the output score by ϕ_i. However, we prove that transformer models strikingly struggle to represent simple linear functions even with multiple layers. This is due to the attention-mechanism's softmax-normalization, which introduces dependencies over the entire sequence length. Our work highlights a fundamental misalignment between the transformer as a predictive model and popular surrogate models resulting in low fidelity. For improved fidelity we propose the Softmax-Linked Additive Log-Odds Model (SLALOM), a surrogate model that can be easily represented by transformers and is better aligned with the transformer architecture. We prove that SLALOM is easier to learn for transformers

and can be efficiently estimated. Experiments on synthetic and real datasets confirm our theoretical analysis, highlighting that SLALOM results in explanations with higher fidelity. SLALOM quantifies the contribution of each token through two values instead of one: a token *value score* (describing its independent contribution) and an *importance score* (describing its interaction weight).

3.3.1 Discussion

This work shows that the classical linear interpretation of feature attributions does not work well for transformers. Our work theoretically outlines why transformers struggle or may even not be able to represent such a model at all. Thereby linear explanations may only describe the model at an exact point (similar to a derivative for a function), but do not generalize to a neighborhood around the instance that is being explained. This renders the explanations incapable of quantitatively describing the behavior of the model for practical changes such as token insertions and removal, which are discrete and non-infinitesimal. We argue that such grounding is essential for auditing black-box explanations. In a black-box setting, explanations can only be audited by submitting perturbed inputs and checking the resulting outputs against the explanation, e.g., using the infidelity metric by Yeh et al. (2019). As tokens are discrete, the valid area for the explanation needs to cover complete removals, and hyper-local linear approximations are insufficient for auditing and may further open the door to attacks such as "fairwashing" (Slack et al., 2020). Concerning the user perspective, we find that our explanations align better with human perception, when we run an experiment about identifying the most essential tokens. SLALOM has been successfully applied to explain online trauma detection in follow-up work (Schirmer et al., 2024). We underline that there is no one-fits-all explanation and encourage further work on the compatibility of XAI methods and ML models, but also on multi-dimensional attribution methods that quantify feature importance by more than one dimension.

Published in Transactions on Machine Learning Research (12/2024)

Attention Mechanisms Don't Learn Additive Models: Rethinking Feature Importance for Transformers

Tobias Leemann *University of Tübingen, Technical University of Munich* *tobias.leemann@uni-tuebingen.de*

Alina Fastowski *Technical University of Munich* *alina.fastowski@tum.de*

Felix Pfeiffer *University of Tübingen* *felix.pfeiffer@protonmail.com*

Gjergji Kasneci *Technical University of Munich* *gjergji.kasneci@tum.de*

Reviewed on OpenReview: *https://openreview.net/forum?id=yawWz4qWkF*

Abstract

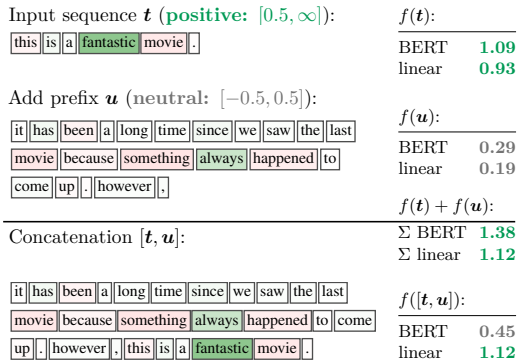
We address the critical challenge of applying feature attribution methods to the transformer architecture, which dominates current applications in natural language processing and beyond. Traditional attribution methods to explainable AI (XAI) explicitly or implicitly rely on linear or additive surrogate models to quantify the impact of input features on a model's output. In this work, we formally prove an alarming incompatibility: transformers are structurally incapable of representing linear or additive surrogate models used for feature attribution, undermining the grounding of these conventional explanation methodologies. To address this discrepancy, we introduce the Softmax-Linked Additive Log Odds Model (SLALOM), a novel surrogate model specifically designed to align with the transformer framework. SLALOM demonstrates the capacity to deliver a range of insightful explanations with both synthetic and real-world datasets. We highlight SLALOM's unique efficiency-quality curve by showing that SLALOM can produce explanations with substantially higher fidelity than competing surrogate models or provide explanations of comparable quality at a fraction of their computational costs. We release code for SLALOM as an open-source project online at `https://github.com/tleemann/slalom_explanations`.

1 Introduction

The transformer architecture (Vaswani et al., 2017) has been established as the status quo in modern natural language processing (Devlin et al., 2018; Radford et al., 2018; 2019; Touvron et al., 2023). However, the current and foreseeable adoption of large language models (LLMs) in critical domains such as the judicial system (Chalkidis et al., 2019) and the medical domain (Jeblick et al., 2023) comes with an increased need for transparency and interpretability. Methods to enhance the interpretability of an artificial intelligence (AI) system are developed in the research area of Explainable AI (XAI, Adadi & Berrada, 2018; Gilpin et al., 2018; Molnar, 2019; Burkart & Huber, 2021). A recent meta-study (Rong et al., 2023) shows that XAI has the potential to increase users' understanding of AI systems and their trust therein. Local feature attribution methods that quantify the contribution of each input to a decision outcome are among the most popular explanation methods, and a variety of approaches have been suggested for the task of computing such attributions (Kasneci & Gottron, 2016; Ribeiro et al., 2016; Sundararajan et al., 2017; Lundberg & Lee, 2017; Covert et al., 2021; Modarressi et al., 2022).

It remains hard to formally define the contribution of an input feature for non-linear functions. Recent work (Han et al., 2022) has shown that common explanation methods do so by implicitly or explicitly performing a local approximation of the complex black-box function, denoted as f, using a simpler surrogate function g from a predefined class $\mathcal{G}$. For instance, Local Interpretable Model-agnostic Explanations (LIME, Ribeiro

Published in Transactions on Machine Learning Research (12/2024)



Common Surrogate Models in XAI

Predictive Models	Surrogate Models	Explanation Techniques
Logistic / Linear Regression, ReLU networks	Linear Model: ✗no non-linearities ✗no interactions	C-LIME, Gradients, IG
GAM, ensembles, boosting	GAM: ✓non-linearities ✗no interactions	Removal-based, e.g. Shapley Values, Local GAM approximation
Transformers	**SLALOM (ours): ✓non-linearities ✓interactions**	**Local SLALOM approximation**

Figure 1: **Transformers cannot be well explained through additive models.** Left: We exemplarily show the log odds for the outputs of a BERT model and a linear Naïve-Bayes model ("linear") assigning each word a weight trained on the IMDB movie review dataset. The token colors indicate the weights assigned by the linear model. We pass two sequences to the models independently and in concatenation. For the linear model, the output of the concatenated sequence can be described by the sum, but this is not the case for BERT. We show that this phenomenon is not due to a non-linearity in this particular model but stems from a general incapacity of transformers to represent additive functions. Right: To overcome this difficulty, we propose SLALOM, a novel surrogate model specifically designed to better approximate transformer models.

et al., 2016) or input gradient explanations (Baehrens et al., 2010) use a linear surrogate model to approximate the black-box f; the model's coefficients can be used as the feature contributions.

Surrogate model explanations have the advantage that they directly describe the behavior of the model in the proximity of a specific input, i.e., under small perturbations, a property known as *fidelity* (Guidotti et al., 2018; Nauta et al., 2023). Fidelity can be quantified through the difference between the prediction model's outputs and the surrogate model's outputs (Yeh et al., 2019; Zhou et al., 2019). Explanations that quantitatively describe the prediction model's output under perturbations with low error have *high fidelity*.

An implication of models with high representative capacity and high-fidelity explanations is the *recovery property*: If the true relation f between features and labels in the data is already within the function class $\mathcal{G}$, the model will learn this function and we can effectively reconstruct the original f from the explanations. For example, suppose that the black-box function we consider is of linear form, i.e., $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$, and has been correctly learned. In this case, a gradient explanation as well as continuous LIME (C-LIME, Agarwal et al., 2021) will recover the original model's parameters up to an offset (Han et al., 2022, Theorem 1). Shapley value explanations (Lundberg & Lee, 2017) possess a comparable relationship: It is known that they correspond to the pointwise feature contributions of Generalized Additive Models (GAM, Bordt & von Luxburg, 2023). The significance of recovery properties lies in their role when explanations are leveraged to gain insights into the underlying data. Particularly when XAI is used for scientific applications such as drug discovery (Mak et al., 2023), preserving the path from the input data to the explanation through a learned model is crucial. However, such guarantees can only be provided when surrogate function class $\mathcal{G}$ can effectively mimic the model's learned relation, at least within some local region.

In this study, we demonstrate that the transformer architecture, the main building block of LLMs such as the GPT models (Radford et al., 2019), is inherently incapable of learning additive models on the input tokens, both theoretically and empirically. By additive models, we refer to models that assign each token a weight. The sum of the individual token weights then gives the output of the additive model. Linear models are a subset of this class. We formally prove that simple encoder-only and decoder-only transformers structurally cannot represent such additive models due to the attention mechanism's softmax normalization, which necessarily introduces token dependencies over the entire sequence length. An example is illustrated in Figure 1 (left). Our finding that the function spaces represented by additive models and transformers are

disjoint when dismissing trivial cases implies that prevalent feature attribution explanations *are insufficient* to model transformers. They cannot possess high fidelity, i.e., they cannot quantitatively describe these models' behavior well. This also undermines the recovery property, highlighting a significant oversight in current XAI practices. As our results suggest that the role of tokens cannot be described through a single score, we introduce the Softmax-Linked Additive Log Odds Model (SLALOM, cf. Figure 1, right), which represents the role of each input token in two dimensions: The *token value* describes the independent effect of a token, whereas the *token importance* provides a token's interaction weight when combined with other tokens in a sequence. In summary, our work offers the following contributions beyond the related literature:

(1) We theoretically and empirically demonstrate that common transformer architectures fail to represent additive and linear models on the input tokens, jeopardizing current attribution methods' fidelity.
(2) To mitigate these issues, we propose the Softmax-Linked Additive Log Odds Model (SLALOM), which uses a combination of two scores to quantify the role of input tokens.
(3) We theoretically analyze SLALOM and show that (i) it can be represented by transformers (i.e., the fidelity property), (ii) it can be uniquely identified from data (i.e., the recovery property), and (iii) it is highly efficient to estimate.
(4) Experiments on synthetic and real-world datasets with common language models (LMs) confirm the mismatch between surrogate models and predictive models, underline that two scores cover different angles of interpretability, and that SLALOM explanations can be computed that have substantially higher fidelity or efficiency than competing techniques.

2 Related Work

Explainability for transformers. Various methods exist to tackle model explainability (Molnar, 2019; Burkart & Huber, 2021). Furthermore, specific approaches have been devised for the transformer architecture (Vaswani et al., 2017): As the attention mechanism at the heart of transformer models is supposed to focus on relevant tokens, it seems a good target for explainability methods. Several works turn to attention patterns as model explanation techniques. A central attention-based method is put forward by Abnar & Zuidema (2020), who propose two methods of aggregating raw attentions across layers, *flow* and *rollout*. Brunner et al. (2020) focus on effective attentions, which aim to identify the portion of attention weights actually influencing the model's decision. While these approaches follow a scalar approach considering only attention weights, Kobayashi et al. (2020; 2023) propose a norm-based vector-valued analysis, arguing that relying solely on attention weights is insufficient and the other components of the model need to be considered. Building on the norm-based approach, Modarressi et al. (2022; 2023) further follow down the path of decomposing the transformer architecture, presenting global-level explanations with the help of rollout. Beyond that, many more attention-based explanation approaches have been put forward (Chen et al., 2020; Hao et al., 2021; Ferrando & Costa-jussà, 2021; Qiang et al., 2022; Sun et al., 2023; Yang et al., 2023) and relevance-propagation methods such as LRP have been adapted to the transformer architecture (Achtibat et al., 2024). A drawback with these model-specific explanations remains the implementational overhead that is required to adapt these methods for each architecture. On the formal side, there is no explicit method to quantitatively predict the transformer model's behavior under perturbations leaving the fidelity of these explanations unclear.

Model-agnostic XAI. In contrast to transformer-specific methods, researchers have devised model-agnostic explanations that can be applied without precise knowledge of a model's architecture. Model-agnostic local explanations like LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2017) and others (Shrikumar et al., 2017; Sundararajan et al., 2017; Smilkov et al., 2017; Xu et al., 2020; Covert et al., 2021, etc.) are a particularly popular class of explanations that are applied to LMs as well (Szczepański et al., 2021; Schirmer et al., 2023; Dolk et al., 2022). Surrogate models are a common subform (Han et al., 2022), which locally approximate a black-box model through a simple, interpretable function.

Linking models and explanations. Prior work has distilled the link between classes of surrogate models that can be recovered by explanations (Agarwal et al., 2021; Han et al., 2022, Theorem 3). Notable works include Garreau & von Luxburg (2020), which provides analytical results on different parametrizations of LIME, and Bordt & von Luxburg (2023), which formalizes the connection between Shapley values and GAMs for classical Shapley values as well as n-Shapley values that can also model higher-order interactions.

Interpreting feature attributions as linear models. Many feature attribution methods do not use explicit surrogate models, e.g., SHAP or LRP. To predict model behavior under perturbations, they can be intuitively interpreted as a linear surrogate model (cf. Yeh et al., 2019): If feature i has a contribution of ϕ_i, removing i should reduce the model output by ϕ_i (Achtibat et al., 2024), giving rise to an implicit linear model.

We contribute to the literature by theoretically showing that transformers are inherently incapable to represent additive models, casting doubts on the applicability of local LIME, SHAP, attention-based, and other attribution methods to transformers. These methods fit an explicit additive surrogate model or can be interpreted as one. Our finding that additive scores are insufficient to predict the behavior of transformers leaves us with no method that enables strict fidelity. To bridge this gap, we provide SLALOM, a novel surrogate model with substantially increased fidelity and a recovery property for transformers.

3 Preliminaries

3.1 Input and output representations

In this work, we focus on classification problems of token sequences. For the sake of simplicity, we initially consider a 2-class classification problem with labels $y \in \mathcal{Y} = \{0, 1\}$. We will outline how to generalize our approach to multi-class problems in Appendix C.1.

The input consists of a sequence of tokens $\boldsymbol{t} = [t_1, \ldots, t_{|\boldsymbol{t}|}]$ where $|\boldsymbol{t}| \in 1, \ldots, C$ is the sequence length that can span at most C tokens (the context length). All tokens t_i in the sequence $\boldsymbol{t}$ stem from a finite size vocabulary $\mathcal{V}$, i.e., $t_i \in \mathcal{V}, \forall i = 1, \ldots, |\boldsymbol{t}|$. To transform the tokens into a representation amenable to processing with computational methods, the tokens need to be encoded as numerical vectors. To this end, an embedding function $\boldsymbol{e} : \mathcal{V} \to \mathbb{R}^d$ is used, where d is the embedding dimension. Let $\boldsymbol{e}_i = \boldsymbol{e}(t_i)$ be the embedding of the i-th token. The output is given by a logit vector $\boldsymbol{l} \in \mathbb{R}^{|\mathcal{Y}|}$, such that softmax($\boldsymbol{l}$) contains individual class probabilities.

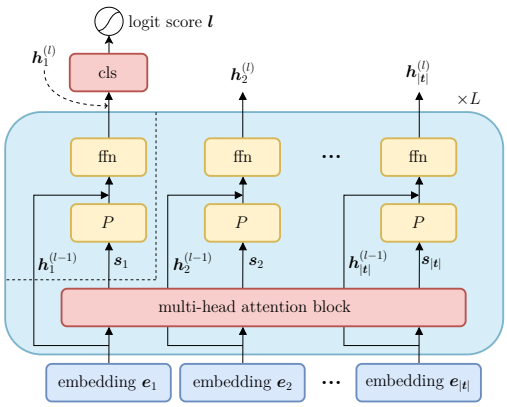


Figure 2: **Transformer architecture.** In each layer l=1, ..., L, input embeddings $\boldsymbol{h}_i^{(l-1)}$ for each token i are transformed into output embeddings $\boldsymbol{h}_i^{(l)}$. When detaching the part prior to the classification head ("cls"), we see that the output only depends on the last embedding $\boldsymbol{h}_1^{(L-1)}$ and attention output $\boldsymbol{s}_1$.

3.2 The common transformer architecture

Many popular LMs follow the transformer architecture introduced by Vaswani et al. (2017) with only minor modifications. We will introduce the most relevant building blocks of the architecture in this section. A complete formalization is given in Appendix B.1. A schematic overview of the architecture is visualized in Figure 2. Let us denote the input embedding of token $i = 1, \ldots, |\boldsymbol{t}|$ in layer $l \in 1, \ldots, L$ by $\boldsymbol{h}_i^{(l-1)} \in \mathbb{R}^d$, where $\boldsymbol{h}_i^{(0)} = \boldsymbol{e}_i$. The core component of the attention architecture is the attention head.[1] For each token, a *query*, *key*, and a *value* vector are computed by applying an affine-linear transform to the input embeddings. Keys and queries are projected onto each other and normalized by a row-wise softmax operation resulting in attention weights $\alpha_{ij} \in [0, 1]$, denoting how much token i is influenced by token j. The attention output for token i can be computed as $\boldsymbol{s}_i = \sum_{j=1}^{|\boldsymbol{t}|} a_{ij} \boldsymbol{v}_j$, where $\boldsymbol{v}_j \in \mathbb{R}^{d_h}$ denotes the value vector for token j. The final $\boldsymbol{s}_i$ are projected back to dimension d by a projection operator $P : \mathbb{R}^{d_h} \to \mathbb{R}^d$ before they are added to the corresponding input embedding $\boldsymbol{h}_i^{(l-1)}$ as mandated by skip-connections. The sum is then transformed by a nonlinear function that we denote by ffn : $\mathbb{R}^d \to \mathbb{R}^d$, finally resulting in a transformed embedding $\boldsymbol{h}_i^{(l)}$. This procedure is repeated iteratively for layers $1, \ldots, L$ such that we finally arrive at output embeddings $\boldsymbol{h}_i^{(L)}$. To perform classification, a classification head cls : $\mathbb{R}^d \to \mathbb{R}^{|\mathcal{Y}|}$ is put on top of a token at some index r (how

[1] Although we only formalize a single head here, our theoretical results cover multiple heads as well

this token is chosen depends on the architecture, common choices include $r \in \{1, |\boldsymbol{t}|\}$), such that we get the final logit output $\boldsymbol{l} = \text{cls}\left(\boldsymbol{h}_r^{(L)}\right)$. The logit output is transformed to a probability vector via another softmax operation. Note that in the two-class case, we obtain the log odds $F(\boldsymbol{t})$ by taking the difference (Δ) between the two logits, i.e., $F(\boldsymbol{t}) \coloneqq \log \frac{p(y=1|\boldsymbol{t})}{p(y=0|\boldsymbol{t})} = \Delta(\boldsymbol{l}) = \boldsymbol{l}_1 - \boldsymbol{l}_0$.

3.3 Encoder-only and decoder-only models

Practical architectures can be seen as parametrizations of the process described previously. We introduce the ones relevant to this work in this section. Commonly, a distinction is made between *encoder-only* models, that include BERT (Devlin et al., 2018) and its variants, and *decoder-only* models such as the GPT models (Radford et al., 2018; 2019).

Encoder-only models. Considering BERT as an example of an encoder-only model, the first token is used for the classification head, i.e., $r = 1$. Usually, a special token `[CLS]` is prepended to the text at position 1. However this is not strictly necessary for the functioning of the model.

Decoder-only models. In contrast, decoder-only models like GPT-2 (Radford et al., 2019) add the classification head on top of the last token for classification, i.e., $r = |\boldsymbol{t}|$. A key difference is that in GPT-2 and other decoder-only models, a causal mask is laid over the attention matrix, resulting in $\alpha_{i,j} = 0$ for $j > i$. This encodes the constraint that tokens can only attend to themselves or to previous ones.

To make our model amenable to theoretical analysis, the transformer model in our analysis contains one slight deviation from practical models. We do not consider positional embeddings, which are added to the embeddings based on their position in the sentence. We will empirically demonstrate that using positional embeddings does not affect the validity of our findings on practical models.

4 Analysis

Let us initially consider a transformer with only a single layer and head. Our first insight is that the classification output can be determined only by two values: the input embedding at the classification token r, $\boldsymbol{h}_r^{(0)}$, and the attention output s_r. This can be seen when plugging in the different steps:

$$F(\boldsymbol{t}) = \Delta\left(\text{cls}(\boldsymbol{h}_r^{(1)})\right) = \Delta\left(\text{cls}\left(\text{ffn}(\boldsymbol{h}_r^{(0)} + \boldsymbol{P}(\boldsymbol{s}_r))\right)\right) \coloneqq g(\boldsymbol{h}_r^{(0)}, \boldsymbol{s}_r) = g\Big(\boldsymbol{h}_r^{(0)}, \sum_{j=1}^{|\boldsymbol{t}|} a_{rj}\boldsymbol{v}_j\Big). \tag{1}$$

The attention output is given by a sum of the token value vectors $\boldsymbol{v}_j$ weighted by the respective attention weights α_{rj}.

4.1 Transformers cannot represent additive models

We now consider how this architecture would represent a linear model. In this model, each token is assigned a weight $w : \mathcal{V} \to \mathbb{R}$. The output is obtained by adding weights and an offset $b \in \mathbb{R}$, consequently requiring

$$F([t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]) = b + \sum_{i=1}^{|\boldsymbol{t}|} w(t_i) \tag{2}$$

for all possible input sequences.

The transformer shows surprising behavior when considering sequences of identical tokens but of different lengths, i.e., $[\tau], [\tau, \tau], \ldots$ We first note that the sum of the attention scores is bound to be $\sum_{j=1}^{|\boldsymbol{t}|} a_{rj} = 1$. The output of the attention head will thus be a weighted average of the value vectors $\boldsymbol{v}_i$. We find that the first-layer value vectors $\boldsymbol{v}_j(t_j)$ are determined purely by the input tokens t_j (cf. full formalization in Appendix B.1). For a sequence of identical tokens, we will thus have the same value vectors, resulting in identical vectors being averaged. This makes the transformer produce the same output for each of these sequences. This contradicts the form in (2), where the output should successively increase by $w(t_i)$. We are now ready to state our result, which formalizes this intuition for the more general class of additive models where the token weight may also depend on its position i in the input (equivalent to a GAM).

Proposition 4.1 (Single-layer transformers cannot represent additive models.)**.** *Let $\mathcal{V}$ be a vocabulary and $C \geq 2, C \in \mathbb{N}$ be a maximum sequence length (context length). Let $w_i : \mathcal{V} \to \mathbb{R}, \forall i \in 1, ..., C$ be any map that assigns a token encountered at position i a numerical score including at least one token $\tau \in \mathcal{V}$ with non-zero weight $w_i(\tau) \neq 0$ for some $i \in 2, \ldots, C$. Let $b \in \mathbb{R}$ be an arbitrary offset. Then, there exists no parametrization of the encoder or decoder single-layer transformer F such that for every sequence $\boldsymbol{t} = [t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]$ with length $|\boldsymbol{t}| \leq C$, the output of the transformer network is equivalent to $F([t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]) = b + \sum_{i=1}^{|\boldsymbol{t}|} w_i(t_i)$.*

Proof Sketch. We prove the statement by concatenating the token τ to sequences of different length. We then show that the inputs to the final part g of the transformer will be independent of the sequence length. Due to g being deterministic, the output will also be independent of the sequence length. This is contradictory to the additive model with a weight $w_j(\tau) \neq 0$ requiring different outputs for sequences of length $j-1$ and j. Formal proofs for all results can be found in Appendix B. □

In simple terms, the proposition states that the transformer cannot represent any additive models on sequences of more than one token besides constant functions or those fully determined by the first input token. Importantly, the class of functions stated in the above theorem includes the prominent case of linear models in Eqn. (1), where each token has a certain weight w independent of its position in the input vector (i.e., $w_i \equiv w, \forall i$, see Corollary B.2). We would like to emphasize that this statement includes the converse:

Corollary 4.2. *Transformers whose outputs are not constant or fully determined by the first token of the input sequence cannot be functionally equivalent to an additive model.*

4.2 Transformer networks with multiple layers cannot represent additive models

In this section, we will show how the argument can be extended to multi-layer transformer networks. Denote by $\boldsymbol{h}_i^{(l-1)}$ the input embedding of the ith token at the lth layer. The output is governed by the recursive relation

$$\boldsymbol{h}_i^{(l)} = \mathrm{ffn}_l(\boldsymbol{h}_i^{(l-1)} + \boldsymbol{P}_l(\boldsymbol{s}_i)) = g_l(\boldsymbol{h}_i^{(l-1)}, \boldsymbol{s}_i). \tag{3}$$

Exploiting the similar form allows us to generalize the main results to more layers recursively.

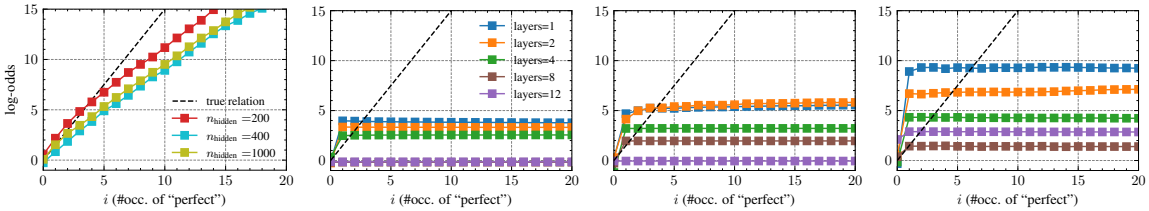
Corollary 4.3 (Multi-Layer transformers cannot learn additive models either)**.** *Under the same conditions as in Proposition 4.1, a stack of multiple transformer blocks as the model F, neither has a parametrization sufficient to represent the additive model.*

Practical considerations. As stated earlier, the transformer model in our analysis does not consider positional embeddings that are added on the token embeddings. However, this does not have major ramifications in practice: While the transformer would be able to differentiate between sequences of different lengths with positional embeddings in theory, the softmax operation must be inverted for any input sequence by the linear feed-forward block that follows the attention mechanism. This is a highly non-linear operation and the number of possible sequences grows exponentially with the context length and vocabulary size. Learning-theoretic considerations suggest that this inversion is impossible for reasonably-sized networks as outlined in Appendix C.2. We will confirm our results with empirical findings obtained exclusively on non-modified models with positional embeddings.

5 A Surrogate Model for Transformers

In the previous section, we theoretically established that transformer models struggle to represent additive functions. While this must not necessarily be considered a weakness, it certainly casts doubts on the suitability of additive models as surrogate models for explanations of transformers. For a principled approach, we consider the following four requirements to be of importance:

(1) **Interpretability.** The surrogate model should be simple enough such that its parameters are inherently interpretable for humans (Molnar, 2019, Chapter 9.2).



(a) Fully-Connected Model (b) BERT (c) DistilBERT (d) GPT-2

Figure 3: **Transformers fail to learn linear models.** We train different models on a synthetically sampled dataset where the log odds obey a linear relation to the features. Fully connected models (2-layer ReLU networks with different hidden layer widths) capture the linear form of the relationship well despite some estimation error (a). However, common transformer models fail to model this relationship and output almost constant values (b)-(d). This does not change with more layers.

(2) **Learnability.** The surrogate model should be easily representable by common transformers. Learnability is crucial because using a surrogate model that is hard to represent for the predictive model will likely result in low-fidelity explanations.

(3) **Recovery.** If the predictive model falls into the surrogate model's class, the fitted surrogate model's parameters should match those of the predictive model. Together with learnability, recovery ensures that a model can pick up the true relations present in the data and they can be re-identified by the explanation, which is essential, e.g., in scientific discovery with XAI.

(4) **Efficiency.** The surrogate model should be efficient to estimate even for larger models.

5.1 The Softmax-Linked Additive Log Odds Model

To meet the requirements, we propose a novel discriminative surrogate model that explicitly models the behavior of the softmax function. Instead of only assigning a single weight w to each token, we separate two characteristics: We introduce the *token importance* as a map $s : \mathcal{V} \to \mathbb{R}$ and a *token value* in form of a map $v : \mathcal{V} \to \mathbb{R}$. Subsequently, we consider the following discriminative model:

$$F(\boldsymbol{t}) = \log \frac{p(y=1|\boldsymbol{t})}{p(y=0|\boldsymbol{t})} = \sum_{\tau_i \in \boldsymbol{t}} \alpha_i(\boldsymbol{t}) v(t_i), \qquad \text{where } \alpha_i(\boldsymbol{t}) = \frac{\exp(s(t_i))}{\sum_{t_j \in \boldsymbol{t}} \exp(s(t_j))}. \tag{4}$$

Due to the shift-invariance of the softmax function, we observe that the maps s and s' given by $s'(\tau) = s(\tau) + \delta$ result in the same softmax-score and thus the same log odds model for any input $\boldsymbol{t}$. Therefore, the parameterization would not be unique. To this end, we introduce a *normalization constraint* on the sum of token importances for uniqueness. Formally, we constrain it to a user-defined constant $\gamma \in \mathbb{R}$ such that $\sum_{\tau \in \mathcal{V}} s(\tau) = \gamma$, where natural choices include $\gamma \in \{0, 1\}$. We refer to the discriminative model given in Eqn. (4) together with the normalization constraint as the *softmax-linked additive log odds model* (SLALOM).

As common in surrogate model explanations, we can fit SLALOM to a predictive model's outputs globally or locally and use tuples of token importance scores and token values scores, $(v(\tau), s(\tau))$ to give explanations for an input token τ. While the value score provides an absolute contribution of τ to the output, its token importance $s(\tau)$ determines its weight with respect to the other tokens. For instance, if only one token τ is present in a sequence, the output is only determined by its value score $v(\tau)$. However, in a sequence of multiple tokens, the importance of each token with respect to the others – and thereby the contribution of this token's value – is determined by the token importance scores s. This intuitive relation makes SLALOM interpretable, thereby satisfying Property (1).

5.2 Theoretical properties of SLALOM

We analyze the proposed SLALOM theoretically to ensure that it fulfills Properties (2) and (3), Learnability and Recovery, and subsequently provide efficient algorithms to estimate its parameters (4). First, we show that – unlike linear models, SLALOMs can be easily learned by transformers.

Proposition 5.1 (Transformers can fit SLALOM)**.** *For any maps s, v, and a transformer with an embedding size d and head dimension d_h with $d, d_h \geq 3$, there exists a parameterization of the transformer to reflect SLALOM in Equation* (4) *together with the normalization constraint.*

This statement can be proven by explicitly constructing the corresponding weight matrix (cf. Appendix B.5). This proposition highlights that – unlike linear models – there are simple ways for the transformer to represent relations governed by SLALOMs. We demonstrate this empirically in our experimental section and conclude that SLALOM fulfills Property (2). For Property (3), Recovery, we make the following proposition:

Proposition 5.2 (Recovery of SLALOMs)**.** *Suppose query access to a model G that takes sequences of tokens $\boldsymbol{t}$ with lengths $|\boldsymbol{t}| \in 1, \ldots, C$ and returns the log odds according to a non-constant SLALOM on a vocabulary $\mathcal{V}$, normalization constant $\tau \in \mathbb{R}$, but with unknown parameter maps $s : \mathcal{V} \to \mathbb{R}$, $v : \mathcal{V} \to \mathbb{R}$. For $C \geq 2$, we can recover the true maps s, v with $2|\mathcal{V}| - 1$ forward passes of F.*

This statement confirms property (3) and shows that SLALOM can be uniquely re-identified when we rule out the corner case of constant models. We prove it in Appendix B.6.

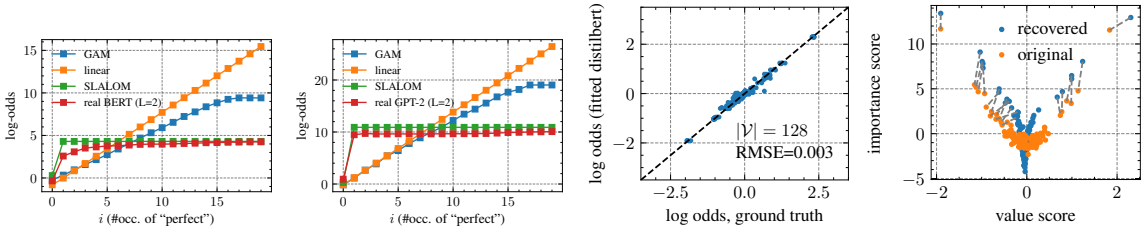
Complexity considerations. Computational complexity can be a concern for XAI methods. To estimate exact Shapley values, the model's output on exponentially many feature coalitions needs to be evaluated. However, as the proof of Proposition 5.2 shows, to estimate SLALOM's parameters for an input sequence of $\mathcal{V}$ tokens, only $2|\mathcal{V}| - 1$ forward passes are required, verifying Property (4). We empirically show that computing SLALOM explanations is about **5× faster** than computing SHAP explanations when using the same number of samples in our experimental section.

5.3 Numerical algorithms for computing SLALOMs

Having derived SLALOM as a better surrogate model, we require numerical algorithms to estimate v and s. Unfortunately, the strategy derived in Proposition 5.2 using a minimal number of samples is numerically unstable. We make two key implementation choices for SLALOM to be used as an explanation technique. First, we can control the sample set of features and labels obtained through queries of the predictive model. Second, we can use different optimization strategies to fit SLALOM on this sample set. We suggest two algorithms to fit SLALOMs post-hoc on input-output pairs of a trained predictive model:

SLALOM-`eff`. The first version of the algorithm to fit SLALOM models is designed for maximum efficiency while maintaining reasonable performance across several XAI metrics. Obtaining a large dataset of input-output pairs can incur substantial computational costs as a forward pass of the models needs to be executed for each sample. To speed up this process, SLALOM-`Eff` uses very short sequences (we use only two tokens in this work) randomly sampled from the vocabulary for this purpose. To efficiently fit the surrogate model, we perform stochastic gradient descent on SLALOM's parameters using the mean-squared-error loss between the score output by SLALOM and the score by the predictive model as an objective. SLALOM-`eff` is our default technique used unless stated otherwise.

SLALOM-`fidel`. We provide another technique to fit SLALOM optimized for maximum fidelity under input perturbations such as token removals. To explain a specific sample, we sample input where we remove up to K randomly sampled tokens. The sequences with tokens removed and their predictive model scores are used to fit the model, similar to LIME (Ribeiro et al., 2016). Instead of SGD, we can leverage optimizers for Least-Square-Problems to fit the parameters iteratively, however incurring a higher latency. We provide details and pseudocode for both fitting routines in Appendix D.



(a) Fitting SLALOM to BERT outputs (b) Fitting SLALOM to GPT-2 outputs (c) Comparing SLALOM and predictive output (d) Recovering parameters of SLALOM dataset

Figure 4: **Verifying properties with synthetic data: SLALOM describes outputs of transformer models well (a, b).** We fit SLALOM to the outputs of the BERT and GPT-2 models trained on the linear synthetic dataset. The linear and GAM models (despite having $C/2$=15× more parameters) do not match the transformer's behavior. We provide another empirical counterexample and additional quantitative results in Appendix F.1. **Verifying recovery (c, d).** We verify the recovery property on a second synthetic dataset where features and labels obey a SLALOM relation. We train a 2-layer DistilBERT model on the data and fit SLALOM to the trained model. We can recover the original logit scores (c) and see a strong connection between original SLALOM parameters and the recovered ones (d). These findings verify the learnability and recovery properties. More results in Appendix F.2.

5.4 Relating SLALOM scores to linear attributions

Importantly, SLALOM scores can be readily converted to locally linear interpretability scores where necessary. For this purpose, a differentiable model for soft removals is required. We consider the weighted model:

$$F(\boldsymbol{\lambda}) = \frac{\sum_{t_i \in \boldsymbol{t}} \lambda_i \exp(s(t_i)) v(t_i)}{\sum_{t_i \in \boldsymbol{t}} \lambda_i \exp(s(t_i))}, \tag{5}$$

where $\lambda_i = 1$ if a token is present and $\lambda_i = 0$ if it is absent. We observe that setting $\lambda_i = 0$ has the desired effect of making the output of the soft-removal model equivalent to that of the standard SLALOM on a sequence without this token. Taking the gradients at $\boldsymbol{\lambda} = \mathbf{1}$ we obtain $\frac{\partial F}{\partial \lambda_i}\Big|_{\boldsymbol{\lambda}=\mathbf{1}} \propto v(t_i)\exp(s(t_i))$, which can be used to rank tokens according to the linearized attributions. We defer the derivation to Appendix B.7 and refer to these scores as *linearized* SLALOM scores. As SLALOM is just another multi-variate function, it also possible to compute SHAP values for it. We discuss this relation in Appendix B.7 as well.

6 Experimental Evaluation

We run a series of experiments to show the mismatch between surrogate model explanations and the transformers. Specifically, we verify that (1) real transformers fail to learn additive models, (2) SLALOM better captures transformer output, (3) SLALOM models can be recovered from fitted models with tolerable error, (4) SLALOM scores are versatile and align well with linear attribution scores and human attention, and (5) that SLALOM performs well in faithfulness metrics and has substantially higher fidelity than competing techniques. For experiments (1)-(3), we require knowledge of the ground truth and use synthetic datasets. To demonstrate the practical strengths of our method, all the experiments for (4) and (5) are conducted on real-world datasets and in comparison with state-of-the-art XAI techniques.

6.1 Experimental setup

LM architectures. We study three representative transformer language model architectures in our experiments. In sequence classification, mid-sized transformer LMs are most popular on platforms such as the Huggingface hub (Huggingface, 2023) often based on the BERT-architecture (Devlin et al., 2018), which is reflected in our experimental setup. To represent the family of encoder-only models, we deploy BERT

(Devlin et al., 2018) and DistilBERT (Sanh et al., 2019). We further experiment with GPT-2 (Radford et al., 2019), which is a decoder-only model. We use the `transformers` framework to run our experiments. While not the main scope of this work concerned with sequence classification, we will also show that due to its model-agnostic nature, SLALOM can be applied to LLMs with up to 7B parameters and non-transformer models such as Mamba (Gu & Dao, 2023) with plausible results. We provide a proof-of-concept to show that SLALOM can be applied to black-box models such as GPT-4o in Appendix F.7.

Datasets. We use synthetic datasets and two real-world datasets for sentiment classification. Specifically, we study the IMDB dataset, consisting of movie reviews (Maas et al., 2011) and Yelp-HAT, a dataset of restaurant reviews, for which human annotators have provided annotations on which tokens are relevant for the classification outcome (Sen et al., 2020). We provide additional details on hyperparameters, training, and datasets in Appendix E.

6.2 Evaluation with known ground truth

Transformers fail to capture linear relationships. We empirically verify the claims made in Proposition 4.1 and Corollary 4.3. To ascertain that the underlying relation captured by the models is additive, we resort to a synthetic dataset with a linear relation. The dataset is created as follows: First, we sample different sequence lengths from a binomial distribution with a mean of 15. Second, we sample words independently from a vocabulary of size 10. This vocabulary was chosen to include positive words, negative words, and neutral words, with manually assigned weights $w \in \{-1.5, -1, 0, 1, 1.5\}$, that can be used to compute a linear log odds model. We evaluate this model and finally sample the sequence label accordingly, thereby ensuring a linear relation between input sequences and log odds. We train transformer models on this dataset and evaluate them on sequences containing the same word ("perfect") multiple times. Our results in Figure 3 show that the models fail to capture the relationship regardless of the model or number of layers used. In Appendix A, we show how this undermines the recovery property with Shapley value explanations.

Fitting SLALOM as a surrogate to transformer models. Having demonstrated the mismatch between additive functions and transformers, we turn to SLALOM as a more suitable surrogate model. As shown in Proposition 5.1, transformers can easily fit SLALOMs, which is why we hypothesize that they should model the output of such a model well in practice. We fit the different surrogate models on a dataset of input sequences and real transformer outputs from our linear synthetic dataset and observe that linear models and additive models fail to capture the relationship learned by the transformer as shown in Figure 4(a, b). On the contrary, SALO manages to model the relationship well, even if it has considerably less trainable parameters than the additive model (GAM).

Verifying recovery. We run an experiment to study whether, unlike linear models, SLALOM can be fitted and recovered by transformers. To test this, we sample a second synthetic dataset that exactly follows the relation given by SLALOM. We then train transformer models on this dataset. The results in Figure 4(c, d) show that the surrogate model fitted on transformer outputs as a post-hoc explanation recovers the correct log odds mandated by SLALOM (c) and that there is a good correspondence between the true model's parameters and the recovered model's parameters (d).

6.3 Examining real-world predictions from different angles

We increase the difficulty and deploy SLALOM (fitted using SLALOM-`eff`) to explain predictions on real-world datasets. As there is no ground truth for these datasets, it is challenging to evaluate the quality of the explanations (Rong et al., 2022). To better understand SLALOM explanations, we study them from several angles: We compare to linear scores obtained when fitting a Naïve-Bayes Bag-of-Words (BoW) model, scores on removal and insertion benchmarks (Tomsett et al., 2020; DeYoung et al., 2020), the human attention scores available on the Yelp-HAT dataset (Sen et al., 2020), and provide qualitative results.

Explaining Sentiment Classification. We show qualitative results for explaining a movie review in Figure 5. The figure shows that both negative and positive words are assigned high importance scores but have value scores of different signs. Furthermore, we see that some words ("the") have positive value scores, but a very low importance. This means that they lead to positive scores on their own but are easily

LM	values v	importances s	lin.
BERT	0.619 ± 0.01	0.349 ± 0.01	**0.626** ± 0.01
DistilBERT	0.692 ± 0.01	0.373 ± 0.01	**0.693** ± 0.01
GPT-2	0.618 ± 0.01	0.292 ± 0.01	**0.619** ± 0.01
average	0.643	0.338	**0.646**

(a) Measuring average rank-correlation (Spearman) between Naive-Bayes scores and SLALOM scores. Linearized performs best.

LM	values v	importances s	lin.
Bert	0.786 ± 0.01	**0.807** ± 0.01	0.801 ± 0.01
DistilBERT	**0.688** ± 0.01	0.681 ± 0.01	0.686 ± 0.01
GPT-2	0.674 ± 0.01	**0.685** ± 0.01	0.683 ± 0.01
average	0.716	**0.724**	0.724
BLOOM-7.1B	0.739 ± 0.02	0.712 ± 0.03	0.740 ± 0.02
Mamba-2.8B	0.615 ± 0.03	0.437 ± 0.03	0.535 ± 0.03

(b) Measuring average AU-ROC between SLALOM explanations and human token attention. The importance scores and most strongly predictive of human attention for the classical models. Applying SLALOM to larger models underlines its scalability, but it remains less reliable for non-transformer models like Mamba.

	SLALOM-`fidel`		SLALOM-`eff`						
LM	v-scores	lin.	v-scores	lin.	LIME	SHAP	IG	Grad	LRP
BERT	0.025 ± 0.002	**0.023** ± 0.001	0.031 ± 0.002	0.031 ± 0.002	0.024 ± 0.002	0.026 ± 0.003	0.557 ± 0.034	0.611 ± 0.033	0.030 ± 0.006
DistilBERT	0.028 ± 0.003	0.024 ± 0.002	0.027 ± 0.002	0.027 ± 0.002	0.027 ± 0.003	0.029 ± 0.003	0.495 ± 0.027	0.508 ± 0.028	**0.023** ± 0.002
GPT-2	0.052 ± 0.008	0.050 ± 0.008	0.089 ± 0.008	0.089 ± 0.008	0.230 ± 0.017	**0.042** ± 0.005	0.454 ± 0.022	0.493 ± 0.023	0.069 ± 0.010
average	0.035 ± 0.004	**0.032** ± 0.004	0.049 ± 0.004	0.049 ± 0.004	0.094 ± 0.007	**0.032** ± 0.003	0.502 ± 0.028	0.537 ± 0.028	0.041 ± 0.006

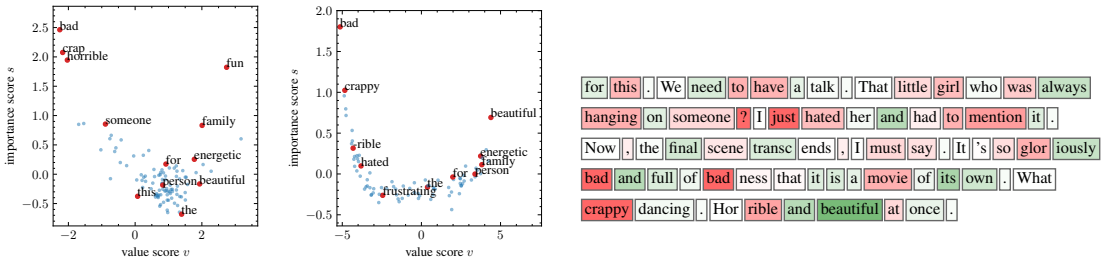
(c) Area Over Perturbation Curve for deletion. Linearized scores and SHAP performs best in the XAI metric.

Table 1: Evaluation of SLALOM scores ("values", "importance", "lin.") with std. errors across explanation quality measures highlights that SLALOM's different scores serve different purposes. IMDB dataset shown.

overruled by other words. We compare the SLALOM scores obtained on 100 random test samples to a linear Naïve-Bayes model (obtained though counting class-wise word frequencies) as a surrogate ground truth in Table 1a through the Spearman rank correlation. We observe good agreement with the value scores v and the combined linearized SLALOM scores ("lin", see Section 5.4).

Predicting Human Attention. To study alignment with a user perspective, we predict human attention from SLALOM scores. We compute AU-ROC for predicting annotated human attention as suggested in Sen et al. (2020) in Table 1b. We use absolute values of all signed explanations as human attention is unsigned as well. In contrast to the previous experiments, where value scores were more effective than importances, we observe that the importance scores are often best at predicting where the human attention is placed. In summary, these findings highlight that the two scores serve different purposes and cover different dimensions of interpretability. SLALOM offers higher flexibility through its 2-dimensional representation.

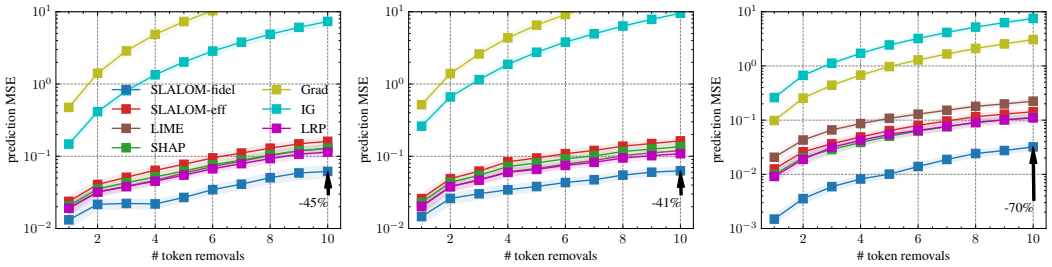
We also use this opportunity to study the applicability of SLALOM to larger and non-transformer models. To this end, we train Mamba and BLOOM models (Le Scao et al., 2023) with a classification head on the



(a) BERT (b) GPT-2 (c) Token highlighting (linearized SLALOM, GPT-2)

Figure 5: Explaining a real review with SLALOM (qualitative results). SLALOM assigns two scores to each token (a,b) and can be used to compute attributions via its linearization (c). We observe that the impactful words have high importances and the value scores indicate the sign of their contribution (positive or negative words). See Figure 12 (Appendix) for fully annotated plots.

Published in Transactions on Machine Learning Research (12/2024)



(a) BERT (b) DistilBERT (c) GPT-2

Figure 6: **Assessing Fidelity.** We plot the MSE for predicting model outputs under token removal and find that SLALOM's predictions have up to 70% less error than the closest competitor when up to 10 random tokens from a sentence are removed (log-y plots). We interpret LRP scores as a linear model.

Yelp dataset. Our results in the lower part of Table 1b prove that SLALOM can be well applied to larger models with sizes of up to 7B. For BLOOM the results are promising and reach the same level as for the classical transformer models. SLALOM can also be applied to non-transformer models like Mamba (Gu & Dao, 2023) due to its model-agnostic nature. Due to its general expressivity, the explanations are capable of predicting human attention (at least the value scores), but we observe a slight reduction in quality. We thus see SLALOM's main scope with models that follow the classical transformer paradigm. We show that SLALOM's results for BoW correlations and human attention prediction are in the same range and often outperform competing XAI techniques in Appendix F.3.2, but defer a comparative analysis to the next sections.

Assessing Fidelity. Having established the roles of its components, we verify that SLALOM can produce explanations that have substantially higher fidelity than competing surrogate or non-surrogate explanation techniques. To assess this, we remove up to 10 tokens from the input sequences and use the explanations to predict the change in model output using the surrogate model (SLALOM or linear). We compare the two SLALOM versions to baselines such as LIME (Ribeiro et al., 2016), Kernel-SHAP (Lundberg & Lee, 2017), Gradients (Simonyan et al., 2013), and Integrated Gradients (IG, Sundararajan et al., 2017) and layer-wise relevance propagation (LRP) for transformers (Achtibat et al., 2024), a non-surrogate technique. We report the Mean-Squared-Error (MSE) between the predicted change and the observed change when running the model on the modified inputs in Figure 6. We observe that SLALOM-`fidel` offers substantially higher fidelity with a reduction of 70% in MSE for predicting the output changes over the second-best method (LRP). Other surrogate approaches and LRP remain cluttered together, potentially highlighting the frontier of maximum fidelity possible with a linear surrogate model.

Evaluating XAI Metrics. There are several other metrics to quantify the quality of explanations and to compare different explanation techniques. As a sanity check and to show that SLALOM explanations do not lag behind other techniques in established metrics, we run the classical insertion/removal benchmarks (Tomsett et al., 2020). For the insertion benchmark, we successively add the tokens with the highest attributions to the sample, which should result in a high score for the target class. We iteratively insert more tokens and compute the "Area Over the Perturbation Curve" (AOPC, see DeYoung et al. (2020)), which should be low for insertion. This metric quantifies the alignment of explanations and model behavior but only considers the feature ranking and not the assigned score. For surrogate techniques (LIME, SHAP, SLALOM) we use 5000 samples each. Our results in Tab. 1c highlight that linearized SLALOM scores outperform LIME and LRP and perform on par with SHAP. Removal results for IMDB and results on YELP with similar

Approach	Avg. Time (s)
Grad	0.01 ± 0.00
IG	0.02 ± 0.00
LRP	0.02 ± 0.00
SLALOM-`eff`	2.03 ± 0.01
SLALOM-`fidel`	3.77 ± 0.24
LIME	3.93 ± 0.19
SHAP	11.56 ± 0.03

Table 2: Runtime comparison using 5000 samples to estimate surrogate models.

findings are deferred to Table 11 (Appendix). In conclusion, this shows that on top of SLALOM's desirable properties, it is on par with other techiques in common evaluation metrics.

Computational Costs. Finally, we take a look at the computational costs of the methods, which are mainly determined by sampling the dataset to fit the surrogate model. We provide the runtimes to explain a sample on our hardware when using 5000 samples to estimate surrogates in Table 2 with more results in Appendix F.6. We observe that SHAP incurs the highest computational burden. Among surrogate model explanations SLALOM-`eff` is the most efficient, being about 5× more efficient that SHAP and 2× more efficient than LIME. Nevertheless, non-surrogate techniques are far more efficient as they require only one or few (IG steps) forward or backward passes, but suffer from other disadvantages (e.g., implementation effort, no explicit way to predict model behavior). Overall, our results highlight that the two SLALOM fitting routines can produce explanations of comparable utility to other surrogate models at a fraction of the costs, or produce explanations with higher fidelity at similar costs due to structurally better alignment between surrogate and predictive models.

7 Discussion and Conclusion

In this work, we established that transformer networks are inherently incapable of representing linear or additive models commonly used for feature attribution. We prove that the function spaces learnable by transformers and linear models are disjoint when ruling out trivial cases. This may explain similar incapacities observed in time-series forecasting (Zeng et al., 2023), where they seem incapable of representing certain relations. To address this shortcoming, we have introduced the Softmax-Linked Additive Log Odds Model (SLALOM), a surrogate model for explaining the influence of features on transformers and other complex LMs through a two-dimensional representation.

Our work still has certain limitations that could be addressed in future work. SLALOM is specifically designed to explain the behavior of transformer models and therefore aligned with the classes of functions commonly represented by transformers. However, it would not be a suitable choice to explain models capturing a linear relationship. While SLALOM is generally applicable to any token-based LMs, we recommend using SLALOM only when the model is known to have attention-like non-linearities. Our results indicate that performance for these models is highest. We also note that SLALOM operates at the token level by assigning each individual token importance and value scores. Contextual or higher-order interpretability considering the meaning and impact of phrases, clauses, or sentences is not covered by SLALOM. To complement this theoretical foundation, future work will include further evaluation of SLALOM from a user-centric perspective, for instance, on human-centered evaluation frameworks (Colin et al., 2022). From a broader perspective, we hope that this research paves the way for advancing the interpretability and theoretical understanding of widely adopted transformer models.

Broader Impact Statement

This paper presents theoretical work on better understanding feature attributions in the transformer framework. We advise using caution when using our XAI technique or other model explanation as all explanations present only a simplified view of the complex ML model. Our method works best with models of the transformer architecture. Besides that, we do not see any immediate impact which we feel must be specifically highlighted here.

References

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, 2020.

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. Attnlrp: Attention-aware layer-wise relevance propagation for transformers. In *Forty-first International Conference on Machine Learning*, 2024.

Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 2018.

Sushant Agarwal, Shahin Jabbari, Chirag Agarwal, Sohini Upadhyay, Steven Wu, and Himabindu Lakkaraju. Towards the unification and robustness of perturbation and gradient based explanations. In *International Conference on Machine Learning*, pp. 110–119. PMLR, 2021.

Nabiha Asghar. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*, 2016.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11: 1803–1831, 2010.

Peter Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear vc dimension bounds for piecewise polynomial networks. *Advances in neural information processing systems*, 11, 1998.

Sebastian Bordt and Ulrike von Luxburg. From shapley values to generalized additive models and back. In *International Conference on Artificial Intelligence and Statistics*, pp. 709–745. PMLR, 2023.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. 2020.

Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in english. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4317–4323, 2019.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*, 2020.

Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in neural information processing systems*, 35:2832–2845, 2022.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, 2020.

Alexander Dolk, Hjalmar Davidsen, Hercules Dalianis, and Thomas Vakili. Evaluation of lime and shap in explaining automatic icd-10 classifications of swedish gastrointestinal discharge summaries. In *Scandinavian Conference on Health Informatics*, pp. 166–173, 2022.

Javier Ferrando and Marta R Costa-jussà. Attention weights in transformer nmt fail aligning words between sequences but largely explain model predictions. *arXiv preprint arXiv:2109.05853*, 2021.

Damien Garreau and Ulrike von Luxburg. Explaining the explainer: A first theoretical analysis of lime. In *International conference on artificial intelligence and statistics*, pp. 1287–1296. PMLR, 2020.

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *DSAA*, 2018.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

Tessa Han, Suraj Srinivas, and Himabindu Lakkaraju. Which explanation should i choose? a function approximation perspective to characterizing post hoc explanations. *Advances in Neural Information Processing Systems*, 35:5256–5268, 2022.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 12963–12971, 2021.

Huggingface. Hugging face - models: Most downloaded sequence classification models, 2023. URL `https://huggingface.co/models?pipeline_tag=text-classification&sort=downloads`.

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Oliver Sabel, Jens Ricke, et al. Chatgpt makes medicine easy to swallow: an exploratory case study on simplified radiology reports. *European radiology*, pp. 1–9, 2023.

Gjergji Kasneci and Thomas Gottron. Licon: A linear weighting scheme for the contribution ofinput variables in deep artificial neural networks. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pp. 45–54, 2016.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pp. 7057–7075. Association for Computational Linguistics (ACL), 2020.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Feed-forward blocks control contextualization in masked language models. *arXiv preprint arXiv:2302.00456*, 2023.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. 2023.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Kit-Kay Mak, Yi-Hang Wong, and Mallikarjuna Rao Pichika. Artificial intelligence in drug discovery and development. *Drug Discovery and Evaluation: Safety and Pharmacokinetic Assays*, pp. 1–38, 2023.

Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. Globenc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 258–271, 2022.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. DecompX: Explaining transformers decisions by propagating token decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2649–2664, 2023.

Christoph Molnar. *Interpretable Machine Learning*. 2019.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55(13s):1–42, 2023.

Yao Qiang, Deng Pan, Chengyin Li, Xin Li, Rhongho Jang, and Dongxiao Zhu. Attcat: Explaining transformers via attentive class activation tokens. *Advances in Neural Information Processing Systems*, 35: 5052–5064, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A consistent and efficient evaluation strategy for attribution methods. In *International Conference on Machine Learning*, pp. 18770–18795. PMLR, 2022.

Yao Rong, Tobias Leemann, Thai-Trang Nguyen, Lisa Fiedler, Peizhu Qian, Vaibhav Unhelkar, Tina Seidel, Gjergji Kasneci, and Enkelejda Kasneci. Towards human-centered explainable ai: A survey of user studies for model explanations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Miriam Schirmer, Isaac Misael Olguín Nolasco, Edoardo Mosca, Shanshan Xu, and Jürgen Pfeffer. Uncovering trauma in genocide tribunals: An nlp approach using the genocide transcript corpus. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pp. 257–266, 2023.

Cansu Sen, Thomas Hartvigsen, Biao Yin, Xiangnan Kong, and Elke Rundensteiner. Human attention maps for text classification: Do humans and neural networks focus on the same words? In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4596–4608, 2020.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Tianli Sun, Haonan Chen, Yuping Qiu, and Cairong Zhao. Efficient shapley values calculation for transformer explainability. In *Asian Conference on Pattern Recognition*, pp. 54–67. Springer, 2023.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Mateusz Szczepański, Marek Pawlicki, Rafał Kozik, and Michał Choraś. New explainability method for bert-based model in fake news detection. *Scientific reports*, 11(1):23705, 2021.

Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6021–6029, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shawn Xu, Subhashini Venugopalan, and Mukund Sundararajan. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9680–9689, 2020.

Sen Yang, Shujian Huang, Wei Zou, Jianbing Zhang, Xinyu Dai, and Jiajun Chen. Local interpretation of transformer based on linear decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10270–10287, 2023.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in neural information processing systems*, 32, 2019.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zihan Zhou, Mingxuan Sun, and Jianhua Chen. A model-agnostic approach for explaining the predictions on clustered data. In *2019 IEEE international conference on data mining (ICDM)*, pp. 1528–1533. IEEE, 2019.

A Motivation: Failure Cases For Model Recovery

We provide another motivational example that shows a failure case of current explanation methods on transformer architectures. In this example we test the recovery property for a linear model. We create a synthetic dataset where each word in a sequence $\boldsymbol{t}$ has a linear contribution to the log odds score, formalized by

$$\log \frac{p(y=1|\boldsymbol{t})}{p(y=0|\boldsymbol{t})} = F([t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]) = b + \sum_{i=1}^{|\boldsymbol{t}|} w(t_i). \tag{6}$$

We create a dataset of 10 words (cf. Table 3) and train transformer models on samples from this dataset. We subsequently create sequences that repeatedly contain a single token τ (in this case, τ="perfect"), pass them through the transformers, and use Shapley values (approximated by Kernel-Shap) to explain their output. The result is visualized in Figure 7, and shows that a fully connected model (two-layer, 400 hidden units, ReLU) recovers the correct scores, whereas transformer models fail to reflect the true relationship. This shows that explanation methods that are explicitly or implicitly based on additive models lose their ability to recover the true data-generating process when transformer models are explained.

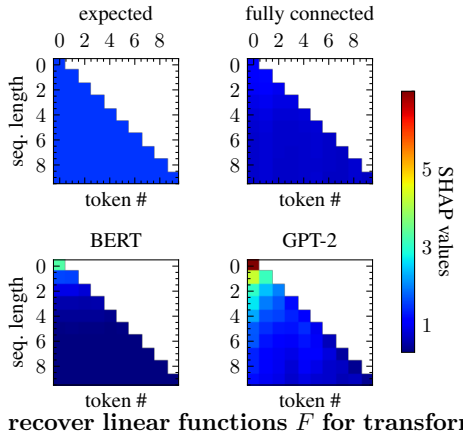


Figure 7: **SHAP values do not recover linear functions F for transformers.** We compute SHAP values for token sequences that repeatedly contain a single token τ with a ground truth score of 1.5 (i.e., $F([\tau])$=1.5, $F([\tau, \tau])$=3.0, ...) such that the ground truth attributions should yield 1.5 independent of the sequence length. While this approximately holds true for a fully connected model, BERT and GPT-2 systematically overestimate the importance for short sequences and underestimate it for longer ones.

B Proofs

B.1 Formalization of the transformer

Many popular LLMs follow the transformer architecture introduced by Vaswani et al. (2017) with only minor modifications. We will introduce the most relevant building blocks of the architecture in this section. A schematic overview of the architecture is visualized in Figure 2. Let us denote the input embeddings for Layer $l \in 1, \ldots L$ by $\boldsymbol{H}^{(l-1)} = [\boldsymbol{h}_1^{(l-1)}, \ldots, \boldsymbol{h}_{|\boldsymbol{t}|}^{(l-1)}]^\top \in \mathbb{R}^{|\boldsymbol{t}| \times d}$, where a single line $\boldsymbol{h}_i$ contains the embedding for token i. The input embeddings of the first layer consist of the token embeddings, i.e., $\boldsymbol{H}^{(0)} = \boldsymbol{E}$, where $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{|\boldsymbol{t}|}]^\top \in \mathbb{R}^{|\boldsymbol{t}| \times d}$ is a matrix of the individual token embeddings. At the core of the architecture lies the attention head. For each token, a *query*, *key*, and a *value* vector are computed by applying an

affine-linear transform. In matrix notation this can be written as

$$\boldsymbol{Q}^{(l)} = \boldsymbol{H}^{(l-1)}\boldsymbol{W}_Q^{(l)} + \mathbf{1}_{|\boldsymbol{t}|}\boldsymbol{b}_Q^{(l)\top}, \tag{7}$$

$$\boldsymbol{K}^{(l)} = \boldsymbol{H}^{(l-1)}\boldsymbol{W}_K^{(l)} + \mathbf{1}_{|\boldsymbol{t}|}\boldsymbol{b}_K^{(l)\top}, \tag{8}$$

$$\boldsymbol{V}^{(l)} = \boldsymbol{H}^{(l-1)}\boldsymbol{W}_V^{(l)} + \mathbf{1}_{|\boldsymbol{t}|}\boldsymbol{b}_V^{(l)\top}, \tag{9}$$

where $\mathbf{1}_{|\boldsymbol{t}|} \in \mathbb{R}^{|\boldsymbol{t}|}$ denotes a vector of ones of length $|\boldsymbol{t}|$, $\boldsymbol{b}_Q^{(l)}, \boldsymbol{b}_V^{(l)}, \boldsymbol{b}_K^{(l)} \in \mathbb{R}^{d_h}$, $\boldsymbol{W}_Q^{(l)}, \boldsymbol{W}_K^{(l)}, \boldsymbol{W}_V^{(l)} \in \mathbb{R}^{d \times d_h}$, are trainable parameters and d_h denotes the dimension of the attention head.[2] Keys and queries are projected onto each other and normalized by a row-wise softmax operation,

$$\boldsymbol{A}^{(l)} = \text{rowsoftmax}\left(\frac{\boldsymbol{Q}^{(l)}\boldsymbol{K}^{(l)\top}}{\sqrt{d_k}}\right). \tag{10}$$

This results in the attention matrix $\boldsymbol{A}^{(l)} \in \mathbb{R}^{|t| \times |t|}$, where row i indicates how much the other tokens will contribute to its output embedding. To compute output embeddings, we obtain attention outputs $\boldsymbol{s}_i$,

$$\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{|\boldsymbol{t}|}]^\top = \boldsymbol{A}^{(l)}\boldsymbol{V}^{(l)}. \tag{11}$$

Note that an attention output can be computed as $\boldsymbol{s}_i = \sum_{j=1}^{|\boldsymbol{t}|} a_{ij}\boldsymbol{v}_j$, where $\boldsymbol{v}_j$ denotes the value vector in the line corresponding to token j in $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{|\boldsymbol{t}|}]^\top$ and $a_{ij} = \boldsymbol{A}_{i,j}^{(l)}$. The final $\boldsymbol{s}_i$ are projected back to the original dimension d by some projection operator $P : \mathbb{R}^{d_h} \to \mathbb{R}^d$ before they are added to the corresponding input embedding $\boldsymbol{h}_i^{(l-1)}$ due to the skip-connections. The sum is then transformed by a nonlinear function that we denote by $\text{ffn} : \mathbb{R}^d \to \mathbb{R}^d$. In summary, we obtain the output for the layer, $\boldsymbol{h}_i^{(l)}$, with

$$\boldsymbol{h}_i^{(l)} = \text{ffn}_l(\boldsymbol{h}_i^{(l-1)} + \boldsymbol{P}(\boldsymbol{s}_i)). \tag{12}$$

This procedure is repeated iteratively for layers $1, \ldots, L$ such that we finally arrive at output embeddings $\boldsymbol{H}^{(L)}$. To perform classification, a classification head $\text{cls} : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{Y}|}$ is put on top of a token at classification index r (how this token is chosen depends on the architecture, common choices include $r \in \{1, |\boldsymbol{t}|\}$), such that we get the final logit output $\boldsymbol{l} = \text{cls}\left(\boldsymbol{h}_r^{(L)}\right)$. The logit output is transformed to a probability vector via another softmax operation. Note that in the two-class case, we obtain the log odds $F(\boldsymbol{t})$ by taking the difference (Δ) between logits

$$F(\boldsymbol{t}) \coloneqq \log\frac{p(y=1|\boldsymbol{t})}{p(y=0|\boldsymbol{t})} = \Delta(\boldsymbol{l}) = \boldsymbol{l}_1 - \boldsymbol{l}_0. \tag{13}$$

B.2 Proof of Proposition 4.1

Proposition B.1 (Proposition 4.1 in the main paper)**.** *Let $\mathcal{V}$ be a vocabulary and $C \geq 2, C \in \mathbb{N}$ be a maximum sequence length (context window). Let $w_i : \mathcal{V} \to \mathbb{R}, \forall i \in 1, ..., C$ be any map that assigns a token encountered at position i a numerical score including at least one token $\tau \in \mathcal{V}$ with non-zero weight $w_i(\tau) \neq 0$ for some $i \in 2, \ldots, C$. Let $b \in \mathbb{R}$ be an arbitrary offset. Then, there exists no parametrization of the encoder or decoder single-layer transformer F such that for every sequence $\boldsymbol{t} = [t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]$ with length $1 \leq |\boldsymbol{t}| \leq C$, the output of the transformer network is equivalent to*

$$F([t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]) = b + \sum_{i=1}^{|\boldsymbol{t}|} w_i(t_i). \tag{14}$$

Proof. We show the statement in the theorem by contradiction. Consider the token, $\tau \in \mathcal{V}$, for which $w_j(\tau) \neq 0$ for some token index $j \geq 2$ which exists by the condition in the theorem. We now consider

[2]We only formalize one attention head here, but consider the analogous caae of multiple heads in our formal proofs.

sequences of length k of the form $\boldsymbol{t}_k = [\underbrace{\tau, \ldots, \tau}_{\text{repeat } k \text{ times}}]$ for $k = 1, \ldots, C$. For example, we have $\boldsymbol{t}_1 = [\tau]$, $\boldsymbol{t}_2 = [\tau, \tau]$, etc. The output of the transformer is given by

$$F(\boldsymbol{t}) = g\left(\boldsymbol{h}_r^{(0)}, \sum_{j=1}^{|\boldsymbol{t}|} a_{rj}\boldsymbol{v}_j\right) = g\left(\boldsymbol{e}(\tau), \sum_{j=1}^{|\boldsymbol{t}|} \alpha_{rj}\boldsymbol{v}_j\right), \tag{15}$$

where r is the token index on which the classification head is placed. Note that with $r = |\boldsymbol{t}|$ for the decoder architecture, the sum always goes up to $|\boldsymbol{t}|$ (for the encoder architecture this is always true). As all tokens in the sequence have a value of τ, we obtain $\boldsymbol{h}_r^{(0)} = \boldsymbol{e}(t_r) = \boldsymbol{e}(\tau)$. The first input to the final part will thus be equal for all sequences $\boldsymbol{t}_k$. We will now show that the second part will also be equal.

We compute the value, key, and query vectors for τ. $\boldsymbol{v}, \boldsymbol{k}, \boldsymbol{q} \in \mathbb{R}^{d_h}$ correspond to one line in the respective key, query and value matrices. As the inputs are identical and we omit positional embeddings in this proof, all lines are identical in the matrices. This results in

$$\boldsymbol{v} = \boldsymbol{W}_V^\top \boldsymbol{e}(\tau) + \boldsymbol{b}_V \tag{16}$$

$$\boldsymbol{k} = \boldsymbol{W}_K^\top \boldsymbol{e}(\tau) + \boldsymbol{b}_K \tag{17}$$

$$\boldsymbol{q} = \boldsymbol{W}_Q^\top \boldsymbol{e}(\tau) + \boldsymbol{b}_Q \tag{18}$$

We omit the layer indices for simplicity. As pre-softmax attention scores (product of key and value vector), we obtain $s = \boldsymbol{q}^\top \boldsymbol{k}/\sqrt{d_k}$. Subsequently, the softmax computation will be performed over the entire sequence, resulting in

$$\boldsymbol{\alpha}_r = \text{softmax}(\underbrace{[s, \ldots, s]}_{k \text{ times}}) = \left[\frac{\exp(s)}{k\exp(s)}\right] \tag{19}$$

$$= \underbrace{\left[\frac{1}{k}, \ldots, \frac{1}{k}\right]}_{k \text{ times}} \tag{20}$$

The second input $\sum_{j=1}^{|\boldsymbol{t}|} \alpha_{rj}\boldsymbol{v}_j$ to the feed-forward part is given by

$$\sum_{j=1}^{|\boldsymbol{t}|} \alpha_{rj}\boldsymbol{v}_j = \sum_{j=1}^{k} \alpha_{rj}\boldsymbol{v}_j = \sum_{j=1}^{k} \frac{1}{k}\boldsymbol{v} = \boldsymbol{v}, \tag{21}$$

as α_{rj} and $\boldsymbol{v}$ are independent of the token index j. We observe that the total input to final part g is independent of k in its entirety, as the first input $\boldsymbol{e}(\tau)$ is independent of k and the second input is independent of k as well. As g is a deterministic function, also the log odds output will be the same for all input sequences $\boldsymbol{t}_k$ and be independent of k. By the condition we have a non-zero weight $w_j(\tau) \neq 0$ for some $j \geq 2$. In this case, there are two sequences $\boldsymbol{t}_{j-1}$ (length $j-1$) and $\boldsymbol{t}_j$ (length j) consisting of only token τ, where the outputs of the additive model (GAM) follow

$$f_{\text{GAM}}(\boldsymbol{t}_j) = b + \sum_{i=1}^{j} w_i(\tau) \tag{22}$$

$$= b + \sum_{i=1}^{j-1} w_i(\tau) + w_j(\tau) \tag{23}$$

$$= f_{\text{GAM}}(\boldsymbol{t}_{j-1}) + w_j(\tau) \tag{24}$$

As we suppose $w_j(\tau) \neq 0$, it must be that $f_{\text{GAM}}(\boldsymbol{t}_j) \neq f_{\text{GAM}}(\boldsymbol{t}_{j-1})$ which is a contradiction, with the output being equal for all sequence lengths.

Multi-head attention. In the case of multiple heads, we have

$$F(\boldsymbol{t}) = \Delta\left(\text{cls}(\boldsymbol{h}_r^{(1)})\right) \tag{25}$$

$$= \Delta\left(\text{cls}\left(\text{ffn}(\boldsymbol{h}_r^{(0)} + \boldsymbol{P}_{h=1}(\boldsymbol{s}_r^{h=1}) + \boldsymbol{P}_{h=2}(\boldsymbol{s}_r^{h=2}) + \ldots + \boldsymbol{P}_{h=1}(\boldsymbol{s}_r^{h=H}))\right)\right) \tag{26}$$

$$= g(\boldsymbol{h}_r^{(0)}, \boldsymbol{s}_r^{h=1}, \ldots, \boldsymbol{s}_r^{h=H}) \tag{27}$$

As before, we can make the same argument, if we show that all inputs to g are the same. This is straightforward, as we can extend the argument made for one head for every head, because none of the head can differentiate between the sequence lengths. The first input will still correspond to $\boldsymbol{h}_r^{(0)} = \boldsymbol{e}(\tau)$, which results in the same contradiction. □

B.3 Corollary: Transformers cannot represent linear models

Corollary B.2 (Transformers cannot represent linear models)**.** *Let the context window be $C > 2$ and suppose the same model as in Proposition 4.1. Let $w : \mathcal{V} \to \mathbb{R}$ be any weighting function that is independent of the token position with $w(\tau) \neq 0$ where for at least one token $\tau \in \mathcal{V}$. Then, the single layer transformer cannot represent the linear model*

$$F([t_1, t_2, \ldots, t_N]) = b + \sum_{i=1}^{N} w(t_i). \tag{28}$$

Proof. This can be seen by setting $w_i \equiv w$ for every i in Proposition 4.1. With $w(\tau) \neq 0$, the condition from Proposition 4.1, i.e., having one w_i with $w_i(\tau) \neq 0$ for $i \geq 2$ is fulfilled as well such that the result of the proposition as well. □

This statement has a strong implication on the capabilities of transformers as it shows that they struggle to learn linear models.

B.4 Proof of Corollary 4.3

Corollary B.3 (Corollary 4.3 in the main paper)**.** *Under the same conditions as in Proposition 4.1, a stack of multiple transformer blocks as in the model F neither has a parametrization sufficient to represent the additive model.*

Proof. We show the result by induction with the help of a lemma.

Lemma: Suppose a set S of sequences. If (1) for every sequence $\boldsymbol{t} \in S$ the input matrix $\boldsymbol{H}^{(l)} = [\boldsymbol{h}_1^{(l)}, \ldots, \boldsymbol{h}_{|\boldsymbol{t}|}^{(l)}]$ will consist of input embeddings that are identical for each token i, and (2) single input embeddings also have the same value for every sequence $\boldsymbol{t} \in S$, in the output $\boldsymbol{H}^{(l+1)}$ (1) the output embeddings will be identical for all tokens i and (2) they will have equal value for all the sequences $\boldsymbol{t} \in S$ considered before.

For the encoder-only architecture, the proof from Proposition 4.1 holds analogously for each token output embedding (in the previous proof, we only considered the output embedding at the classification token r). Without restating the full proof the main steps consist of

- considering same-token sequences of variable length
- showing the attention to be equally distributed across tokens, i.e., $\alpha_{ij} = 1/|\boldsymbol{t}|$
- showing the value vectors $\boldsymbol{v}_i$ to be equal because they only depend on the input embeddings which are equal
- concluding that the output will be equal regardless of the number of inputs

This shows that for each sequence $\boldsymbol{t} \in S$, the output at token i remains constant. To show that all tokens i result in the same output, we observe that the the only dependence of the input token to the output is through the query, which however is also equivalent if we have the same inputs.

For the decoder-only architecture, for token i, the attention weights are taken only up to index i resulting in a weight of $\frac{1}{i}$ for each previous token and a weight of 0 (masked) for subsequent ones. However, with the sum also being equal to 1 and the value vectors being equivalent, there is no difference in the outcome. This proves the lemma.

Having shown this lemma, we consider a set S of two sequences $S = \{\boldsymbol{t}_{j-1}, \boldsymbol{t}_j\}$ where $\boldsymbol{k}_{j-1}$ contains $j-1$ repetitions of token τ and $\boldsymbol{t}_j$ contains j repetitions of token τ. We chose $j \geq 2, \tau$ such that $w_j(\tau) \neq 0$, which is possible by the conditions of the theorem. We observe that for $\boldsymbol{H}^{(0)}$, the embeddings are equal for each token and their value is the same for both sequences. We then apply the lemma for layers $1, \ldots, L$, resulting in the output embeddings of $\boldsymbol{H}^{(L)}$ being equal for each token, and most importantly identical for $\boldsymbol{t}_{j-1}$ and $\boldsymbol{t}_j$. As we perform the classification by $F(\boldsymbol{t}) = \Delta\left(\text{cls}\left(\boldsymbol{h}_r^{(L)}\right)\right)$, this output will also not change with the sequence length. This result can be used to construct the same contradiction as in the proof of Proposition 4.1. □

B.5 Proof of Proposition 5.1

Proposition B.4 (Transformers can easily fit SLALOM models)**.** *For any map s, v and a transformer with an embedding size $d, d_h \geq 3$, there exists a parameterization of the transformer to reflect the SLALOM model in Equation* (4)*.*

Proof. We can prove the theorem by constructing a weight setup to reflect this map. We let the embedding $\boldsymbol{e}(\tau)$ be given by

$$\boldsymbol{e}(\tau) = [s(\tau), v(\tau), 0, 0, \ldots, 0]. \tag{29}$$

We then set the key map matrix K to be

$$\boldsymbol{W}_k = \boldsymbol{0} \tag{30}$$
$$\boldsymbol{b}_k = [1, 0, \ldots, 0]. \tag{31}$$

such that we have

$$\boldsymbol{W}_k \boldsymbol{e}(\tau) + \boldsymbol{b}_k = [1, 0, \ldots, 0]. \tag{32}$$

For the query map we can use

$$\boldsymbol{W}_q = \boldsymbol{I} \tag{33}$$
$$\boldsymbol{b}_q = \boldsymbol{0} \tag{34}$$

such that

$$\boldsymbol{W}_v \boldsymbol{e}(\tau) + \boldsymbol{b}_v = [s(\tau), v(\tau), 0, \ldots, 0]. \tag{35}$$

This results in the non-normalized attention scores for query $\tau \in \mathcal{V}$ and key $\theta \in \mathcal{V}$

$$a(t_i, t_j) = (\boldsymbol{W}_q \boldsymbol{e}(t_i) + \boldsymbol{b}_q)^\top (\boldsymbol{W}_k \boldsymbol{e}(t_j) + \boldsymbol{b}_k) = s(t_j) \tag{36}$$

We see that regardless of the query token, the pre-softmax score will be $s(\theta)$. For the value scores, we perform a similar transform with

$$\boldsymbol{W}_v = \text{diag}\left([0, 1, 0, \ldots, 0]\right) \tag{37}$$
$$\boldsymbol{b}_v = \boldsymbol{0} \tag{38}$$

such that

$$\boldsymbol{v}_i = \boldsymbol{W}_v \boldsymbol{e}(t_i) + \boldsymbol{b}_v = [0, 0, v(t_i), 0, \ldots, 0]. \tag{39}$$

We then obtain

$$\boldsymbol{s}_r = \sum_{t_i \in \boldsymbol{t}} a_{ri} \boldsymbol{v}_i = \sum_{t_i \in \boldsymbol{t}} \text{softmax}_i[s(t_1), \ldots, s(t_{|\boldsymbol{t}|})] \boldsymbol{v}_i \tag{40}$$

$$= \left[0, 0, \sum_{t_i \in \boldsymbol{t}} \alpha_i(\boldsymbol{t}) v(t_i), \ldots, 0\right]^\top \tag{41}$$

We saw that the final output can be represented by

$$F(\boldsymbol{t}) = \Delta\left(\text{cls}\left(\text{ffn}\left(\boldsymbol{e}(t_0) + P(\boldsymbol{s}_r)\right)\right)\right) \tag{42}$$

The projection operator is linear, which can set to easily forward in input by setting $\boldsymbol{P} \equiv \boldsymbol{I}$. Due to the skip connection of the feed-forward part, we can easily transfer the second part through the first ffn part. In the classification part, we output the third component and zero by applying the final weight matrix

$$\boldsymbol{W}_{class} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix} \tag{43}$$

and a bias vector of $\mathbf{0}$.

Multiple Heads. Multiple heads can represent the pattern by choosing $P = \boldsymbol{I}$ for one head and choosing $P = \mathbf{0}$ for the other heads.

Multiple Layers. We can extend the argument to multiple layers by showing that the input vectors can just be forwarded by the transformer. This is simple by setting $\boldsymbol{P} \equiv \mathbf{0}$, the null-map, which can be represented by a linear operator. We then use the same classification hat as before. □

B.6 Proof of Proposition 5.2

Proposition B.5 (Proposition 5.2. in the main paper)**.** *Suppose query access to a model G that takes sequences of tokens $\boldsymbol{t}$ with lengths $|\boldsymbol{t}| \in 1, \ldots, C$ and returns the log odds according to a non-constant SLALOM on a vocabulary $\mathcal{V}$ with unknown parameter maps $s : \mathcal{V} \to \mathbb{R}$, $v : \mathcal{V} \to \mathbb{R}$. For $C \geq 2$, we can recover the true maps s, v with $2|\mathcal{V}| - 1$ queries (forward passes) of F.*

Proof. We first compute $G([\tau]), \forall \tau \in \mathcal{V}$. We know that for single token sequences, all attention is on one token, i.e., $(\alpha_i = 1)$ and we thus have

$$G([\tau]) = v(\tau) \tag{44}$$

We have obtained the values scores v for each token through $|\mathcal{V}|$ forward passes. To identify the token importance scores s, we consider token sequences of length 2.

We first note that if the SLALOM is non-constant and $|\mathcal{V}| > 1$, for every token $\tau \in \mathcal{V}$, we can find another token θ for which $v(\tau) \neq v(\theta)$. This can be seen by contradiction: If this would not be the case, i.e., we cannot find a token ω with a different value $v(\omega)$, all tokens have the same value and the SLALOM would have to be constant. For $|\mathcal{V}| = 1$, SLALOM is always constant and does not fall under the conditions of the theorem.

We now select an arbitrary reference token $\theta \in \mathcal{V}$. We select another token $\hat{\theta}$ for which $v(\hat{\theta}) \neq v(\theta)$. By the previous argument such a token always exists if the SLALOM is non-constant. We now compute relative importances w.r.t. θ that we refer to as η_θ. We let $\eta_\theta(\tau) = s(\tau) - s(\theta)$ denote the difference of the importance between the importance scores of tokens $\tau, \theta \in \mathcal{V}$. We set $\eta_\theta(\theta) = 0$

We start with selecting token $\tau = \hat{\theta}$ and subsequently use each other token $\tau \neq \theta$ to perform the following steps **for each** $\tau \neq \theta$:

1. Identify reference token $\hat{\tau}$. We now have to differentiate two cases: If $v(\tau) = v(\theta)$, we select $\hat{\tau} = \hat{\theta}$ as the reference token. If $v(\tau) \neq v(\theta)$, we select $\hat{\tau} = \theta$ as the reference token. By doing so, we will always have $v(\hat{\tau}) \neq v(\tau)$.

2. Compute $G([\tau, \hat{\tau}])$. We now compute $G([\tau, \hat{\tau}])$. From the model's definition, we know that

$$
\begin{aligned}
G([\tau, \hat{\tau}]) =& \frac{\exp(s(\tau))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))} v(\tau) && (45)\\
&+ \frac{\exp(s(\hat{\tau}))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))} v(\hat{\tau}) && (46)\\
=& \frac{\exp(s(\tau))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))} G([\tau]) && (47)\\
&+ \frac{\exp(s(\hat{\tau}))}{\exp(s(\tau)) + \exp(s(\theta))} G([\hat{\tau}]) && (48)\\
=& \frac{\exp(s(\tau))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))} G([\tau])) && (49)\\
&+ \left(1 - \frac{\exp(s(\tau))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))}\right) G([\hat{\tau}]) && (50)
\end{aligned}
$$

which we can rearrange to

$$
G([\tau, \hat{\tau}]) - G([\hat{\tau}]) = \frac{\exp(s(\tau))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))} \left(G([\tau]) - G([\hat{\tau}])\right) \tag{51}
$$

and finally to

$$
\frac{\exp(s(\tau))}{\exp(s(\tau)) + \exp(s(\hat{\tau}))} = \frac{G([\tau, \hat{\tau}]) - G([\hat{\tau}])}{G([\tau]) - G([\hat{\tau}])} \coloneqq g(\tau, \hat{\tau}) \tag{52}
$$

and

$$
\begin{aligned}
\frac{1}{1 + \frac{\exp(s(\hat{\tau}))}{\exp(s(\tau))}} &= g(\tau, \hat{\tau}) && (53)\\
\Leftrightarrow \frac{1}{g(\tau, \hat{\tau})} &= 1 + \frac{\exp(s(\hat{\tau}))}{\exp(s(\tau))} && (54)\\
\Leftrightarrow \log\left(\frac{1}{g(\tau, \hat{\tau})} - 1\right) &= s(\tau) - s(\hat{\tau}) \coloneqq d(\tau, \hat{\tau}) && (55)
\end{aligned}
$$

This allows us to express the importance of every token $\tau \in \mathcal{V}$ relative to the base token $\hat{\tau}$.

3. Compute importance relative to θ, i.e., $\eta_\theta(\tau)$. In case we selected $\hat{\tau} = \theta$, we set $\eta_\theta(\tau) = d(\tau, \hat{\tau}) = s(\tau) - s(\theta)$. In case we selected $\hat{\tau} = \hat{\theta}$, we set

$$
\eta_\theta(\tau) = d(\tau, \hat{\tau}) - d(\hat{\theta}, \theta) = s(\tau) - s(\hat{\theta}) + (s(\hat{\theta}) - s(\theta)) = s(\tau) - s(\theta) \tag{56}
$$

The value of $d(\hat{\theta}, \theta)$ is already known from the first iteration of the loop, where we consider $\tau = \hat{\theta}$ (and needs to be computed only once).

Having obtained a value of $\eta_\theta(\tau)$ for each token $\tau \neq \theta$, with $|\mathcal{V}| - 1$ forward passes, we can then use the normalization in constraint to solve for $s(\theta)$ as in

$$
\sum_{\tau \in \mathcal{V}} (\eta_\theta(\tau) + s(\theta)) =^! 0 \tag{57}
$$

such that we obtain

$$
s(\theta) = \frac{\sum_{\tau \in \mathcal{V}} \eta_\theta(\tau)}{|\mathcal{V}|} \tag{58}
$$

We can plug this back in to obtain the values for all token importance scores $s(\tau) = s(\theta) + \eta_\theta(\tau)$. We have thus computed the maps s and v in $2|\mathcal{V}| - 1$ forward passes, which completes the proof. □

B.7 Relating SLALOM to other attribution techniques.

Local Linear Attribution Scores. We can consider the following weighted model:

$$F(\boldsymbol{\lambda}) = \frac{\sum_{t_i \in \boldsymbol{t}} \lambda_i \exp(s(t_i)) v(t_i)}{\sum_{t_i \in \boldsymbol{t}} \lambda_i \exp(s(t_i))} \tag{59}$$

where $\lambda_i = 1$ if a token is present and $\lambda_i = 0$ if it is absent. We observe that setting $\lambda_i = 0$ has the desired effect of making the output of the weighted model equivalent to that of the unweighted SLALOM on a sequence without this token.

Taking the derivative at $\boldsymbol{\lambda} = \mathbf{1}$ results in

$$\frac{\partial F}{\partial \lambda_i} = \frac{\exp(s(t_i)) v(t_i) \left(\sum_{t_j \in \boldsymbol{t}, j \neq i} \lambda_j \exp(s(t_j)) \right)}{\left(\sum_{t_j \in \boldsymbol{t}} \lambda_j \exp(s(t_j)) \right)^2} \tag{60}$$

$$- \frac{\exp(s(t_i)) \left(\sum_{t_j \in \boldsymbol{t}, j \neq i} \lambda_j \exp(s(t_j)) v(t_j) \right)}{\left(\sum_{t_j \in \boldsymbol{t}} \lambda_j \exp(s(t_j)) \right)^2} \tag{61}$$

Plugging in $\boldsymbol{\lambda} = \mathbf{1}$, and using $\alpha_i(\boldsymbol{t}) = \frac{\exp(s(t_i))}{\sum_{t_j \in \boldsymbol{t}} \lambda_j \exp(s(t_j))}$ we obtain

$$\left. \frac{\partial F}{\partial \lambda_i} \right|_{\boldsymbol{\lambda} = \mathbf{1}} = \alpha_i \left(v(t_i)(1 - \alpha_i(\boldsymbol{t})) - (F(\mathbf{1}) - \alpha_i v(t_i)) \right) \tag{62}$$

$$= \alpha_i \left((v(t_i) - \alpha_i v(t_i))) - (F(\mathbf{1}) - \alpha_i v(t_i)) \right) \tag{63}$$

$$= \alpha_i \left(v(t_i) - F(\mathbf{1}) \right) \tag{64}$$

Noting that $\alpha_i = \frac{\exp(s(t_i))}{R}$, where R and $F(\mathbf{1})$ are independent of i, we obtain

$$\left. \frac{\partial F}{\partial \lambda_i} \right|_{\boldsymbol{\lambda} = \mathbf{1}} \propto v(t_i) \exp(s(t_i)), \tag{65}$$

which can be used to rank tokens according the locally linear attributions. We refer to this expression as linearized SLALOM scores ("lin").

Shapley Values. We can convert SLALOM scores to Shapley values $\phi(i)$ using the explicit formula:

$$\phi(i) = \frac{1}{n} \sum_{S \subset [N] \setminus \{i\}} \binom{n-1}{|S|} (F(S \cup \{i\}) - F(S)) \tag{66}$$

$$= \frac{1}{n} \sum_{S \subset [N] \setminus \{i\}} \binom{n-1}{|S|} \left(F(S \cup \{i\}) - \frac{F(S \cup \{i\}) - \alpha_i v_i}{1 - \alpha_i} \right) \tag{67}$$

$$= \frac{1}{n} \sum_{S \subset [N] \setminus \{i\}} \binom{n-1}{|S|} \left(\frac{\alpha_i (v_i - F(S \cup \{i\}))}{1 - \alpha_i} \right) \tag{68}$$

$$\left(\frac{\alpha_i (v_i - F(\mathbf{1}))}{1 - \alpha_i} \right) \tag{69}$$

However, computing this sum remains usually intractable, as the number of coalitions grows exponentially. We can resort to common sampling approaches (Castro et al., 2009; Maleki et al., 2013) to approximate the sum.

C Additional Discussion and Intuition

C.1 Generalization to multi-class problems

We can imagine the following generalizing SLALOM to multi class problems as follows: We keep an importance map $s : \mathcal{V} \to \mathbb{R}$ that still maps each token to an importance score as previously. However, we now introduce a value score map $v_c : \mathcal{V} \to \mathbb{R}$ for each class $c \in \mathcal{Y}$. Additionally to requiring

$$\sum_{\tau \in \mathcal{V}} s(\tau) = 0. \tag{70}$$

we now require

$$\sum_{c \in \mathcal{Y}} v_c(\tau) = 0, \forall \tau \in \mathcal{V} \tag{71}$$

For an input sequence $\boldsymbol{t}$, the SLALOM model then computes

$$F_c(\boldsymbol{t}) = \log \frac{p(y=1|\boldsymbol{t})}{p(y=0|\boldsymbol{t})} = \sum_{\tau_i \in \boldsymbol{t}} \alpha_i(\boldsymbol{t}) v_c(t_i), \tag{72}$$

The posterior probabilities can be computed by performing a softmax operation over the F-scores, as in

$$p(y=c|\boldsymbol{t}) = \frac{\exp(F_c(\boldsymbol{t}))}{\sum_{c' \in \mathcal{Y}} \exp(F_{c'}(\boldsymbol{t}))} \tag{73}$$

We observe that this model has $\big(|\mathcal{Y}|-1\big)|\mathcal{V}|-1$ free parameters (for the two-class problem, this yields $2|\mathcal{V}|-1$ as before) and can be fitted and deployed as the two-class SLALOM without major ramifications.

C.2 Practical Considerations

Our theoretical model contains slight deviations from real-world transformers to make it amendable to theoretical analysis. To represent token order, common architectures use positional embeddings, tying the embedding vectors to the token position i. The behavior that we show in this work's analysis does however also govern transformers with positional embeddings for the following reason: While the positional embeddings could be used by the non-linear ffn part to differentiate sequences of different length in theory, our proofs show that to represent the linear model, the softmax operation must be inverted for any input sequence. This is a highly nonlinear operation and the number of possible sequences grows exponentially at a rate of $|\mathcal{V}|^C$ with the context length C. Learning-theoretic considerations (e.g., Bartlett et al., 1998) show that the number of input-output pairs the two-layer networks deployed can maximally represent is bounded by $\mathcal{O}\left(dn_{\text{hidden}} \log(dn_{\text{hidden}})\right)$, which is small ($d$=786, n_{hidden}=3072 for BERT) in contrast to the number of sequences ($C = 1024$, $|\mathcal{V}| \approx 3 \times 10^4$). We conclude that the inversion is therefore impossible for realistic setups and positional embeddings can be neglected, which is confirmed by our empirical findings.

Common models such as BERT also use a special token referred to as `CLS`-token where the classification head is placed on. In this work, we consider the `CLS` token just as a standard token in our analysis. In our empirical sections, we always append the `CLS` token as mandated by the architecture to make the sequences valid model inputs.

D Algorithm: Local SLALOM approximation

We propose two algorithms to compute local explanations for a sequence $\boldsymbol{t} = [t_1, \ldots, t_{|\boldsymbol{t}|}]$ with SLALOM scores. In particular, we use the Mean-Squared-Error (MSE) to fit SLALOM on modified sequences consisting of the individual tokens in the original sequence $\boldsymbol{t}$. To speed up the fitting we can sample a large collection of samples offline before the optimization.

D.1 Efficiently fitting SLALOM with SGD

For the efficient implementation SLALOM-`eff` given in Algorithm 1 we sample minibatches from this collection in each step and perform SGD steps on them. We perform this optimization using $b = 5000$ samples in this work. We use sequences of $n = 2$ random tokens from the sample for SLALOM-`eff`, making the forward passes through the model highly efficient.

D.2 Fitting SLALOM through iterative optimization

For the high-fidelity implementation SLALOM-`fidel` (Algorithm 2) we first use a different set of sequences and model scores to fit the surrogate: We delete up to 5 tokens from the original sequence randomly to create the estimation dataset (similar to LIME). The fitting algorithm optimized for maximum fidelity uses an iterative optimization scheme to fit SLALOM models. It works by iteratively fitting $\boldsymbol{v}$ and $\boldsymbol{s}$ to the dataset obtained. Denote by $\boldsymbol{f} \in \mathbb{R}^b$ the model scores obtained for the b input sequences $\boldsymbol{t}_i, i = 1...b$. As the SLALOM model in Equation (4) is a linear combination of the values score weighted by the normalized importance score, we can set up a matrix $\boldsymbol{A}$, where element $\boldsymbol{a}_{i,j} = \frac{\exp(s(t_i))}{\sum_{t_j \in \boldsymbol{t}_i} \exp(s(t_j))}$ provides the normalized importance for a given $\boldsymbol{s}$. We solve

$$\min_{\boldsymbol{v}} (\boldsymbol{A}\boldsymbol{v} - \boldsymbol{f})^\top (\boldsymbol{A}\boldsymbol{v} - \boldsymbol{f}), \tag{74}$$

for $\boldsymbol{v}$, which is a linear ordinary least squared problem that can be solved through the normal equation. This results in the optimal $\boldsymbol{v}$ for the given $\boldsymbol{s}$. In a second step, we keep $\boldsymbol{v}$ fixed and find better $\boldsymbol{s}$ scores. We can reformulate the equations for the samples as

$$\sum_{t_j \in \boldsymbol{t}_i} \exp(s(t_j)) v(t_j) = \left(\sum_{t_j \in \boldsymbol{t}_i} \exp(s(t_j)) \right) \boldsymbol{f}_i \tag{75}$$

$$\Leftrightarrow \sum_{t_j \in \boldsymbol{t}_i} \underbrace{\exp(s(t_j))}_{\bar{s}_j} \underbrace{(v(t_j) - \boldsymbol{f}_i)}_{e_{i,j}} = 0. \tag{76}$$

This problem can be written with a vector $\bar{\boldsymbol{s}} \in \mathbb{R}^{|\mathcal{V}|}$ and a matrix $\boldsymbol{E} \in \mathbb{R}^{b \times |\mathcal{V}|}$ and results in an optimization problem

$$\min_{\bar{\boldsymbol{s}}} (\boldsymbol{E}\bar{\boldsymbol{s}})^\top (\boldsymbol{E}\bar{\boldsymbol{s}}), \tag{77}$$

$$\text{s.t. } \hat{\boldsymbol{s}} \geq \boldsymbol{0} \tag{78}$$

$$\|\hat{\boldsymbol{s}}\|_1 \geq |\mathcal{V}| \tag{79}$$

The conditions ensure that we can obtain the original $\boldsymbol{s}$-scores as $\log \hat{\boldsymbol{s}}$ (element-wise) and that the trivial solution $\hat{\boldsymbol{s}} = \boldsymbol{0}$ is not assumed. We solve this problem using a solver implemented in `scipy.optimize.least_squares`.

E Experimental Details

In this section, we provide details on the experimental setups. We provide the full source-code in our GitHub repository.

E.1 Fitting transformers on a synthetic dataset

E.1.1 Dataset construction: Linear Dataset

We create a synthetic dataset to ensure a linear relationship between features and log odds. Before sampling the dataset, we fix a vocabulary of tokens, ground truth scores for each token, and their occurrence probability. This means that each of the possible tokens already comes with a ground-truth score w that has been manually assigned. The tokens, their respective scores w, and occurrence probabilities are listed in Table 3. Samples of the dataset are sampled in four steps that are exectued repeatedly for each sample:

Algorithm 1 Local efficient SLALOM approximation (SLALOM-`eff`)

Require: Sequence $\boldsymbol{t}$, trained model F (outputs log odds), random sample length n, learning rate λ, batch size r, sample pool size b, number of steps c
Initialize $v(t_i) = 0$, $s(t_i) = 0 \ \ \forall$ unique $t_i \in \boldsymbol{t}$
$B \leftarrow b$ samples of random sequences of length n obtained through uniform sampling of unique tokens in $\boldsymbol{t}$.
Precompute $F(B[i]), i = 1, \ldots, b$ *# perform model forward-pass for each sample in pool*
steps $\leftarrow 0$
while steps $< c$ **do**
  $B' \leftarrow$ minibatch of r samples uniformly sampled from the sample pool B
  loss $\leftarrow \frac{1}{r}\sum_{k=1}^{r}\left(F(B'[k]) - \text{SLALOM}_{v,s}(B'[k])\right)^2$ *# compute MSE btw. F and SLALOM using precomputed models outputs $F(B')$*
  $v \leftarrow v - \lambda\nabla_v$loss *# Back-propagate loss to update SLALOM parameters*
  $s \leftarrow s - \lambda\nabla_s$loss
  steps $\leftarrow$ steps $+ 1$
end while
return v, $s - \text{mean}(s)$ *# normalize s to zero-mean*

Algorithm 2 Local high-fidelity SLALOM approximation (SLALOM-`fidel`)

Require: Sequence $\boldsymbol{t}$, trained model F (outputs log odds), max. number of deletions n, learning rate λ, batch size r, sample pool size b, number of steps c
Initialize $\boldsymbol{s} = 0$, $\boldsymbol{s} = 0 \ \ \forall$ unique $t_i \in \boldsymbol{t}$
$B \leftarrow b$ samples of random sequences of length n obtained through deleting up to n tokens randomly from $\boldsymbol{t}$.
Precompute $F(B[i]), i = 1, \ldots, b$ *# perform model forward-pass for each sample in pool*
steps $\leftarrow 0$
while steps $< c$ **do**
  $\boldsymbol{v} = \arg\min_{\boldsymbol{v}'}\sum_{i=1}^{b}\left(F(B[i]) - \text{SLALOM}_{\boldsymbol{v}',\boldsymbol{s}}(B[i])\right)^2$ *# Fix $\boldsymbol{s}$ and optimize $\boldsymbol{v}$, OLS problem*
  $\boldsymbol{s} = \arg\min_{\boldsymbol{s}'}\sum_{i=1}^{b}\left(F(B[i]) - \text{SLALOM}_{\boldsymbol{v},\boldsymbol{s}'}(B[i])\right)^2$ *# Fix $\boldsymbol{v}$ and optimize $\boldsymbol{s}$, Quadratic problem*
  steps $\leftarrow$ steps $+ 1$
end while
return $\boldsymbol{v}$, $\boldsymbol{s} - \text{mean}(\boldsymbol{s})$ *# normalize s to zero-mean*

1. A sequence length $|\boldsymbol{t}| \sim \text{Bin}(p = 0.5, n{=}30)$ is binomially distributed with an expected value of 15 tokens and a maximum of 30 tokens

2. We sample $|\boldsymbol{t}|$ tokens independently from the vocabulary according to their occurrence probability (Table 3)

3. Third, having obtained the input sequence, we can evaluate the linear model by summing up the scores of the individual tokens in a sequence:

$$F(\boldsymbol{t}) = F([t_1, t_2, \ldots, t_{|\boldsymbol{t}|}]) = \sum_{i=1}^{|\boldsymbol{t}|} w(t_i). \tag{80}$$

4. Having obtained the log odds ratio for this sample F(t), we sample the labels according to this ratio. We have $p(y = 1)/p(y = 0) = \exp(F(\boldsymbol{t}))$, which can be rearranged to $p(y = 1) = \frac{\exp(F(\boldsymbol{t}))}{1+\exp(F(\boldsymbol{t}))}$. We sample a binary label y for each sample according to this probability.

The tokens appear independently with the probability $p_{\text{occurrence}}$ given in the table.

E.1.2 Dataset Construction: SLALOM dataset

We resort to a second synthetic dataset to study the recovery property for the SLALOM. This required a SLALOM relation between the data features and the labels. To find a realistic distribution of scores, we

word	"the"	"we"	"movie"	"watch"	"good"	"best"	"perfect"	"ok"	"bad"	"worst"
linear weight w	0.0	0.0	0.0	0.0	0.6	1.0	1.5	-0.6	-1.0	-1.5
$p_{\text{occurrence}}$	1/6	1/6	1/6	1/6	1/15	1/20	1/20	1/15	1/20	1/20

Table 3: Tokens in the linear dataset with their corresponding weight

compute a BoW importance scores for input tokens of the BERT model on the IMDB dataset by counting the class-wise occurrence probabilities. We select 200 tokens randomly from this dataset. We use these scores as value scores v but multiply them by a factors of 2 as many words have very small BoW importances. In realistic datasets, we observed that value scores v are correlated with the importance scores s. Therefore, we sample

$$s(\tau) \sim 5\left(v(\tau)^{\frac{3}{2}}\right) + \frac{1}{2}\mathcal{N}(0,1)\,, \tag{81}$$

which results in the value/importance distribution given in Figure 8. We assign each word an equal occurrence probability and sample sequences of words at uniformly distributed lengths in random $[1, 30]$. After a sequence is sampled, labels are subsequently sampled according to the log odds ratio of the SLALOM.

E.2 Post-hoc fitting of surrogate models

We train the models on this dataset for 5 epochs, where one epoch contains 5000 samples at batch size of 20 using default parameters otherwise.

For the results in Figure 3, we query the models with sequences that contain growing numbers of the work perfect, i.e. ["perfect", "perfect", ...]. We prepend a CLS token for the BERT models.

For the results in Figure 4(a,b), we then sample 10000 new samples from the linear synthetic dataset (having the same distribution as the training samples) and forward them through the trained transformers. The model log odds score together with the feature vectors are used to train the different surrogate models, linear model, GAM, and SLALOM. For the linear model, we fit an OLS on the log odds returned by the model. We use the word counts for each of the 10 tokens as a feature vector. The GAM provides the possibility to assign each token a different weight according to its position in the sequence. To this end, we use a different feature vector of length $30 \cdot 10$. Each feature corresponds to a token and a position and is set to one if the token i is present at this position, and set to 0 otherwise. We then fit a linear model using regularized (LASSO) least squares with a small regularization constant of $\lambda = 0.01$ because the system is numerically unstable in its unregularized form.

E.3 Recovering SLALOMs from observations

To obtain the results in Figure 4(c,d), we train the transformer models on $20 \cdot 20000$ samples from the second synthetic dataset (SLALOM dataset). When using a smaller vocabulary size, we only sample the sequences out of the first $|\mathcal{V}|$ possible tokens (but keeping the ground truth scores identical).

E.4 Training Details for real-world data experiments

Training details. In these experiments, we use the IMDB (Maas et al., 2011) and Yelp (Asghar, 2016) datasets to train transformer models on. Specifically, the results in Table 1a are obtained by training 2-layer versions of BERT, DistilBERT and GPT-2 with on 5000 samples from the IMDB dataset for 5 epochs, respectively. We did not observe significant variation in terms of number of layers, so we stick to the simpler models for the final experiments. For the experiments in Table 1b we train and use 6-layer versions of the above models for 5 epochs on 5000 samples of the Yelp dataset. We report the accuracies of these models in Table 4 and additional hyperparameters in Table 5.

SLALOM vs. Naïve Bayes Ground Truth. To arrive at the Spearman rank-correlations between SLALOM importance scores s, value scores v and their combination $(\exp(s) \cdot v)$ with a ground truth, we fit

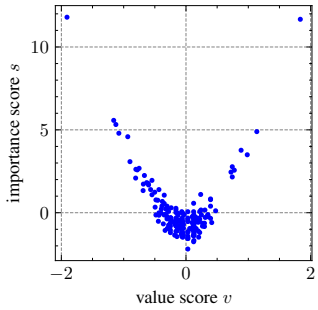


Figure 8: Score distribution for the tokens for the analytical SLALOM used in the recovery experiment.

dataset	DistilBERT	BERT	GPT-2
IMDB	0.88	0.90	0.74
Yelp	0.86	0.88	0.88

Table 4: **Accuracies of models used in this paper.** For IMDB ($|trainset| = 5000$), we use 2-layer versions of the models. For Yelp, ($|trainset| = 5000$), we use 6-layer versions of the models. For both datasets, the $|testset| = 100$. The models are trained for 5 epochs after which we found the accuracy of the model on the test set to have converged.

SLALOM on each of the trained models and use a Naïve-Bayes model for ground truth scores. The model is given as follows:

$$\log \frac{p(y=1|\boldsymbol{t})}{p(y=0|\boldsymbol{t})} = \log \frac{p(y=1)}{p(y=0)} + \sum_{t_i \in \boldsymbol{t}} \log \frac{p(t_i|y=1)}{p(t_i|y=0)} \tag{82}$$

We obtain $\frac{p(t_i|y=1)}{p(t_i|y=0)}$ by counting class-wise word frequencies, such that we obtain a linear score w for each token τ given by $w(\tau) = \frac{(\#\text{occ. of } \tau \text{ in class } 1)+\alpha}{(\#\text{occ. of } \tau \text{ in class } 0)+\alpha}$. We use Laplace smoothing with $\alpha = 40$. The final correlations are computed over a set of 50 random samples, where we observe good agreement between the Naïve Bayes scores, and the value and linearized SLALOM scores, respectively. Note that the importance scores are considered unsigned, such that we compute their correlation with the absolute value of the Naive Bayes scores.

SLALOM vs. Human Attention. The Yelp Human Attention (HAT) (Sen et al., 2020) dataset consists of samples from the original Yelp review dataset, where for each review real human annotators have been asked to select the words they deem important for the underlying class (i.e. positive or negative). This results in binary attention vectors, where each word either is or is not important according to the annotators. Since each sample is processed by multiple annotators, we use the consensus attention map as defined in Sen et al. (2020), requiring agreement between annotators for a token to be considered important to aggregate them into one attention vector per sample. Since HAT, unlike SLALOM, operates on a word level, we map each word's human attention to each of its tokens (according to the employed model's specific tokenizer).
To compare SLALOM scores with human attention in Table 1b, we choose the AU-ROC metric, where the binary human attention serves as the correct class, and the SLALOM scores as the prediction. We observe how especially the importance scores of SLALOM are reasonably powerful in predicting human attention. Note that the human attention scores are unsigned, such that we also use absolute values for the SLALOM value scores and the linearized version of the SLALOM scores for the HAT prediction.

parameter	value
learning rate	5e-5
batch size	5
epochs	5
dataset size used	5000
number of heads	12
number of layers	2 (IMDB), 6 (Yelp)
num. parameter	31M - 124M

(a) Hyperparameters

specification	value
CPU core:	AMD EPYC 7763
Num. CPU cores	64-Core (128 threads)
GPU type used	1xNvidia A100
GPU-RAM	80GB
Compute-Hours	$\approx$ 150 h

(b) Hardware used (internal cluster)

Table 5: Overview over relevant hyperparameters and hardware

architecture	L (num.layers)	linear model	GAM	SLALOM
GPT-2	1	20.31 ± 2.02	48.78 ± 2.70	**16.92** ± 1.33
GPT-2	2	24.81 ± 3.11	54.33 ± 3.26	**22.17** ± 1.98
GPT-2	6	32.66 ± 7.60	57.08 ± 7.19	**21.59** ± 4.14
GPT-2	12	25.74 ± 4.18	54.36 ± 3.94	**20.25** ± 2.37
DistilBERT	1	28.28 ± 4.30	44.43 ± 2.22	**10.83** ± 2.13
DistilBERT	2	32.58 ± 7.75	53.87 ± 7.20	**16.82** ± 4.38
DistilBERT	6	31.49 ± 4.06	49.35 ± 3.13	**17.26** ± 3.64
DistilBERT	12	50.82 ± 9.21	71.64 ± 9.19	**27.50** ± 4.18
BERT	1	26.33 ± 1.90	43.30 ± 0.88	**7.34** ± 0.70
BERT	2	28.43 ± 3.75	48.28 ± 3.23	**9.92** ± 1.19
BERT	6	50.82 ± 6.34	68.23 ± 4.59	**23.99** ± 3.38
BERT	12	44.58 ± 13.15	51.38 ± 14.71	**18.77** ± 6.78

Table 6: MSE (×100) when fitting SLALOM to the outputs of transformer models trained on the linear dataset. SLALOM manages to describe the outputs of the transformer significantly better than other surrogate models *even if the underlying relation in the data was linear.*

F Additional Experimental Results

F.1 Fitting SLALOM as a Surrogate to Transformer outputs

We provide an additional emprical counterexample for why GAMs cannot describe the transformer output in Figure 9. The example provides additional intuition for why the GAM is insufficient to describe transformers acting like a *weighted* sum of token importances.

We report Mean-Squared Errors when fitting SLALOM to transformer models trained on the linear dataset in Table 6. These results underline that SLALOM outperforms linear and additive models when fitting them to the transformer outputs. Note that even if the original relation in the data was linear, the transformer does not represent this relation such that the SLALOM describes its output better. We present additional qualitative results for other models in Figure 10 that support the same conclusion.

F.2 Fitting SLALOM on Transformers trained on data following the SLALOM distribution

We report Mean-Squared Errors in the logit-space and the parameter-space between original SLALOM scores and recovered scores. The logit output are evaluated on a test set of 200 samples that are sampled from the original SLALOM. We provide these quantitative results in Table 7 in Table 8 for the parameter space. In logits the differences are negligibly small, and seem to decrease further with more layers. This finding highlight that a) transformers with more layers still easily fit SLALOMs and such model can be recovered in

Input Sequence	BERT score	GAM weight assignments	SLALOM weight assignments
perfect	**2.5**	Assign w_1("perfect") = 2.5	Assign v("perfect") = 2.5
perfect perfect	**2.5**	Assign w_2("perfect") = 0	Expected SLALOM output: 2.5
worst	**-2.6**	Assign w_1("worst") = −2.6	Assign v("worst") = −2.6
worst worst	**-2.5**	Assign w_2("worst") = 0.1	Expected SLALOM output: -2.6
worst perfect	**0.0**	↯ Expected GAM output w_1("perfect") + w_2("worst") = 2.6	Assign s("perfect") = s("worst") = 0 Expected SLALOM output: -0.05

Figure 9: A simple empirical counterexample for why GANs cannot describe transformer output. We report rounded scores by a real 4-layer BERT model (similar behavior was observed for other layers/architectures) and iteratively fit the GAM $F(\boldsymbol{t}) = \sum_{t_i \in \boldsymbol{t}} w_i(t_i)$ to match observed outputs on the two tokens "perfect" and "worst". We quickly arrive at a contradiction for the GAM. On the contrary, we can assign SLALOM scores that model this behavior with minor error. Because transformers behave like a weighted sum of importances, GAMs are insufficient to model their behavior. In conjunction with Figure 4(a,b) this underlines that GAMs and linear models are insufficient as surrogates.

L (num.layers)	DistilBERT	BERT	GPT-2
1	0.002 ± 0.001	0.002 ± 0.000	0.011 ± 0.009
2	0.003 ± 0.002	0.003 ± 0.002	0.017 ± 0.011
6	0.001 ± 0.001	0.011 ± 0.007	<0.001 ± 0.000
12	<0.001 ± 0.000	<0.001 ± 0.000	<0.001 ± 0.000

Table 7: MSE (×100), logit space, averaged over 5 runs

parameters space. The results on the MSE in parameter space show no clear trend, but are relatively small as well (with the largest value being MSE=0.015 (note that results in the table are multiplied by a factor of 100 for readability). Together with our quantitative results in Figure 4(c,d), this highlights that SLALOM has effective recovery properties.

F.3 Additional Results on Real-World Data

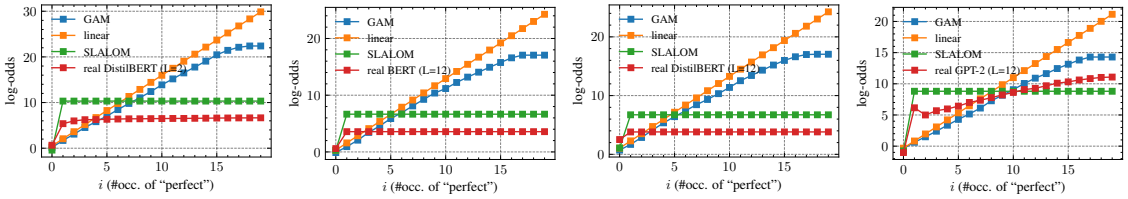
We obtain SLALOM explanations for real-world data using the procedure outlined in Algorithm 1 (SLALOM-`eff`) with sequences of length $n = 2$ and with Algorithm 2 (SLALOM-`fidel`) removing up to 5 tokens that we compare with Naive-Bayes scores and Human Attention.

F.3.1 Additional Qualitative Results

Figure 12 shows the full results from the sample used in Figure 5, where we only visualized a choice of words for readability purposes. After running SLALOM-`eff` on our trained IMDB models, we use to explain a movie review taken from the dataset, visualizing value scores v against importance scores s.

L (num.layers)	DistilBERT	BERT	GPT-2
1	0.092 ± 0.045	0.540 ± 0.301	0.940 ± 0.392
2	0.094 ± 0.049	0.368 ± 0.085	1.652 ± 0.903
6	0.124 ± 0.030	0.830 ± 0.182	0.569 ± 0.177
12	0.287 ± 0.088	0.394 ± 0.255	0.385 ± 0.126

Table 8: MSE (×100), parameter space, averaged over 5 runs



(a) DistilBERT (L=2) (b) BERT (L=12) (c) DistilBERT (L=12) (d) GPT-2 (L=12)

Figure 10: **SLALOM describes outputs of transformer models well.** Fitting SLALOM to the outputs of the models shown in Figure 3 using the synthetic dataset. We show results for a sequence containing repetitions of the token "perfect". Note however that the models were trained on a larger dataset of random sequences samples as described in Appendix E.1.1, but these sequences were chosen for visualization purposes. Results on additional models. Despite having $C/2{=}15\times$ more parameters than the SLALOM model, the GAM model does not describe the output as accurately. We provide quantitative results in Table 6.

	SLALOM-`fidel`			SLALOM-`eff`		
LM	values v	importances s	lin. SLALOM	values v	importances s	lin. SLALOM
DistilBERT	0.602 ± 0.10	0.020 ± 0.08	0.602 ± 0.10	0.692 ± 0.05	0.373 ± 0.09	0.693 ± 0.05
BERT	0.475 ± 0.12	0.031 ± 0.09	0.474 ± 0.12	0.619 ± 0.08	0.349 ± 0.09	0.626 ± 0.08
GPT-2	0.467 ± 0.17	0.017 ± 0.08	0.468 ± 0.17	0.618 ± 0.08	0.292 ± 0.10	0.619 ± 0.08

LM	LIME	SHAP	IG	Grad	LRP
Distilbert	0.691 ± 0.05	0.619 ± 0.06	-0.285 ± 0.12	-0.215 ± 0.12	0.706 ± 0.05
BERT	0.616 ± 0.08	0.554 ± 0.09	-0.125 ± 0.14	-0.123 ± 0.14	0.639 ± 0.08
GPT2	0.213 ± 0.13	0.560 ± 0.09	0.033 ± 0.13	0.031 ± 0.13	0.615 ± 0.08

Table 9: Correlation with linear Naive Bayes Scores. The scores obtained with SLALOM-`eff` (value, lin.) are better than those from LIME, SHAP and comparable to LRP scores.

F.3.2 Correlation with Naive-Bayes Scores

We compare the scores obtained with SLALOM with the scores obtained with other methods in Table 9, obtaining scores that are reliable with SLALOM-`eff` (value scores and linear scores) in particular. While SHAP achieves higher correlation on BERT, SLALOM achieves higher correlation than LIME and SHAP on all models and higher correlations than LRP for GPT-2 while obtaining slightly inferior values for the BERT-based architectures.

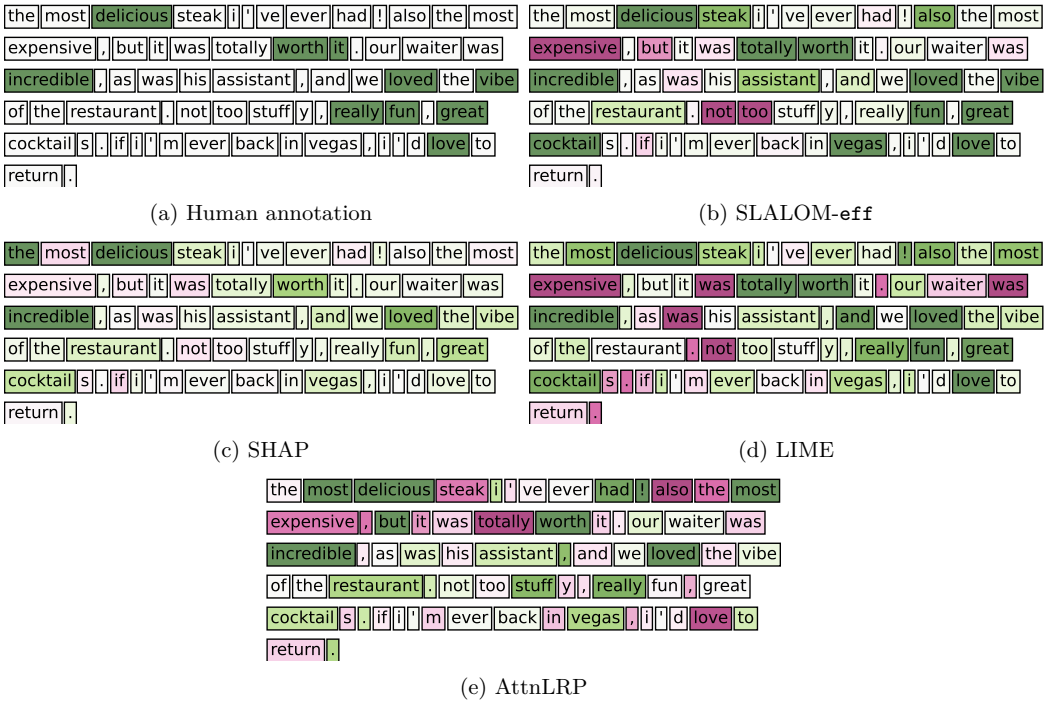
F.3.3 Human Attention

In Figure 11, we show qualitative results for a sample from the Yelp-HAT dataset. After fitting SLALOM on top of the resulting model, we can extract the importance scores given to each token in the sample. We can see that the SLALOM scores manage to identify many of the tokens which real human annotators also deemed important for this review to be classified as positive. We also show qualitative results for the other methods. However, we suggest caution when interpreting explanations visually without ground truth. We argue that (1) theoretical properties of explanations (2) comparing to a known ground truth as well as (3) consideration of metrics from different domains, e.g., faithfulness, human perspective, are required to allow for a comprehensive evaluation. This is the approach taken in our work.

We show a quantitative comparison of the scores obtained with SLALOM with the scores obtained with other methods on the comparison with Human-Attention in Table 10.

F.4 Insertion and Removal Benchmarks

It is important to verify that SLALOM scores are competitive to other methods in classical explanation benchmarks as well. We therefore ran the classical removal and insertion benchmarks with SLALOM



(a) Human annotation

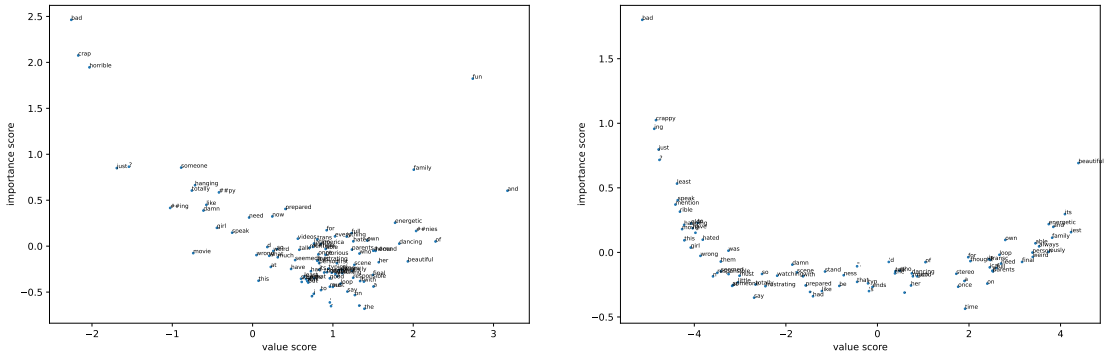
(b) SLALOM-`eff`

(c) SHAP

(d) LIME

(e) AttnLRP

Figure 11: **Qualitative comparisons of attribution maps.** We provide attribution maps for the different techniques in this figure. Many words deemed important by human annotators are likewise highlighted by SLALOM and other techniques.



(a) BERT

(b) GPT-2

Figure 12: Full scatter plots of SLALOM scores for the sample shown in the main paper (please zoom in for details). We observe that words like “bad” or “fun” get assigned high importance scores and value scores of high magnitude (albeit with different signs) by SLALOM.

LM	values v	importances s	lin.	LIME	SHAP	LRP
Bert	0.786 ± 0.01	0.807 ± 0.01	0.801 ± 0.01	0.805 ± 0.01	0.800 ± 0.01	0.813 ± 0.01
Distil-BERT	0.688 ± 0.01	0.681 ± 0.01	0.686 ± 0.01	0.702 ± 0.01	0.668 ± 0.01	0.703 ± 0.01
GPT-2	0.674 ± 0.01	0.685 ± 0.01	0.683 ± 0.01	0.632 ± 0.01	0.671 ± 0.01	0.699 ± 0.01

Table 10: Comparison of techniques to predict Human Attention. SLALOM-eff (importances) perform better than SHAP, comparably to LIME and slightly below LRP.

	SLALOM-`fidel`		SLALOM-`eff`						
LM	v-scores	lin.	v-scores	lin.	LIME	SHAP	IG	Grad	LRP
BERT	0.893 ± 0.012	0.901 ± 0.012	0.881 ± 0.010	0.885 ± 0.010	0.875 ± 0.012	0.881 ± 0.011	0.084 ± 0.010	0.069 ± 0.008	0.852 ± 0.019
DistilBERT	0.841 ± 0.014	0.854 ± 0.013	0.888 ± 0.008	0.886 ± 0.008	0.838 ± 0.012	0.864 ± 0.009	0.143 ± 0.012	0.131 ± 0.012	0.865 ± 0.011
GPT-2	0.837 ± 0.013	0.844 ± 0.013	0.782 ± 0.013	0.784 ± 0.012	0.479 ± 0.024	0.859 ± 0.012	0.289 ± 0.021	0.269 ± 0.020	0.833 ± 0.013
average	0.857 ± 0.013	0.866 ± 0.013	0.851 ± 0.010	0.852 ± 0.010	0.731 ± 0.016	0.868 ± 0.011	0.172 ± 0.014	0.156 ± 0.013	0.850 ± 0.014

(a) IMDB: Area-Over Perturbation Curve (deletion, higher is better)

	SLALOM-`fidel`		SLALOM-`eff`						
LM	v-scores	lin.	v-scores	lin.	LIME	SHAP	IG	Grad	LRP
BERT	0.015 ± 0.005	0.011 ± 0.005	0.011 ± 0.005	0.011 ± 0.005	0.017 ± 0.009	0.012 ± 0.005	0.224 ± 0.024	0.214 ± 0.022	0.010 ± 0.005
DistilBERT	0.018 ± 0.006	0.019 ± 0.006	0.032 ± 0.009	0.032 ± 0.009	0.014 ± 0.005	0.020 ± 0.005	0.250 ± 0.027	0.249 ± 0.026	0.017 ± 0.009
GPT-2	0.033 ± 0.007	0.032 ± 0.007	0.045 ± 0.007	0.045 ± 0.007	0.129 ± 0.019	0.021 ± 0.004	0.251 ± 0.024	0.244 ± 0.024	0.039 ± 0.007
average	0.022 ± 0.006	0.021 ± 0.006	0.029 ± 0.007	0.029 ± 0.007	0.053 ± 0.011	0.018 ± 0.005	0.242 ± 0.025	0.236 ± 0.024	0.022 ± 0.007

(b) Yelp: Area-Over Perturbation Curve (insertion, lower is better)

	SLALOM-`fidel`		SLALOM-`eff`						
LM	v-scores	lin.	v-scores	lin.	LIME	SHAP	IG	Grad	LRP
GPT-2	0.747 ± 0.024	0.753 ± 0.024	0.726 ± 0.021	0.727 ± 0.021	0.444 ± 0.028	0.849 ± 0.015	0.292 ± 0.026	0.290 ± 0.026	0.740 ± 0.025
BERT	0.657 ± 0.038	0.667 ± 0.038	0.865 ± 0.012	0.863 ± 0.013	0.797 ± 0.022	0.859 ± 0.013	0.249 ± 0.028	0.281 ± 0.029	0.855 ± 0.017
DistilBERT	0.645 ± 0.033	0.642 ± 0.033	0.813 ± 0.017	0.813 ± 0.018	0.746 ± 0.025	0.854 ± 0.013	0.201 ± 0.026	0.243 ± 0.028	0.768 ± 0.024
average	0.683 ± 0.032	0.687 ± 0.032	0.801 ± 0.017	0.801 ± 0.017	0.663 ± 0.025	0.854 ± 0.014	0.247 ± 0.027	0.271 ± 0.028	0.788 ± 0.022

(c) Yelp: Area-Over Perturbation Curve (deletion, higher is better)

Table 11: Additional results for removal/insertion tests: We show results on the IMDB dataset for removal as well as insertion and removal on the Yelp datset.

compared to baselines such as LIME, SHAP, Grad (Simonyan et al., 2013), and Integrated Gradients (IG, Sundararajan et al., 2017). For the insertion benchmarks, the tokens with the highest attributions are inserted to quickly obtain a high prediction score to the target class. For the deletion benchmark, the tokens with the highest attributions are deleted from the sample to obtain a low score for the target class. We subsequently delete/insert more tokens and compute the "Area Over the Perturbation Curve" (AOPC) as in DeYoung et al. (2020), which should be high for deletion and low for insertion. In addition to the insertion results in Table 1c, the removal results are shown in Table 11a. We show results for the Yelp dataset in Table 11b and Table 11c. We claim that linear SLALOM scores perform on par with LIME and SHAP in this benchmark, but do not always outperform them in this metric. For surrogate techniques (LIME, SHAP, SLALOM) we use 5000 samples each.

F.5 Error Analysis for non-transformer models

We also investigate the behavior of SLALOM for models that do not precisely follow the architecture described in the Analysis section of this paper. In the present work, we consider an attribution method that is specifically catered towards the transformer architecture, which is the most prevalent in sequence classification. We advise caution when using our model when the type of underlying LM is unknown. In this case, model-agnostic interpretability methods may be preferred.

However, we investigate this issue further: We applied our SLALOM-`eff` approach to a simple, non-transformer sequence classification model on the IMDB dataset, which is a three-layer feed-forward network based on a

	SHAP	LIME	lin. SLALOM-`eff`
Insertion (lower)	0.0120 ± 0.005	0.0053 ± 0.003	0.0039 ± 0.001
Deletion (higher)	0.8022 ± 0.026	0.9481 ± 0.005	0.9601 ± 0.004

Table 12: AOPC explantion fidelity metrics for the Fully Connected TF-IDF model. The scores highlight that SLALOM can also provide faithful explanations for non-transformer models due to its general expressivity.

(a) DistilBERT

Approach / # samples	1000	2000	5000	10000
SHAP	2.35 ± 0.01	4.62 ± 0.02	11.56 ± 0.03	23.08 ± 0.08
LIME	0.80 ± 0.04	1.58 ± 0.07	3.93 ± 0.19	8.04 ± 0.39
SLALOM-`fidel`	0.74 ± 0.03	1.42 ± 0.06	3.77 ± 0.24	7.95 ± 0.41
SLALOM-`eff`	0.42 ± 0.01	0.80 ± 0.01	2.03 ± 0.01	4.13 ± 0.02
LRP	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
IG	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Grad	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00

(b) BLOOM-7B

Runtime (s)	SHAP	LIME	IG	Grad	SLALOM-`eff`
1000 Samples	25.04 ± 2.37	15.73 ± 2.37	0.29 ± 0.04	0.06 ± 0.01	0.62 ± 0.02

Table 13: Runtime results for computing different explanations. SLALOM is substantially more efficient that other surrogate model explanations (e.g., LIME, SHAP). Gradient-based explanations can be computed even quicker, but are very noisy and require backward passes. Runtimes are given in seconds (s).

TF-IDF representation of the inputs. We compute the insertion and deletion Area-over-perturbation-curve metrics that are given in Table 12.

These results show that due to its general expressivity, the SLALOM model also succeeds to provide explanations for non-transformer models that outperform LIME and SHAP in the removal and insertion tests. We also invite the reader to confer Table 1b, where we show that SLALOM can predict human attention for large models, including the non-transformer Mamba model (Gu & Dao, 2023).

F.6 Runtime analysis

We ran SLALOM as well as other feature attribution methods using surrogate models and compared their runtime to explain a single classification of a 6-layer BERT model. We note that the runtime is mainly determined by the number of forward passes to obtain the samples to fit the surrogates. While this number is independent of the dataset size, longer sequences require more samples for the same approximation quality. The results are shown in Table 13.

While IG and Gradient explanations are the quickest, they also require backward passes which have large memory requirements. As expected, the computational complexity for surrogate model explanation (LIME, SHAP, SLALOM) is dominated by the number of samples and forward passes done. **Our implementation of SLALOM is around 2x faster than LIME and almost 5x faster than SHAP** (all approaches used a GPU-based, batching-enabled implementation), which we attribute to the fact that SLALOM can be fitted using substantially shorter sequences than are used by LIME and SHAP.

We are interested to find out how many samples are required to obtain an explanation of comparable quality to SHAP. We successively increase the number of samples used to fit our surrogates and report the performance in the deletion benchmark (where the prediction should drop quickly when removing the most important tokens). We report the Area over the Perturbation Curve (AOPC) as before (this corresponds to their

Number of samples	Deletion AOPC
SHAP (nsamples="auto")	0.9135 ± 0.0105
SLALOM, 500 samples	0.9243 ± 0.0105
SLALOM, 1000 samples	0.9236 ± 0.005
SLALOM 2000 samples	0.9348 ± 0.005
SLALOM, 5000 samples	0.9387 ± 0.005
SLALOM, 10000 samples	0.9387 ± 0.005

Table 14: Ablation study on the number of samples required to obtain good explanations. The results highlight that a number as low as 500 samples can be sufficient to fit the surrogate model at a quality comparable to SHAP.

Comprehensiveness metric of ERASER (DeYoung et al., 2020), higher scores are better). We compare the performance to `shap.KernelExplainer.shap_values(nsamples=auto)` method of the shap package in Table 14. Our results indicate that sampling sizes as low as 500 per explained instance (which is as low as predicted by our theory, with average sequence length of 200) already yields competitive results.

F.7 Applying SLALOM to Large Language Models

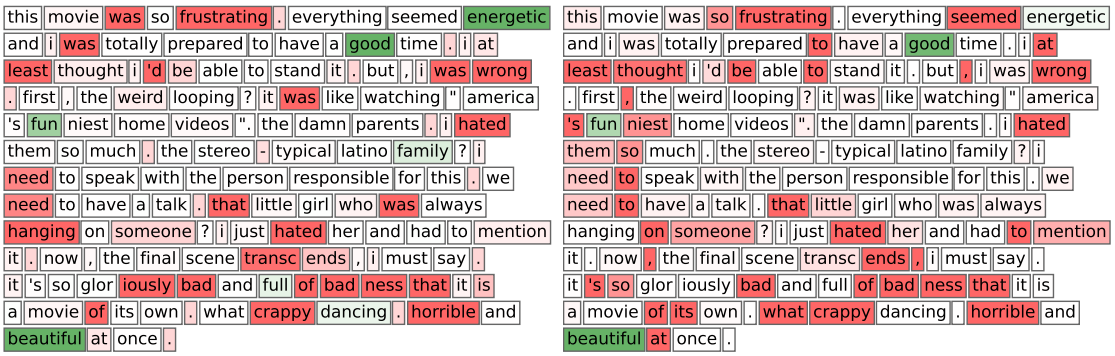
Our work is mainly concerned with sequence classification. In this application, we observe mid-sized models like BERT to be prevalent. On the huggingface hub, among the 10 most downloaded models on huggingface, 9 are BERT-based and the remaining one is another transformer with around 33M parameters[3] (as of September 2023). In common benchmarks like DBPedia classification[4], the top-three models are transformers with two of them also being variants of BERT. We chose our experimental setup to reflect this. Nevertheless, we are interested to see if SLALOM can provide useful insights for larger models as well and therefore experiment with larger models. To this end, we use a model from the BLOOM family (Le Scao et al., 2023) with 7.1B parameters as well as the recent Mamba model (2.8B) (Gu & Dao, 2023) on the Yelp-HAT dataset and compute SLALOM explanations. Note that the Mamba model does not follow the transformer framework considered in this work. We otherwise follow the setup described in Figure 3 and assess whether our explanations can predict human attention. The results on the bottom of Table 1b highlight that this is indeed the case, even for larger models. The ROC scores are in a range comparable to the ones obtained for the smaller models. For the non-transformer Mamba model we observe a drop in the value of the importance scores. This may suggest that value scores and linearized SLALOM scores are more reliable for large, non-transformer models.

Applying SLALOM to blackbox models. Finally, we would like to emphasize that SLALOM, as a surrogate model can be applied to black-box models as well. To impressively showcase this, we apply SLALOM to OpenAI's GPT-4 models via the API (Appendix F.7). We use the larger GPT-4-turbo and smaller GPT-4o-mini for comparison. We prompt the model with the following template to classify the review and only output either 0 or 1 as response.

SYSTEM: You are assessing movie reviews in an online forum. Your goal is to assess the reviews' overall sentiment as 'overall negative' (label '0') or 'overall positive' (label '1'). Your will see a review now and you will output a label. Make sure to only answer with either '0' or '1'.
USER: <the review>

We then use the token probabilities returned by the API to compute the score. We create 500 samples as a training dataset for the surrogate model and fit SLALOM the model using SLALOM-eff. We obtain the importance plots shown in Figure 13. This highlight that SLALOM scales up to large models. However, we would like to stress that there can be no guarantees as we have no knowledge about the specific structure of the model.

[3] `https://huggingface.co/models?pipeline_tag=text-classification&sort=downloads`
[4] `https://paperswithcode.com/sota/text-classification-on-dbpedia`



(a) GPT-4-turbo (b) GPT-4o-mini

Figure 13: We apply SLALOM to OpenAI GPT-4-turbo and GPT-4o to showcase its scalability. We use SLALOM-`eff` scores and linearize them to obtain the highlighed attributions.

F.8 Case study: Identifying Vulnerabilites and Weaknesses in classification models with SLALOM

In our public code repository, we provide a case study to highlight how SLALOM can be used to uncover practical flaws in transformer classification models. We use a trained BERT model on IMDB sentiment classification as an example and compute SLALOM scores for the tokens in the first 15 samples of the test set ($\approx$ 1200 tokens). SLALOM's scores make it easy to see which tokens have a potentially severe impact on the movie review outcome. We first select all tokens with an importance score > 2, all other words have no substantial impact, in particular when added to a larger sequence. We provide a visualization of these tokens in Figure 14. We then make some interesting discoveries mainly based on value scores:

1. There are more words that are negatively interpreted by the model than positive words
2. Out of the words that have highly negative value scores (+importance >2), we identify several words that are some that are not directly negatively connotated, e.g., “anyway”, “somehow”, “never”, “anymore”, “probably”, “doesn”, “maybe”, “without”, “however”, “surprised”

We then show that by a few (4) minor modification steps, e.g., adding some of these words to a review, we can change the classification decision for a review from positive to negative, without essentially altering its content (i.e., we manually construct an adversarial example.) This highlights how SLALOM can intuitively help to uncover 1) the influential tokens that contribute most to the decision and 2) allow for a fine-grained analysis that can help uncover spurious concepts and give practitioners an intuitive understanding of a model's weaknesses.

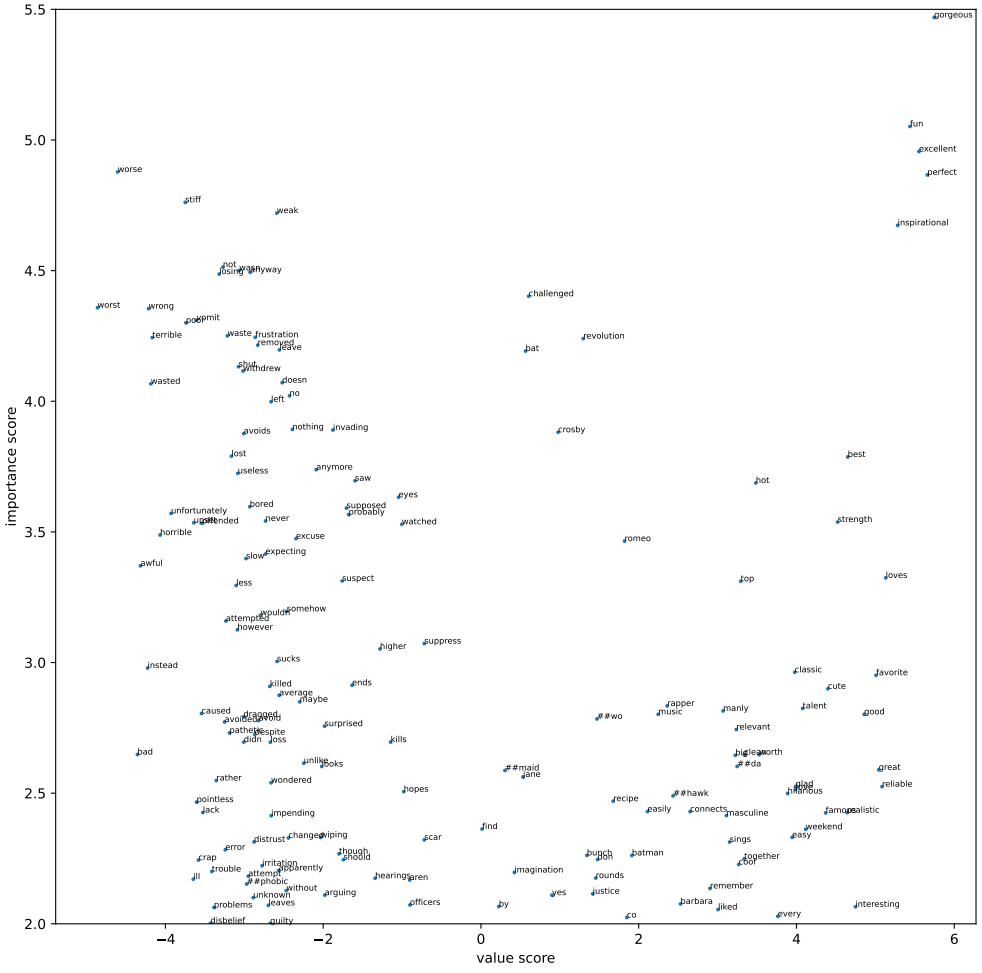


Figure 14: High-impact tokens for BERT identified in our case study. Use zoom for best view.

3.4 Protecting User Consent in Models with Optional Information

In the second part of this thesis, we consider privacy in machine learning. Privacy can be implemented at several stages in the ML pipeline: We first consider privacy at the interaction between the user and the model owner by providing the user with a choice to share data or not. Second, we consider how model owners can effectively preserve privacy during training. We show that implementing privacy at the first stage may entail a *discrimination risk*, while implementing it at the second stage may entail an *inaccuracy risk* for users.

Publication 4

Tobias Leemann, Martin Pawelczyk, Christian Eberle, and Gjergji Kasneci: I Prefer not to Say: Protecting User Consent in Models with Optional Personal Data. *AAAI Conference on Artificial Intelligence*, 2024.

Author Contributions. Gjergji Kasneci suggested the problem considered in this work in a joint discussion between us. I formalized the problem and developed the initial solution and theoretical results. Martin Pawelczyk then joined the project and helped to further refine the formulation and design the experimental evaluation. Christian Eberle implemented the initial codebase and ran the main experiments to show the bias that is introduced through the non-consent. I implemented the remaining experiments. Gjergji Kasneci contributed to the final narrative and supported us during the entire research process by providing insights from industry practice.

Note. This publication is included together with the supplementary materials in this thesis for completeness, but only the main paper is published with the proceedings.

Summary. In this work, we consider a seemingly simple strategy for privacy preservation: We allow users to voluntarily decide whether to share or not to share some features with a decision-making system. However, the sharing decision may be correlated with the label, e.g., in the example of health insurance, fitter patients are more likely to share data about workouts. This raises discrimination concerns, as users may be explicitly penalized for trying to protect their privacy. Empirically, we find that these non-consenting users may receive significantly lower prediction outcomes than justified by their provided information alone (i.e., when not using information about the non-consent). This observation gives rise to the overlooked problem of ensuring that users who protect

their personal data are not penalized. We offer the first solution to this problem by proposing the notion of Protected User Consent (PUC), which we prove to be a loss-optimal trade-off between protection requirements and performance. Briefly, PUC demands that a user who does not consent to sharing optional information should be treated as the average user with all available characteristics being equal, independently of the sharing decision. When the data is shared, the model owner may use the information passed to either increase, but also decrease the score. We propose PUC-IDA, a model-agnostic data augmentation strategy to implement PUC for any ML model. Our findings highlight the effectiveness of PUC at protecting non-sharing users from penalties. They also confirm that the performance loss when introducing PUC is almost negligible.

3.4.1 Discussion

The principle of Data Minimization is anchored in the GDPR (Art. 5). Data Minimization limits data collection to the data that is strictly necessary for the purpose of processing. Instead of leaving the choice of which data is necessary to the model owner, we hand this decision back to the user by letting them decide which information they want to share. This entails a discrimination risk that cannot be mitigated through classical fairness notions (Verma and Rubin, 2018). Our work shows that it is possible to prevent discrimination of non-consenting users while leveraging the optional features to make more accurate decisions for the consenting users. Notably, our work also explicitly models users that have an incentive to obtain a high score (i.e., in credit scoring) and strategically decide to share data. We find that in scenarios where the model owner is not allowed to introduce a penalty for non-sharers (e.g., to comply with legislation that forbids such a penalization), this has little effect and PUC stays the loss-optimal solution. With insurance and credit scoring companies more heavily relying on optional data sharing,[8] we believe it is crucial to have such mechanisms in place.

[8] see https://www.sueddeutsche.de/wirtschaft/schufa-superscore-kontoauszug-konto-horror-1.5128963. This artical details an attempt by German credit scoring provider SCHUFA to offer customers to share bank account statements to improve their score (in German, accessed 5 January 2025)

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

I Prefer Not To Say: Protecting User Consent in Models with Optional Personal Data

Tobias Leemann[1,2], **Martin Pawelczyk**[3], **Christian Thomas Eberle**[1], **Gjergji Kasneci**[2]

[1]University of Tübingen, Tübingen, Germany
[2]Technical University of Munich, Munich, Germany
[3]Harvard University, Cambridge, MA, USA
tobias.leemann@uni-tuebingen.de, martin.pawelczyk.1@gmail.com, ct.eberle@protonmail.ch, gjergji.kasneci@tum.de

Abstract

We examine machine learning models in a setup where individuals have the choice to share optional personal information with a decision-making system, as seen in modern insurance pricing models. Some users consent to their data being used whereas others object and keep their data undisclosed. In this work, we show that the decision not to share data can be considered as information in itself that should be protected to respect users' privacy. This observation raises the overlooked problem of how to ensure that users who protect their personal data do not suffer any disadvantages as a result. To address this problem, we formalize protection requirements for models which only use the information for which active user consent was obtained. This excludes implicit information contained in the decision to share data or not. We offer the first solution to this problem by proposing the notion of Protected User Consent (PUC), which we prove to be loss-optimal under our protection requirement. We observe that privacy and performance are not fundamentally at odds with each other and that it is possible for a decision maker to benefit from additional data while respecting users' consent. To learn PUC-compliant models, we devise a model-agnostic data augmentation strategy with finite sample convergence guarantees. Finally, we analyze the implications of PUC on challenging real datasets, tasks, and models.

Introduction

While the day-to-day impact of automated data processing is steadily growing, modern regulations such as the European Union's General Data Protection Regulation (GDPR) (GDPR 2016) or the California Consumer Privacy Act (CCPA) (OAG 2021) strive to give individuals more control over their personal data. In light of these regulations, we consider machine-learned classifiers in which individuals have the freedom to decide themselves on which data they would like to provide to an automated decision system.

Such systems are increasingly being deployed (Henning 2022): As a running example, we consider a realistic use-case of health insurance pricing: Suppose in an automated pricing model all potential customers are asked to fill out an application form where they enter certain *base features*, for instance information such as their state of residence and age.

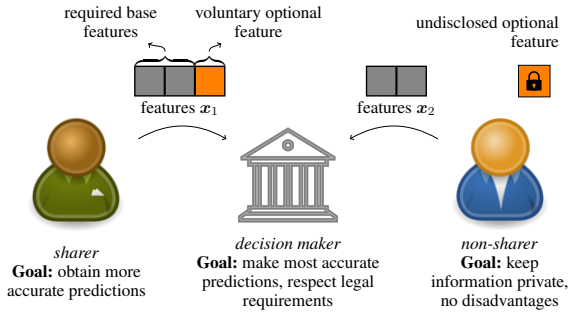


Figure 1: Overview of the relevant stakeholders. We consider a case where users can voluntarily provide information on optional features or choose to leave them undisclosed. The goals of sharers, non-sharers, and the decision maker have to be reconciled.

To improve the pricing model, the insurance offers an additional service, a "companion fitness app", through which additional health data about the customer's physical condition are collected. The customers decide whether to use the app or not; alternatively, customers can sign up for a policy without consenting to use the app. The health data that customers share may however influence the premium of the insurance policy they receive. We refer to data that provide additional, non-mandatory information beyond the base features as *optional features*. With fitness trackers and smartwatches rapidly gaining popularity (Reeder and David 2016; Zimmer et al. 2020; Statista 2023), such systems are already being deployed in practice, e.g., by major health insurance firms in Australia (Henning 2022).

The outlined scenario is challenging as there are three groups of stakeholders whose interests need to be reconciled: (1) The group of non-sharing individuals who do not want to provide additional information, for instance due to privacy concerns. We refer to them as *non-sharers*. For this group, the decision maker does not want to or cannot force them to provide the additional information for legal reasons. Consequently, the non-sharers do not want the additional information to be considered in the decision making process; in return, they are willing to sacrifice some accuracy, but they do not want to face other systematic disadvantages. (2)

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

On the other hand, individuals who voluntarily share data (*sharers*) explicitly want the additional information to be considered and want to obtain more accurate predictions. (3) Finally, the decision makers themselves desire the most accurate predictions with the lowest overall costs while respecting the users' privacy and legal requirements.

Among these requirements, it is crucial to the non-sharers to explicitly exclude the information contained in the decision to share or not to share. To see this, we note that smartwatch users are more likely to exercise in general than non-wearers (DeMarco 2023) which usually create lower costs for the insurance company as fitter customers take less sick days on average. Thus, only through observing the decision to share data, the insurance firm could make inferences about a person's fitness. This is problematic for two reasons: First, the company would unethically infer private data, that the non-sharers explicitly did not give consent to. Prior work (Wachter and Mittelstadt 2019) has argued for a "right to reasonable inferences". This rules out inferences from unrelated factors that are purely predictive and may infringe privacy, as they open the door for discriminatory and invasive decision-making (Mittelstadt et al. 2016). Second, this would lead to non-sharers being assigned a higher insurance premium than the estimate of the legacy model which only considered their base features. Many countries have laws that prohibit insurers from raising the base premium for users who do not share their data, as this is seen as a coercive and unfair practice. For example, the US only permits five factors to affect the premium, which are location, age, tobacco use, plan category, and dependent coverage (US Government. U.S. Centers for Medicare & Medicaid Services. 2023). It is however possible for insurers – and desired by many users – to award bonuses which reduce the premium based on participation in optional reward and incentive programs (Madison, Schmidt, and Volpp 2013; Henning 2022).

To summarize, we study machine learning models that can handle optional features and meet legal requirements and desiderata of three groups of stakeholders: the sharers, the non-sharers, and the decision makers. We consider it essential for these models to not make inferences based on the unavailability of a feature value for the non-sharers, a constraint that we term *Availability Inference Restriction (AIR)*. Finally, we are interested in obtaining models with optimal performance under this requirement.

Contribution. We address the problem of how to fairly and privately predict outcomes for users who share optional data and those who do not. We tackle this overlooked issue by making the following contributions:

- **Definition.** We introduce models with Protected User Consent (PUC), which are optimal under our protection requirement AIR. We derive performance guarantees, which formally show that it is possible to reconcile the decision maker's interest in improved predictions and the non-sharer's privacy preferences.
- **Algorithm.** We propose a PUC-inducing data augmentation (`PUCIDA`) technique that can be applied to any type of predictive architecture (e.g., tree or neural network) and any convex loss function (e.g., mean squared error or cross-entropy loss) to obtain such models
- **Analysis.** We prove that predictive models trained with `PUCIDA` satisfy PUC asymptotically, and provide finite sample convergence results that demonstrate that `PUCIDA` produces PUC-compliant models in practice.
- **Empirical evaluation.** We empirically show that without enforcing PUC, the average absolute prediction outcome (e.g., insurance quote) of users who do not share data can be almost 20 % worse than justified by their base data. We then evaluate our data augmentation technique on various ML models and show that PUC is achieved regardless of the model.

Related Work

In this Section, we review the most relevant streams of related work (see Appendix A.1 for additional references).

Classification with Missing Values. Classification models that can handle missing data have been studied previously with the goal of minimizing costs or increasing performance (Zhang et al. 2005; Aleryani, Wang, and De La Iglesia 2020), obtaining uncertainty estimates (Kachuee et al. 2020), or fulfilling classical fairness notions (Zhang and Long 2021; Jeong, Wang, and Calmon 2022; Wang and Singh 2021; Fernando et al. 2021). However, the mechanisms underlying missingness is different in this work, as missing values indicate explicit non-consent by the user, leading to different implications. In a related line of work, classification with noisy (Fogliato, Chouldechova, and G'Sell 2020) or missing labels (Kilbertus et al. 2020; Rateike et al. 2022) has been investigated, where the missingness is often a result of *selection bias*. The setting considered in this work is different in the sense that we are not concerned with fulfilling a fairness notion with respect to a sensitive attribute, but consider the interests of subjects that have and have not provided optional information.

Data Minimization. The principle of Data Minimization is anchored in the GDPR (GDPR 2016). Data Minimization demands minimal data collection. Several works are concerned with implementing (Goldsteen et al. 2021) or auditing compliance with this principle (Rastegarpanah, Gummadi, and Crovella 2021). Rastegarpanah et al. (Rastegarpanah, Crovella, and Gummadi 2020) consider decision systems that can handle optional features from a data minimization perspective where the decision maker decides which features are collected for each individual. This principle is distinct from the "right to be forgotten" (Biega et al. 2020), which enables individuals to submit requests to have their data deleted. In response to these regulations, several works consider the problem of updating an ML model without the need of retraining the entire model (Wu, Dobriban, and Davidson 2020; Ginart et al. 2019; Izzo et al. 2021; Golatkar, Achille, and Soatto 2020) or the effect of removals on model explanations (Rong et al. 2022; Pawelczyk et al. 2023). Our work differs from these works as our goal is to train a model where users decide themselves which data they deem relevant through sharing one or many optional features.

Algorithmic Fairness. A multitude of formal fairness definitions have been put forward in the literature (Verma and Rubin 2018). Examples include statistical parity (Dwork et al. 2012), predictive parity (Chouldechova 2017), equalized odds, equality of opportunity (Hardt, Price, and Srebro 2016), and individual fairness (Dwork et al. 2012). However, they are still a topic of discussion, for instance, because these definitions are known to be incompatible (Kleinberg, Mullainathan, and Raghavan 2016; Lipton, McAuley, and Chouldechova 2018). Additionally, there are a several definitions that rely on causal mechanisms to assess fairness, e.g., counterfactual fairness (Kusner et al. 2017), and the notion of unresolved discrimination (Kilbertus et al. 2017). While causal approaches to fairness might be preferable, they require information about the causal structure of the data generating process. Moreover, it has recently been shown that causal definitions may lead to adverse consequences, such as lower diversity (Nilforoshan et al. 2022). We discuss how existing fairness definitions could possibly be applied to the setting with optional features, but we find that none of the fairness definitions aligns with our desiderata theoretically and experimentally (see Appendix A.2).

Strategic Classification. In an even broader context, this work also relates to the field of strategic classification (Hardt et al. 2016). However, it is worth noting that in strategic classification research, the focus primarily revolves around users strategically manipulating their features for optimal outcomes, which may also involve information withholding (Krishnaswamy et al. 2021). In contrast to our work, privacy concerns are neglected in this research stream. As far as we are aware, there are no prior works on the specific problem of balancing the interests of *all three* groups of stakeholders (the non-sharers, sharers, and the decision makers).

Problem Formulation

Formalization and Notation

In this work, each data instance contains a realization of a number of base features $\mathbf{b} \in \mathcal{X}^b$, where $\mathcal{X}^b \subseteq \mathbb{R}^n$ is the space of the base features. Furthermore, let there be some optional information $z \in \mathcal{X}^z$, where $\mathcal{X}^z \subseteq \mathbb{R}$ is the value space of the optional feature.[1] It is the users' choice to decide if they want to disclose z to the system, which results in an availability variable $a \in \{0, 1\}$. Accordingly, only imputed samples $z^* = \{z \text{ if } a{=}1, \text{ else } \texttt{N/A}\}$ are observed, where a value of `N/A` indicates that a user did not reveal the optional information, e.g., did not use the companion app. In summary, the data observations are tuples $\mathbf{x} = (\mathbf{b}, a, z^*)$ that reside in $\mathcal{X} = \mathcal{X}^b \times \{0, 1\} \times (\mathcal{X}^z \cup \{\texttt{N/A}\})$. Each training sample comes with a label $y \in \mathcal{Y}$. Further, there is a data generating distribution $\mathbf{p}$ with support $\mathcal{X} \times \mathcal{Y}$ and we have access to an i.i.d. training sample $(\mathbf{x}, y) \sim \mathbf{p}$. Figure 2 shows such a data sample. We denote the random variables for the respective quantities by $\boldsymbol{B}, A, Z, Z^*, Y$. The label is probabilistically determined through the base features $\mathbf{B}$ and the hidden feature Z but the sharing decision does not influence the true label for a given $\mathbf{B}, Z$, such that $Y \perp\!\!\!\perp A | \mathbf{B}, Z$.

[1] We extend our definitions to integrate multiple optional features a later section.

base features $\mathbf{b}$		opt. feat. z^*	a	label y
state	plan	fitness score	avail.	treatment costs
New South Wales	basic	87 %	1	3k$
Queensland	gold	`N/A`	0	17k$
New South Wales	basic	92 %	1	5k$
New South Wales	basic	`N/A`	0	64k$
Victoria	premium	56 %	1	22k$

Figure 2: Samples for the insurance use-case. We have two base features $\mathbf{b}$ and one optional feature z^*, which either takes an observed value z, or it takes a value of `N/A` if unobserved. The variable $a \in \{0, 1\}$ indicates the availability of the feature. The goal is to predict the label y.

In many applications, the goal is to find a function $f : \mathcal{X} \to \mathcal{Y}$ that models the observed data. In particular, $f : \mathcal{X} \to [0, 1]$ may predict a probability of a positive outcome or $f : \mathcal{X} \to \mathbb{R}$ may return a numerical score. The test data for which the model will be used come from the same distribution $\mathbf{p}$, though with the label y unobserved, and we suppose that the information provided is always correct. We consider a convex loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, e.g., mean-squared-error (MSE) or binary cross entropy (BCE), for which we minimize the expected loss for a sample from the data distribution. For instance, using the common MSE loss $\mathcal{L}(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$, an optimal predictor is given by $f^*_{\mathcal{L}}(\mathbf{x}) = \arg\min_{f(\mathbf{x})} \mathbb{E}_{\mathbf{p}(Y|\mathbf{x})} \left[(f(\mathbf{x}) - Y)^2\right] = \mathbb{E}[Y|\mathbf{x}]$, the conditional expectation. However, this notion can be generalized to other loss functions: An optimal predictor $f^*_{\mathcal{L}}(\mathbf{x})$ for the loss function $\mathcal{L}$ fulfills $\forall \mathbf{x}$:

$$f^*_{\mathcal{L}}(\mathbf{x}) = \mathbb{F}^{\mathcal{L}}_{\mathbf{p}}[Y|\mathbf{x}] \coloneqq \underset{f(\mathbf{x})}{\arg\min}\, \mathbb{E}_{\mathbf{p}(Y|\mathbf{x})} \left[\mathcal{L}(f(\mathbf{x}), Y)\right]. \quad (1)$$

We use $\mathbb{F}^{\mathcal{L}}[Y|\mathbf{x}]$ to denote a generalized expected value that minimizes the expected loss conditioned on $\mathbf{x}$. To ease our derivations, we suppose this minimum to be unique and finite. Intuitively, it represents the best guess of Y given $\mathbf{x}$. For the MSE-Loss, $\mathbb{F}^{\mathcal{L}}$ is equivalent to the expectation operator $\mathbb{E}$. In the following statements, the reader may thus mentally replace $\mathbb{F}^{\mathcal{L}}$ with an expectation $\mathbb{E}$ without further ramifications in order to get the high level intuition. Finally, we introduce two key terms, namely, *base feature model* and *full feature model*. The former refers to a model trained on the base features only, while the latter refers to a model trained on all features where some strategy is used to replace unavailable feature values. Typically these strategies are called *imputation* and replace unavailable values by zeros, a feature's mean or median (Emmanuel et al. 2021).

Desiderata

Our goal is to learn models $f : \mathcal{X} \to \mathcal{Y}$ that comply with the desideratum of *Availability Inference Restriction*, which we briefly introduced in Section , to protect the interests of the non-sharers. Under this constraint, the model should provide the best predictive performance to reflect the need of

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

the sharers and the decision maker for most accurate predictions.

Desideratum 1: Availability Inference Restriction. We start by considering the intricate case of individuals who *do not want to share optional information*. In this case, the model should compute the prediction based on the information the user gave their consent to. In particular, (a) the model should only use the base features *and* (b) should not use information that could be derived from the unavailability of the optional features to compute the prediction to avoid violating the user's consent.

For (a), this requires that the predictor does not use the information as an explicit input, i.e., the predictor should behave as if it only used base features $\mathbf{b}$ via some function $g : \mathcal{X}^b \to \mathcal{Y} : f_{|_{a=0}}(\mathbf{b}, a, z^*) = g(\mathbf{b})$. For (b), although a=0 is not an explicit input to g, a sufficiently complex function may still be implicitly adapting to the group $a = 0$ and thus incorporate information that the user did not give their consent to. We would like to make sure that the predictions of g cannot use more information than contained in the overall conditional distribution, given the base features $\mathbf{b}$. This overadaption can be prevented by constraining the model's loss on the population of non-sharers to match the loss of the optimal base model $f^*_{\mathcal{L}}$ on this population. The reasoning behind this rationale is that all models that would beat the performance of this model must implicitly use some additional side knowledge about this group that was not provided by the users.

Definition 1 (Availability Inference Restriction)**.** *For individuals that choose not to provide the optional feature (a=0), only the provided data* $\mathbf{b}$ *is used to compute the outcome in the decision process, i.e., $f_{|_{a=0}}(\mathbf{b}, a, z^*)$=$g(\mathbf{b})$, where $g : \mathcal{X}^b \to \mathcal{Y}$ is a base feature model. Further, we require*

$$\mathbb{E}\left[\mathcal{L}\left(g(\mathbf{B}), Y\right)|A = 0\right] \geq \mathbb{E}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 0\right]. \quad (2)$$

This definition summarizes our intuition that the information encoded through the unavailability of feature information should neither be used explicitly (a) nor implicitly (b). We show how this constraint can analogously be derived from information-theoretic considerations in Appendix B.3.

Desideratum 2: Optimality. Our Definition 1 restricts the information that the predictor can use when the optional information is unavailable. To meet the interests of the decision maker and the sharers, we also want to find models with optimal performance, i.e., lowest loss, under this constraint.

Protecting User Consent

We are therefore looking for an *optimal* model within the class of predictors that comply with Availability Inference Restriction. In this Section, we derive a novel notion called Protected User Consent (PUC) that fulfills this purpose.

One-Dimensional PUC

The next result encodes an intuitive notion of protection for the users that do not want to share data on the optional features (a=0): Their prediction under f is then constrained to the best estimate for a user with the same base characteristics, no matter if additional data was provided. Contrarily, when additional information through the optional feature is provided, the predictor returns the best estimate using the available optional information:

Theorem 1 (1D-PUC)**.** *Let $f : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}$ be a full feature model (i.e., including optional features). Among all predictors compatible with the Availability Inference Restriction, a model f with minimal loss is given by:*

$$f^*_{PUC}(\mathbf{b}, a, z^*) = \begin{cases} \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}], & \text{if } a = 0 \\ \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, A = 1, Z^* = z^*] & \text{if } a = 1. \end{cases}$$

We defer all proofs in this work to Appendix D. PUC is different from existing notions of group fairness, that do not fulfill the two desiderata in general (see Appendix A.2 for a discussion). Under the mentioned requirements, there is no model that can outperform f^*_{PUC}. We stress that 1D-PUC-compliant models have performance guarantees. These models match or improve upon an optimal base feature model $f^*_{\mathcal{L}}(\mathbf{B}) = \mathbb{F}_{\mathcal{L}}[Y|\mathbf{b}]$. This model can be seen as an upper bound for practical models obtained after model selection. Therefore, models that can beat its performance may offer improvements even after extensive hyper-parameter tuning and model selection, a property which we refer to as Predictive Non-Degradation (PND): a model f fulfills PND if its loss is smaller than that of the base feature model:

$$\mathbb{E}[\mathcal{L}\left(Y, f^*_{\mathcal{L}}(\mathbf{B})\right)] \geq \mathbb{E}[\mathcal{L}\left(Y, f(\mathbf{B}, A, Z^*)\right)]. \quad (3)$$

We prove the following result:

Corollary 1 (Predictive Non-Degradation of f^*_{PUC})**.** *For any density* $\mathbf{p}$*, a PUC-compliant model f^*_{PUC} fulfills Predictive Non-Degradation, i.e., it has a loss upper-bounded by the optimal base feature model $f^*_{\mathcal{L}}$.*

This is a remarkable result as it testifies that the decision maker can benefit from additional information in terms of loss, while protecting the privacy of users. This highlights that the interests of the different stakeholders are not contradictory and models that benefit all stakeholders do exist.

PUC under Strategic Considerations and Monotonicity Constraints

We have initially considered the case where the users desire the highest possible accuracy under data usage restrictions. However, in some cases such as our initial insurance example, the motivation to receive a lower premium might be a more important concern to some users than receiving an accurate prediction or their privacy concerns. If all users have full information (i.e., they see premiums with and without their optional data) and act strategically by sharing the value of z only if it would decrease their premiums, we obtain the following result.

Theorem 2 (Optimality of f^*_{PUC} under strategic actions)**.** *Let $\mathbf{p}'(\mathbf{B}, Z, Y)$ be any prior density on base features, true optional features and labels and let $f(\mathbf{b}, a = 0, z) = \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}]$, i.e., the decision maker uses the base feature model when no optional data is available. Further suppose that users*

strategically choose to share the optional feature z only if $f(\mathbf{b}, a = 1, z) \leq f(\mathbf{b}, a = 0, \texttt{N/A})$. Under these conditions, the model f^*_{PUC} (Theorem 1) has minimal loss among all predictors.

This result underlines that PUC models remain optimal if the decision maker cannot increase the premiums beyond the predictions of the current base model for the non-sharers. This is reasonable in many cases, where legal constraints mandate that the decision maker cannot implicitly force users to share data by inflating the base premium, as outlined in the introduction. The sharing decision can also be automated for the users by simply dropping the optional feature if it does not lead to a decrease in premiums. This would result in the aforementioned bonus systems, where sharing more data cannot increase the premium. We show that among the class of models with such a monotonicity constraint, the outlined PUC-model with automatic sharing decisions is still optimal under the same conditions as in Theorem 2 in Appendix D.5.

r-dimensional PUC

Next, we generalize our notion such that r features can be provided optionally. For example, the insurance firm might also accept voluntary results from prior medical examinations or diagnostic tests. Therefore, let there now be r optional features such that $\boldsymbol{z} \in \mathcal{X}_1^z \times \cdots \times \mathcal{X}_r^z$ and $\boldsymbol{a} \in \{0,1\}^r$, where $\mathcal{X}_i^z$ are the respective supports of each optional feature. By $\mathcal{I} \subseteq [r] = \{1, \ldots, r\}$, we denote an index set that contains all feature indices present, i.e., $\mathcal{I}(\boldsymbol{a}) = \{i \mid \boldsymbol{a}_i = 1, i = 1, \ldots, r\}$. When we index vectors with this set, e.g., $\boldsymbol{Z}_\mathcal{I}$, we refer to the subvector that only contains the indices in $\mathcal{I}$.

Definition 2 (Protected User Consent, PUC). *Let $f : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}$ be a full feature model. The model f^*_{PUC} that fulfills Protected User Consent is given by*

$$f^*_{PUC}(\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*) = \mathbb{F}^{\mathcal{L}}_{(\mathbf{B}, \boldsymbol{A}, \boldsymbol{Z}^*) \sim \mathbf{p}} \left[Y \middle| \mathbf{B} = \mathbf{b}, \boldsymbol{A}_{\mathcal{I}(\boldsymbol{a})} = \mathbf{1}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})} = \boldsymbol{z}^*_{\mathcal{I}(\boldsymbol{a})} \right],$$

where $\boldsymbol{A}_{\mathcal{I}(\boldsymbol{a})} = \mathbf{1}$ means that each element that is set to 1 in $\boldsymbol{a}$ needs to be one in $\boldsymbol{A}$ as well.

For a single feature (r=1), the index set can either be $\mathcal{I}$=$\emptyset$ or $\mathcal{I}$=$\{1\}$ and the definition corresponds to 1D-PUC. The conditional expectation with $\boldsymbol{A}_{\mathcal{I}(\boldsymbol{a})}$=$\mathbf{1}$ effectively constrains the features in $\mathcal{I}$ to be available, but marginalizes over samples with or without further information.

Implementing Protected User Consent

In this section, we derive a model-agnostic approach called *PUC-inducing data augmentation* (PUCIDA) to achieve protected user consent. By using theoretical analysis, we establish that PUCIDA will result in exact protected user consent. Furthermore, we establish performance guarantees that provide an upper bound on the deviation between practical, finite sample-based PUC-compliant models and their theoretical infinite sample limits.

	state	plan	score	costs
	NSW	basic	87 %	3k$
⊕	**NSW**	**basic**	**N/A**	**3k$**
	NSW	basic	92 %	5k$
⊕	**NSW**	**basic**	**N/A**	**5k$**
	NSW	basic	N/A	64k$

Figure 3: Explaining PUCIDA. Our data augmentation procedure expands each instance with optional information into two samples: The original instance and a synthetic sample (⊕). The synthetic samples retain the base features and the labels, but the information on the optional features is dropped (fitness score → N/A). The model sees samples with the same base features with a missing value and will thus base its decision only on the base features. In this example, given the base features ("NSW", basic) and no optional statements, the model would estimate the costs to be 24k$, which is the dataset average conditioned on these values.

PUCIDA: PUC-inducing Data Augmentation

Intuitively, we want to prevent the model from making inference from a feature's missingness patterns. The core insight is to leverage synthetic samples that make the *distribution of the labels given missingness equal to the overall label distribution*. Thereby, we prevent the derivation of predictive information from the missingness itself (see Table 3).

For a single optional feature, extensively enumerating all samples as in the table is possible while for multiple features this may be intractable. Therefore, we do not list all samples but propose a stochastic, multifeature variant of the algorithm: **(1)** Instead of drawing samples with uniform probability from the distribution $\mathbf{p}$, we use non-normalized weights w:

$$w(\mathbf{x}) = w(\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*) = 2^{|\mathcal{I}(\boldsymbol{a})|}. \tag{4}$$

This step corresponds to the expansion of an instance into $2^{|\mathcal{I}(\boldsymbol{a})|}$ synthetic ones; e.g., a sample with a single optional feature is assigned a weight of two (cf. Figure 3). Training instances are drawn with a probability proportional to these weights. This results in data instances with optional information being more frequently sampled. **(2)** We require a sample modification where optional features are randomly dropped from the samples. For each sampled item, we drop each available optional feature with probability p=0.5:

$$\mathbf{q}_i \sim \text{Bern}(0.5), i = 1, \ldots, r; \quad \overline{\boldsymbol{a}} = \mathbf{q} \odot \boldsymbol{a}; \tag{5}$$

$$\overline{\boldsymbol{z}}^*_i = \{\boldsymbol{z}^*_i \text{ if } \overline{\boldsymbol{a}}_i{=}1, \text{ else } \texttt{N/A}\}, i = 1, \ldots, r. \tag{6}$$

(3) We train the predictive model on the modified samples $(\overline{\mathbf{x}}, y) = ((\mathbf{b}, \overline{\boldsymbol{a}}, \overline{\boldsymbol{z}}^*), y) \sim \overline{\mathbf{p}}$ derived through this procedure.

Theoretical Analysis

We summarize PUCIDA in pseudo-code in Appendix D.8 and provide the following theorem to demonstrate that PUCIDA leads to PUC-compliant models.

Theorem 3. *The loss-minimal model $f(\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*) = \mathbb{F}^{\mathcal{L}}_{\overline{\mathbf{p}}}[Y | \mathbf{b}, \boldsymbol{A} = \boldsymbol{a}, \boldsymbol{Z}^* = \boldsymbol{z}^*]$ on the modified distribution $\overline{\mathbf{p}}$*

fulfills Protected User Consent with respect to $\mathbf{p}$*, i.e.,*

$$\mathbb{F}^{\mathcal{L}}_{\overline{\mathbf{p}}}[Y|\mathbf{B}=\mathbf{b}, \boldsymbol{A}=\boldsymbol{a}, \boldsymbol{Z}^*=\boldsymbol{z}^*] = \mathbb{F}^{\mathcal{L}}_{\mathbf{p}}\left[Y\middle|\mathbf{B}{=}\mathbf{b}, \boldsymbol{A}_{\mathcal{I}(\boldsymbol{a})}{=}\mathbf{1}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})}{=}\boldsymbol{z}^*_{\mathcal{I}(\boldsymbol{a})}\right] = f^*_{PUC}(\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*).$$

This result is remarkable in its generality as it enables PUC-compliant models using standard optimization procedures by modifying the distribution of the data; i.e., *PUCIDA can be combined with any existing model and training pipeline*. Next, 'we' study the theoretical convergence behavior for PUCIDA on finite samples. To this end, we define the PUC-Gap as the expected squared deviation from PUC:

$$\text{PUC-Gap}^2(f, \mathbf{p}) = \mathbb{E}_{(\mathbf{B},\boldsymbol{A},\boldsymbol{Z}^*)\sim\mathbf{p}}\left[\left(f(\mathbf{B}, \boldsymbol{A}, \boldsymbol{Z}^*) - f^*_{\text{PUC}}(\mathbf{B}, \boldsymbol{A}, \boldsymbol{Z}^*)\right)^2\right]. \tag{7}$$

We will restrict ourselves to $\mathcal{L} \equiv \text{MSE}$ and thus $\mathbb{F}^{\mathcal{L}} \equiv \mathbb{E}$, and study a *baseline conditional expectation estimator* $\hat{\mu}$ which averages the labels conditional on all observations with the same features $\mathbf{x}$. For brevity, we refer to Appendix D.7 (Eqn. 51) for a formal definition of this estimator. Since we usually cannot compute the exact expectation from Theorem 3, we are interested in the number of samples required from $\overline{\mathbf{p}}$ to obtain a fixed average estimation error for which we establish the following result.

Theorem 4 (Finite Sample Convergence)**.** *Let* $\mathcal{X} = \mathcal{X}^b \times (\mathcal{X}^z \cup \{\texttt{N/A}\})$ *be finite feature space and let* $\mathcal{Y} \subseteq \mathbb{R}$ *be the label space. All conditional expectations* $\mu(\mathbf{x}) := \mathbb{E}_{\overline{\mathbf{p}}}[y|\mathbf{x}]$ *and the conditional variances* $\sigma^2(\mathbf{x}) := \text{Var}_{\overline{\mathbf{p}}}[y|\mathbf{x}]$ *exist and are finite. Then there exists a baseline non-parametric regressor* $\hat{\mu} : \mathcal{X} \mapsto \mathbb{R}$ *from a finite number of* N *independent, identically distributed observations* $(\overline{\mathbf{x}}_i, y_i)_{i=1\dots N}$ *from* $\overline{\mathbf{p}}$ *with a convergence rate of* $\mathcal{O}(N^{-1})$*; more specifically*

$$\text{PUC-Gap}^2(\hat{\mu}, \mathbf{p}) = \mathbb{E}_{\mathbf{X}\sim\mathbf{p}}\left[\left(\hat{\mu}(\mathbf{X}) - \mu(\mathbf{X})\right)^2\right] \leq \frac{2^r|\mathcal{X}|^2(\sigma^2_{\max} + \mu^2_{\max})}{N} + \mathcal{O}\left(\frac{1}{N^2}\right),$$

with $\sigma^2_{\max} := \max_{\mathbf{x}\in\mathcal{X}} \sigma^2(\mathbf{x})$ *and* $\mu^2_{\max} := \max_{\mathbf{x}\in\mathcal{X}} \mu^2(\mathbf{x})$.

In conjunction with Theorem 3, this result provides a bound on the expected gap to perfect protected user consent that is dependent of the sample size, which decreases with a rate of $\mathcal{O}\left(N^{-1}\right)$. Several remarks are in place: We obtain a multiplicative constant which depends on the number of optional features r and the size of the feature space $|\mathcal{X}|$. The square of this quantity enters the result because the number of samples available to estimate each conditional mean is not independent, as they need to sum up to N. For large feature spaces, however, they are almost independent and we expect the constant to scale almost linearly in $|\mathcal{X}|$. The growth of 2^r is attributed to the re-sampling strategy which might assign a very low probability to certain inputs, which may only be well approximated with a high number of samples. As the number of optional features is typically limited in realistic use-cases it will be well outgrown by N. Note that more powerful model (e.g., Tree based model + PUCIDA) usually outperform this baseline.

data	base model	Full feature model	PUCIDA
diab.(C)	33.84% ±2.47	31.44% ±2.19	34.01% ±1.71
compas (C)	44.47% ±0.37	41.47% ±1.09	44.54% ±0.54
adult (C)	13.37% ±0.07	12.84% ±0.28	13.41% ±0.12
water (C*)	10.65% ±1.64	10.00% ±1.58	10.97% ±1.21
colic (C*)	13.81% ±0.82	11.34% ±0.46	15.05% ±0.68
income (R)	109.56 ±1.00	109.11 ±1.29	110.73 ±1.29
calif. (R)	15.79 ±0.10	15.16 ±0.28	16.18 ±0.06
insurance (R)	283.47 ±0.53	279.78 ±0.42	285.31 ±0.39

Table 1: Availability Inference Restriction is violated by full feature models (Random Forests). As expected, the full feature models always have lower losses than the base-models, indicating that Availability Inference Restriction is violated while PUCIDA fullfils Availability Inference Restriction. We report misclassification error rates for classification models and MSE loss ($\times$ 100) for regression models.

Practical considerations. For smaller datasets, an alternative approach to random sampling is to use all possible samples to approximate the distribution $\overline{\mathbf{p}}$ by a method we call "exhaustive augmentation". This involves enumerating all possible variations of the original samples, including any optional features, to form a larger dataset $\mathcal{D}'$. The model is then trained on this expanded dataset.

Experimental Evaluation

Here, we empirically validate the effectiveness of our methods using eight real-world datasets and one synthetic dataset. In particular, we highlight that (a) full feature models violate the Availability Inference Restriction and make it harder for non-sharers to obtain the positive outcome, (b) PUCIDA results in PUC-compliant models as suggested by our theory, and that (c) the reduction in terms of model performance due to using PUC are moderate relative to deploying a full feature model.

Common datasets. We use eight real-world datasets commonly found in the related literature. For classification (C), the Diabetes (diab) and the horse colic dataset (colic) study the prediction of diseases, the COMPAS dataset is concerned with estimating likelihood of recidivism and UCI Adult income dataset requires to predict whether individuals have an income of over 50k$. The water treatment dataset (water) predicts the operational state of a facility. We also study the regression tasks (R) of house price estimation in California (calif), income prediction (income), and inferring information from insurance claims (insurance) to link to our initial example. Details about preprocessing, dataset sources and model hyperparameters are provided in Appendix F.2.

Availability. The colic and the water dataset come with inherent missing values that we use (indicated through $*$). For six more datasets we introduce availability dependent on a feature's value. We compute the probability of feature unavailability $\mathbf{p}(A_i = 0|z_i)$ by applying a sigmoid function centered at the feature mean and sample the availability a from the respective conditional distribution. We additionally

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

				Full feature model		PUCIDA	
task	data	optional	Base feature model	pred.	change	pred.	change
C	diab.	Glucose	60.27%	45.19%	-15.08% ±2.01	61.20%	0.93% ±0.93
C	compas	#priors	51.19%	32.86%	-18.33% ±0.89	51.34%	0.15% ±0.59
C	adult	edu-num	13.86%	11.44%	-2.42% ±0.07	13.92%	0.06% ±0.05
C*	water	oxygen. dem.	87.10%	84.52%	-2.58% ±2.81	87.42%	0.32% ±1.58
C*	colic	abdom. app.	6.39%	1.24%	-5.15% ±0.92	7.01%	0.62% ±1.64
R	income	WKHP	100.0%	81.2%	-18.8% ±0.61	101.2%	1.2% ±0.19
R	calif.	m_income	100.0%	94.4%	-5.6% ±0.67	103.8%	3.8% ±0.42
R	insurance	experience	100.0%	94.8%	-5.2% ±0.09	100.1%	0.1% ±0.05

Table 2: Measuring the average predictions for non-sharers. For classification tasks we report the positive outcomes (in %), and for regression tasks, we report relative predictions to the base feature model (set to 100 %). The non-sharers face disadvantages for not providing the voluntary information and are assigned less favorable prediction outcomes by the full feature models. This discrepancy vanishes when PUCIDA is applied.

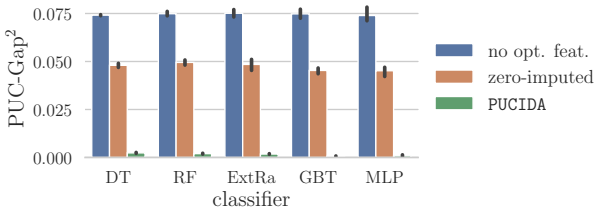


Figure 4: PUCIDA is model-agnostic. The PUC-gaps are close to zero when applying our technique across a variety of common models on the simulated dataset.

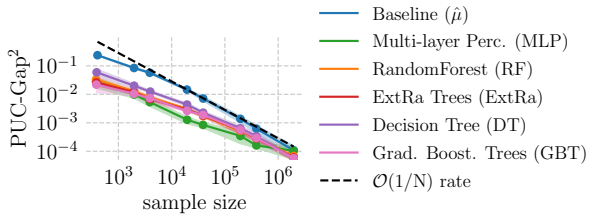


Figure 5: Convergence rate of models under PUCIDA. The estimate of PUC converges to the true value at a rate of $\mathcal{O}(\frac{1}{N})$ for the baseline estimator $\hat{\mu}$ and other commonly used models.

study these datasets in the setting of strategic withholding.

Evaluating PUCIDA

Availability Inference Restriction is violated by full feature models. First, we demonstrate the effect that full feature models have on Availability Inference Restriction. We follow common practices and use zero-imputation to deal with unavailable feature values (Emmanuel et al. 2021). Then, we train a Random Forest model on all features of the dataset where we have introduced stochastic availability into one feature (see previous paragraph). We also train a base feature model that fully drops the optional feature from the dataset. We consider the subset of individuals with unavailable feature values (i.e., a=0) and report the average loss and absolute prediction of the positive class for both models in Table 2. We observe that the full feature models use the information contained in the missingness to obtain a lower loss. This can reduce the chance of obtaining the positive outcome from the full feature model compared to the base feature model by significant margin of up to 18 % for non-sharers. Hence, these results impressively show how the full feature model implicitly infers information from missingness and thereby violates protection requirements. This stays the same when applying established fairness constraints on the models (see Appendix F.1). In contrast, when applying PUC using PUCIDA this gap vanishes or is significantly reduced. We show that the same effect can be observed independently of the imputation techniques, the model class, and the model hyperparameters in Appendix F.3.

Evaluating the Theoretical Bounds

PUCIDA guarantees Predictive Non-Degradation. Usually model performance degrades when training models with additional constraints (e.g., see Corbett-Davies et al. (2017)). To measure model performance, we use the misclassification rate for classification tasks (ROC-AUC scores lead to qualitatively similar results, see Appendix F.4) and the MSE for regression tasks. The results in Table 3a confirm that PUCIDA (using exhaustive augmentation) improves over the base feature model, suggesting that PUCIDA models benefits from using optional information. This is the case even under under strategic actions where users only provide data if it improves their outcome, and aligns with our theoretical result in Corollary 1. Under non-strategic actions, the performance figures show the same characteristics (Appendix F.4). As expected, PUC-compliant models fare moderately worse than full feature models which have no protection requirements.

We now compare two different PUCIDA variants on multiple optional features: the first strategy ensures a fixed dataset size, i.e., the number of samples is equivalent to the original dataset size. The second strategy, which uses ex-

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

task	data	opt. feature	base model	PUCIDA	Full feature model
C	diab.	Glucose	29.30% ±0.62	**26.61%** ±0.56	23.41% ±0.69
C	compas	#priors	42.89% ±0.10	**40.85%** ±0.15	36.67% ±0.36
C	adult	edu-num	16.05% ±0.03	**15.94%** ±0.05	14.86% ±0.06
R	income	WKHP	85.07 ±0.17	**80.22** ±0.15	73.25 ±0.16
R	calif.	m_income	15.62 ±0.14	**14.79** ±0.08	13.40 ±0.03
R	insurance	experience	262.43 ±0.21	**254.35** ±0.39	236.92 ±0.42

(a) One dimensional case, strategic withholding. Metrics: C: (1-Acc)×100, R: MSE×100

		Fair models				Full feature model
task	data (# opt.)	Base feature model	PUCIDA (f)	PUCIDA (e)	(×)	zero-imputed
C	diab. (2)	29.74 ±2.92	26.23 ±4.42	**25.58** ±3.69	2.2	24.16 ±4.18
C	compas (5)	40.83 ±0.56	37.65 ±0.23	**37.21** ±0.71	7.6	36.86 ±1.20
C	adult (5)	17.98 ±0.37	15.35 ±0.36	**15.27** ±0.25	7.9	15.15 ±0.33
R	income (3)	52.40 ±0.92	**49.47** ±1.71	51.21 ±0.86	3.4	46.15 ±1.60
R	calif. (4)	6.64 ±0.79	6.83 ±0.32	**6.36** ±0.08	5.1	5.69 ±0.22
R	insurance (3)	271.72 ±4.14	**242.99** ±4.47	260.77 ±2.74	3.2	232.59 ±2.39

(b) r-dimensional case. Metrics: C: (1-Acc)×100, R: MSE×100

Table 3: PUC-compliant models leverage optional information to improve predictive performance relative to base feature models This is in line with Corollary 1. In the bottom table, two strategies are considered to achieve PUC: *fixed-size (f)* and *exhaustive (e) PUCIDA*. When using exhaustive PUCIDA, the predictive performance is always better than the performance of the base feature model, and often similar to the performance of the full feature models.

haustive data augmentation, leads to an increased dataset size. The factor by which the dataset size is increased is indicated by (×) along with the results in Table 3b. We observe that competitive results can often be obtained without any dataset increase; fixed-size PUCIDA even outperforms the exhaustive variant on the larger income and the insurance dataset, whereas the exhaustive augmentation leads to a more reliable performance increase. We study the performance for sharers in Table 6 (Appendix) and find that it remains on par with the full feature model. Overall, our results demonstrate that optional information can be leveraged in a conscious way through PUC-inducing data augmentation without suffering from prohibitive performance decrease for the decision maker and the sharers.

Convergence of PUCIDA. Finally, we study the convergence behavior of PUCIDA. As a measure of approximation quality, we use the PUC-Gap[2] defined in Equation (7), which measures the squared deviation from perfect PUC. As this notion requires the knowledge of the ground truth distribution, we use a synthetic dataset for this experiment. The dataset consists of eight binary features (five base, three optional). All features in this dataset are sampled independently. Labels are induced via a logistic distribution, and availability of the optional information depends on the label. For experiments on a second synthetic dataset with five continuous features (two base, three optional) and more details, see Appendix F.5.

First, we observe that PUCIDA is model agnostic, i.e., it works with a variety of state-of-the-art models leading to negligible PUC-gaps (see Figure 4). Second, we verify that the PUC-Gaps converge to zero at the rate of $\mathcal{O}(\frac{1}{N})$ as the sample size increases (Figure 5), confirming what we derived in Theorem 4. While common models (e.g., RandomForest, MLP) have a lower error than the baseline estimator $\hat{\mu}$ the models approach the baseline estimator with larger datasets and the gap closes at the suggested rate.

Conclusion and Future Work

In this work, we studied machine learning predictions where users have the option to disclose optional information. To comply with legal regulations and respect user consent, we introduced the notion of Protected User Consent (PUC) that strikes a balance between the interests of sharers, non-sharers, and decision-makers. We demonstrated that leveraging optional information from consenting users through PUC results in superior performance compared to models that disregard the optional information entirely.

Our work gives raise to several follow-up questions. It would be interesting to study possible long-term effects of PUC and how PUC incentivizes improvements. Furthermore, we have only considered users that act entirely strategic or on privacy grounds. Modeling heterogeneous users, who might be willing to accept a certain increase in costs in return for their privacy could be a meaningful extension.

Additional Material

An extended version of this work including technical appendices is available online[2]. We also publish our code as an open-source project[3].

[2] https://arxiv.org/abs/2210.13954

[3] https://github.com/tleemann/protectedconsent

References

Aleryani, A.; Wang, W.; and De La Iglesia, B. 2020. Multiple imputation ensembles (MIE) for dealing with missing data. *SN Computer Science*, 1(3): 1–20.

Biega, A. J.; Potash, P.; Daumé, H.; Diaz, F.; and Finck, M. 2020. Operationalizing the legal principle of data minimization for personalization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 399–408.

Chouldechova, A. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2): 153–163.

Corbett-Davies, S.; Pierson, E.; Feller, A.; Goel, S.; and Huq, A. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, 797–806.

DeMarco, J. 2023. Nearly 70% of Americans Would Wear a Fitness Tracker/Smartwatch for Discounted Health Insurance.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.

Emmanuel, T.; Maupong, T.; Mpoeleng, D.; Semong, T.; Mphago, B.; and Tabona, O. 2021. A survey on missing data in machine learning. *Journal of Big Data*, 8(1): 1–37.

Fernando, M.-P.; Cèsar, F.; David, N.; and José, H.-O. 2021. Missing the missing values: The ugly duckling of fairness in machine learning. *International Journal of Intelligent Systems*, 36(7): 3217–3258.

Fogliato, R.; Chouldechova, A.; and G'Sell, M. 2020. Fairness evaluation in presence of biased noisy labels. In *International Conference on Artificial Intelligence and Statistics*, 2325–2336. PMLR.

GDPR. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. *Official Journal of the European Union*.

Ginart, A.; Guan, M. Y.; Valiant, G.; and Zou, J. 2019. Making AI Forget You: Data Deletion in Machine Learning. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*.

Golatkar, A.; Achille, A.; and Soatto, S. 2020. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goldsteen, A.; Ezov, G.; Shmelkin, R.; Moffie, M.; and Farkash, A. 2021. Data minimization for GDPR compliance in machine learning models. *AI and Ethics*, 1–15.

Hardt, M.; Megiddo, N.; Papadimitriou, C.; and Wootters, M. 2016. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, 111–122.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Henning, L. 2022. Wellness apps and fitness trackers: Why insurers love your smartwatch. *Sydney Morning Herald*.

Izzo, Z.; Anne Smart, M.; Chaudhuri, K.; and Zou, J. 2021. Approximate Data Deletion from Machine Learning Models. In Banerjee, A.; and Fukumizu, K., eds., *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130. PMLR.

Jeong, H.; Wang, H.; and Calmon, F. P. 2022. Fairness without imputation: A decision tree approach for fair prediction with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9558–9566.

Kachuee, M.; Karkkainen, K.; Goldstein, O.; Darabi, S.; and Sarrafzadeh, M. 2020. Generative imputation and stochastic prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kilbertus, N.; Rodriguez, M. G.; Schölkopf, B.; Muandet, K.; and Valera, I. 2020. Fair decisions despite imperfect predictions. In *International Conference on Artificial Intelligence and Statistics*, 277–287. PMLR.

Kilbertus, N.; Rojas Carulla, M.; Parascandolo, G.; Hardt, M.; Janzing, D.; and Schölkopf, B. 2017. Avoiding discrimination through causal reasoning. *Advances in neural information processing systems*, 30.

Kleinberg, J.; Mullainathan, S.; and Raghavan, M. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

Krishnaswamy, A. K.; Li, H.; Rein, D.; Zhang, H.; and Conitzer, V. 2021. Classification with strategically withheld data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 5514–5522.

Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual fairness. *Advances in neural information processing systems*, 30.

Lipton, Z.; McAuley, J.; and Chouldechova, A. 2018. Does mitigating ML's impact disparity require treatment disparity? *Advances in neural information processing systems*, 31.

Madison, K.; Schmidt, H.; and Volpp, K. G. 2013. Smoking, obesity, health insurance, and health incentives in the Affordable Care Act. *Jama*, 310(2): 143–144.

Mittelstadt, B. D.; Allo, P.; Taddeo, M.; Wachter, S.; and Floridi, L. 2016. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2): 2053951716679679.

Nilforoshan, H.; Gaebler, J. D.; Shroff, R.; and Goel, S. 2022. Causal conceptions of fairness and their consequences. In *International Conference on Machine Learning*, 16848–16887. PMLR.

OAG, C. 2021. CCPA regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*.

Pawelczyk, M.; Leemann, T.; Biega, A.; and Kasneci, G. 2023. On the Trade-Off between Actionable Explanations and the Right to be Forgotten. In *International Conference on Learning Representations (ICLR)*.

Rastegarpanah, B.; Crovella, M.; and Gummadi, K. P. 2020. Fair inputs and fair outputs: The incompatibility of fairness in privacy and accuracy. In *Adjunct Publication of the 28th*

ACM Conference on User Modeling, Adaptation and Personalization, 260–267.

Rastegarpanah, B.; Gummadi, K.; and Crovella, M. 2021. Auditing black-box prediction models for data minimization compliance. *Advances in Neural Information Processing Systems*, 34: 20621–20632.

Rateike, M.; Majumdar, A.; Mineeva, O.; Gummadi, K. P.; and Valera, I. 2022. Don't Throw it Away! The Utility of Unlabeled Data in Fair Decision Making. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 1421–1433.

Reeder, B.; and David, A. 2016. Health at hand: A systematic review of smart watch uses for health and wellness. *Journal of biomedical informatics*, 63: 269–276.

Rong, Y.; Leemann, T.; Borisov, V.; Kasneci, G.; and Kasneci, E. 2022. Evaluating feature attribution: An information-theoretic perspective. In *International Conference on Machine Learning*, 18770 – 18795.

Statista. 2023. Wearable Shipments Worldwide.

US Government. U.S. Centers for Medicare & Medicaid Services. 2023. How insurance companies set health premiums.

Verma, S.; and Rubin, J. 2018. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (Fairware)*, 1–7. IEEE.

Wachter, S.; and Mittelstadt, B. 2019. A right to reasonable inferences: re-thinking data protection law in the age of big data and AI. *Colum. Bus. L. Rev.*, 494.

Wang, Y.; and Singh, L. 2021. Analyzing the impact of missing values and selection bias on fairness. *International Journal of Data Science and Analytics*, 12(2): 101–119.

Wu, Y.; Dobriban, E.; and Davidson, S. 2020. DeltaGrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning (ICML)*, 10355–10366. PMLR.

Zhang, S.; Qin, Z.; Ling, C. X.; and Sheng, S. 2005. " Missing is useful": Missing values in cost-sensitive decision trees. *IEEE transactions on knowledge and data engineering*, 17(12): 1689–1693.

Zhang, Y.; and Long, Q. 2021. Assessing Fairness in the Presence of Missing Data. *Advances in neural information processing systems*, 34: 16007–16019.

Zimmer, M.; Kumar, P.; Vitak, J.; Liao, Y.; and Chamberlain Kritikos, K. 2020. 'There's nothing really they can do with this information': unpacking how users manage privacy boundaries for personal fitness information. *Information, Communication & Society*, 23(7): 1020–1037.

A Related Work and other Fairness Notions

A.1 Additional Related Work

Estimation of causal effects in the presence of missing data. The works by Mohan, Thoemmes, and Pearl (2018); Mohan and Pearl (2021) introduce graphical models for incomplete data and study the consistent estimation of causal effects amidst missing values. Our work differs as we are not concerned with estimating true causal effects but focus on building a definition of fairness in the presence of optional data.

Implementing Fairness in ML Systems. There are different strategies to implement fairness and mitigate bias in practical decision-making systems. This can be done by adding additional constraints to the optimization problem (e.g., Zafar et al. (2019)). To solve the such an optimization problem, one can employ the reductions approach (Agarwal et al. 2018), where the fairness constraint is reduced to a series of classification problems with different costs assigned to each sample. Furthermore, another line of work consists of preprocessing approaches to obtain models that are compliant with classical fairness notions. These work through sample selection (Roh et al. 2021; Abernethy et al. 2022) and reweighting approaches (Chai and Wang 2022; Li and Vasconcelos 2019; Li and Liu 2022) or through resampling of the sensitive attribute (Romano, Bates, and Candes 2020). While these approaches can help fulfill common fairness notions, they cannot easily be applied to obtain PUC.

Trade-offs between Privacy and Fairness. Possible trade-offs between classical notions of privacy such as Differential Privacy (DP, Dwork et al. 2006) have been previously studied (Bagdasaryan, Poursaeed, and Shmatikov 2019; Ganev, Oprisanu, and De Cristofaro 2022; Amiri et al. 2022), showing that imposing DP may lead to disparate outcomes across sensitive groups or reinforce existing biases. Suriyakumar et al. (2021) recently found that imposing privacy constraints can lead to an undue influence of majoritiy groups over miniories, thus possibly impacting fairness. Although we are considering personal data in this work, this paper differs from classical privacy literature because we are not concerned with data leakage. Instead, we strive to give users a better choice of which data to provide in the first place.

A.2 Common fairness notions are not applicable

As many definitions of fair outcomes between an advantaged and a disadvantaged group exist, we investigate whether existing definitions can readily be applied or easily adapted to the optional feature setting considered in this work. In other words, here we study whether existing fairness notions comply with our desiderata of Optimality and Availability Inference Restriction. In the conventional fairness literature, the impact of a sensitive attribute on the prediction is restricted. However, in the optional feature setting the point of departure is different since the optional feature may contain discriminative information that we explicitly want to use in come cases (recall that the sharers would like to obtain the most accurate prediction given their information). If not stated otherwise, we consider the availability feature A (see Figure 2) to be the sensitive attribute. We denote the predicted label by $\hat{Y}$ and discuss binary labels $Y \in \{0, 1\}$ as in most of the original definitions.

Fairness through Unawareness. This notion demands that the availability indicator A is not used as an explicit input in the decision-making process. Removing explicit information on the availability can be done easily by dropping the feature A. This makes "Fairness through Unawareness" very easy to implement. However, the group information is still implicitly encoded in the optional feature through the value `N/A` (see Fig. 2). A sufficiently complex classifier can infer this group information and include it into its decision-making. Therefore, this fairness notion cannot be applied in the optional feature setting as it violates Availability Inference Restriction.

Predictive Parity. This notion of fairness constrains the False Discovery Rates to be equal across groups, i.e., $P(Y = 0|\hat{Y} = 1, A = 0) = P(Y = 0|\hat{Y} = 1, A = 1)$. We argue that this definition and other error rate-based ones will not work in our setup because they bound performance an thus violate Optimality. It is desired by the sharers and the decision maker that the predictions will be more accurate when the feature z is present ($A = 1$) because the information in z should explicitly be used in the decision-making process if users decide to share their data on the optional features. One can make an analogous argument for other error-rate based notions such as equalized odds and equal opportunity.

Equalized Odds and Equal Opportunity (Hardt, Price, and Srebro 2016). Equalized odds requires the predicted label $\hat{Y}$ and the the protected attribute A to be conditionally independent given the true label Y. Formally, this means $P(\hat{Y}|A, Y) = P(\hat{Y}|Y)$ for all values of $Y, A, \hat{Y}$. This effectively constrains the true and false positive rates to be equal across groups. However, by the desideratum of Optimality, it is required to use class-discriminative information in the optional feature, which will necessarily lead to lower missclassification rates for subjects with $A{=}1$. Another fairness notion is Equal Opportunity which is a relaxation of Equalized Odds that only demands $P(\hat{Y}|A{=}1, Y{=}1) = P(\hat{Y}|A{=}0, Y{=}1)$, thus constraining the true positive rates across groups. To fulfill this notion, for $A{=}1$, the true positive rate would have to be kept artificially low to match that of the case $A{=}0$, with less information. This would thus result in a lower $P(\hat{Y}{=}1|A{=}0, Y{=}1)$, than could be achieved otherwise. Let $Y{=}1$ be the desirable outcome (e.g., being assigned a low insurance quote); this means that less subjects are rewarded with the justified positive outcome. This is incompatible with our desideratum of Optimality.

Statistical Parity (Dwork et al. 2012; Kusner et al. 2017). This definition is satisfied by a classifier if subjects in both protected and non-protected groups have an equal probability of getting a positive classification outcome: $P(\hat{Y}{=}1|A{=}0) = P(\hat{Y}{=}1|A{=}1)$. If the set of people providing additional information has more favorable base features in general, this definition may lead to different thresholds where people that choose to provide information are getting a lower score to achieve parity. This

definition would even forbid using this base features' full distinctive power, because one has to equalize over both missingness classes, thus contradicting Optimality.

Individual fairness (Dwork et al. 2012). Fairness definitions in this category use a distance metric m to define similarities $m(\mathbf{x}_i, \mathbf{x}_j)$ between individuals x_i and x_j. Considering the application in mind, the sensitive attributes should not play a role in determining the distance. The classifier output distributions for $f(\mathbf{x}_j)$ and $f(\mathbf{x}_i)$ that are compared by some divergence D should not differ more than the distance between these individuals, i.e., $D(f(\mathbf{x}_j), f(\mathbf{x}_i)) \leq m(\mathbf{x}_i, \mathbf{x}_j)$ (Dwork et al. 2012). In the considered setting, following the proposition by Verma and Rubin (2018), we could define the distance to be 0, if individuals have the same base features $\mathbf{b}$. This would effectively constrain the classification outcome to be identical independently of the optional feature specified, effectively prohibiting its use. Even when defining other distance metrics, the classification outcome will still be constrained to a certain range, again contradicting our desideratum of Optimality.

(Conditional) Statistical Parity. Statistical parity (SP) is known to be notoriously unfair on an individual level (Dwork et al. 2012). Therefore, Corbett-Davies et al. (2017) define the notion of conditional statistical parity (CSP), which is an extension of SP, where some attributes are allowed to affect the decision. If we allow all base features $\mathbf{b}$, the resulting definition expressed in expectations would be $P[\hat{Y}|\boldsymbol{B} = \mathbf{b}, A = 0] = P[\hat{Y}|\boldsymbol{B} = \mathbf{b}, A = 1]$. While this definition can be compliant with Availability Inference Restriction, we show that CSP-compliant models cannot meet the desire of the sharers for most accurate predictions. They cannot assign most accurate predictions to sharers or would encounter prohibitively high costs due to the CSP constraint if they did so. Indeed, they can be worse than the performance of a base feature model, even when we assign the sharers the most accurate predictions. This is the case even under idealized conditions (i.e., known expectations) and when Incentivization is perfectly fulfilled. PUC-models do not suffer from this limitations and can assign sharers most accurate predictions while always matching the performance of the base feature model.

Lemma 1 (CSP-compliant models can degrade model performance over base feature model)**.** *There exists a density* $\mathbf{p}$ *for which a CSP-compliant model* $f^*_{CSP} : \mathcal{X} \to [0, 1]$ *which assigns the most accurate predictions to the sharers, i.e.,* $f^*_{CSP|_{a=1}}(\mathbf{b}, a, z^*) = \mathbb{F}_{\mathcal{L}}[Y|\mathbf{b}, A = 1, z^*]$ *leads to higher expected losses (for MSE and BCE losses) than an optimal base feature model* g^*_{BCE} *:*

$$\mathbb{E}_{\mathbf{p}}[\mathcal{L}(g^*_{BCE}(\mathbf{B}), Y)] < \mathbb{E}_{\mathbf{p}}[\mathcal{L}(f^*_{CSP}(\mathbf{B}, A, Z^*), Y)]. \tag{8}$$

A proof is provided in Appendix D.1. There, we give a density $\mathbf{p}$ that serves as such a counterexample. We argue that this fairness notion is incompatible with the desire of the sharers for accurate predictions and the decision makers desire for low overall costs. Thus, we have established that common fairness definitions fail to conform to our desiderata of Availability Inference Restriction or result in models with unreasonable performance characteristics.

B Intuition and Additional Examples

In this section, we provide a simple example to show the problem of possible unfairness and provide more intuition for our notion of Protected User Consent.

B.1 Standard losses may lead to unfair treatment

We revisit the example of college admission, to show how imputation leads to possibly unfair treatment. Suppose we are given the samples $\left\{(\mathbf{x}^i, y^i)\right\}_{i=1..N}$ with $N = 5$ from Fig. 2. Using the standard Mean-Squared Error (MSE) loss, we solve the following empirical risk minimization problem:

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} (f(\mathbf{x}^i) - y^i)^2,$$

with a sufficiently expressive function class $\mathcal{F}$. For samples in the example data set, this will yield the outcome $f^*(\mathbf{x}) = \frac{1}{|\{\mathbf{x}^i = \mathbf{x}\}|} \sum_{\{\mathbf{x}^i = \mathbf{x}\}} y_i$, the empirical mean. Consider the samples $\mathbf{x}^1$ and $\mathbf{x}^4$ in Fig. 2. Candidate 1 chose to share an additional feature, while candidate 4 did not. Although they have the same base features, their classification will be different at test time (as the data in the training set) with $f^*(\mathbf{x}^1) = 3k\$$ and $f^*(\mathbf{x}^4) = 64k\$$.

We argue that in the case of candidate 4, the availability information was implicitly used to compute the score and resulted in a lower outcome. If only the base features had been available, i.e., f^* would have been trained on the data set $\left\{(\mathbf{b}^i, y^i)\right\}_{i=1..k}$, the model outcome would be $f^*(\mathbf{b}) = \frac{1}{|\{\mathbf{b}^i = \mathbf{b}\}|} \sum_{\{\mathbf{b}^i = \mathbf{b}\}} y^i$ with $f^*(\mathbf{b}^4) = 24k\$$ which is the dataset average for all customers from NSW with a basic coverage plan. In this work, we argue that the unavailability of certain features itself should not be used in the determination of the model outcome when no additional information is available.

B.2 Example: Missing at random (MAR) data.

For missing at random data (Rubin 1976), the likelihood of unavailability can be entirely accounted for by the observed base features $\mathbf{b}$ and is not affected by the partially observed z and the label y. Formally, for a single optional feature with random

availability A, $\mathbf{p}(A=0|\mathbf{b}) = \mathbf{p}(A=0|\mathbf{b}, z, y)$ for every $z \in \mathcal{X}^z, y \in \mathcal{Y}$. Therefore,

$$\mathbf{p}\left(y|\mathbf{b}, A=0\right) = \frac{\mathbf{p}(y, \mathbf{b}, A=0)}{\mathbf{p}(\mathbf{b}, A=0)} = \frac{\mathbf{p}(y)\mathbf{p}(\mathbf{b}|y)\mathbf{p}(A=0|\mathbf{b}, y)}{\sum_{y'} \mathbf{p}(y')\mathbf{p}(\mathbf{b}|y')\mathbf{p}(A=0|\mathbf{b}, y')} \tag{9}$$

$$= \frac{\mathbf{p}(y)\mathbf{p}(\mathbf{b}|y)\mathbf{p}(A=0|\mathbf{b})}{\sum_{y'} \mathbf{p}(y')\mathbf{p}(\mathbf{b}|y')\mathbf{p}(A=0|\mathbf{b})} = \frac{\mathbf{p}(y)\mathbf{p}(\mathbf{b}|y)\mathbf{p}(A=0|\mathbf{b})}{\mathbf{p}(A=0|\mathbf{b})\sum_{y'} \mathbf{p}(y')\mathbf{p}(\mathbf{b}|y')} \tag{10}$$

$$= \frac{\mathbf{p}(y)\mathbf{p}(\mathbf{b}|y)}{\sum_{y'} \mathbf{p}(y')\mathbf{p}(\mathbf{b}|y')} = \mathbf{p}(y|\mathbf{b}). \tag{11}$$

Therefore, we also have $\mathbb{E}\left[Y|B=\mathbf{b}, A=0\right] = \mathbb{E}_{\mathbf{p}}\left[Y|B=\mathbf{b}\right]$, indicating that the missingness does not affect the expected value of the label (or that of any other functional of $p(y|\mathbf{b})$) over the entire data distribution. Therefore, a perfect discriminative model with $f(\mathbf{x}) = \mathbb{F}^{\mathcal{L}}_{\mathbf{p}}\left[Y|\mathbf{x}\right]$ will fulfill Theorem 1, our definition of PUC, right away.

B.3 A probabilistic derivation of Non-Penalization

First, we require that the predictor does not use the information as an explicit input, i.e., the predictor should behave as if it only used base features $\mathbf{b}$ via some function $g : \mathcal{X}^b \to \mathcal{Y}$:

$$f_{|a=0}(\mathbf{b}, a, z^*) = g(\mathbf{b}). \tag{12}$$

For (b), although a=0 is not an explicit input to g, a sufficiently complex function may still be implicitly adapting to the group a=0 and thus incorporate information that the user did not give their consent to. Therefore, on its own, the constraint in eqn. (12) is insufficient to enforce Availability Inference Restriction, and we need to formally define which predictors g are not specific to the information provided by the group of non-consenting users. To make matters more concrete, we first consider the case of binary classification with $\mathcal{Y} \in \{0, 1\}$ from a probabilistic perspective and suppose $g(\mathbf{b})$ returns a numerical probability socre in $[0, 1]$. We let $\mathbf{p}_g(\hat{Y}|\mathbf{b})$ denote stochastic predictions $\hat{Y}$ defined through g where $\mathbf{p}_g(\hat{Y}=1|\mathbf{b}) \coloneqq g(\mathbf{b})$. We would like to constrain the information contained in the predictions $G(\mathbf{b})$ to not use any additional information that the users did not actively consent to. To come up with a suitable constraint, we consider two simple others predictors, where one should be allowed and the other should be ruled out.

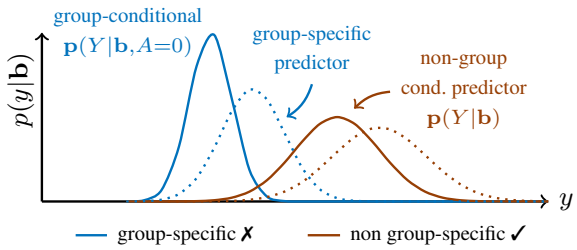


Figure 6: In our definition, predictors are called group-specific (blue) if they are closer to the group conditional distribution than to the overall label distribution (pink). Our requirement forbids the use of such group-specific predictors for the group of users with no additional information.

There are two canonical examples: a probabilistic estimator that is certainly not adapted to a specific group would be the one matching the ground truth overall conditional probability $\mathbf{p}\left(Y|\mathbf{b}\right)$. On the other extreme, the predictor $\mathbf{q}$ equivalent to $\mathbf{q}\left(Y|\mathbf{b}\right) \coloneqq \mathbf{p}\left(Y|\mathbf{b}, A{=}0\right)$ is fully leveraging the protected information and would thus be non-compliant. Generalizing this insight, we rule out all probabilistic predictors that are closer to the most non-compliant predictor than the overall conditional predictor, which we consider valid. These forbidden, group-specific predictors are visualized in Figure 6.

To this end, a suitable distance metric is required between the predictive distributions. A common choice is the Kullback-Leibler divergence $\mathcal{D}_{\text{KL}}$, which results in the following requirement for a predictor g:

$$\mathcal{D}_{\text{KL}}\left(\mathbf{p}(Y|\mathbf{B}, A{=}0)||\mathbf{p}_g(Y|\mathbf{B})\right) \geq \mathcal{D}_{\text{KL}}\left(\mathbf{p}(Y|\mathbf{B}, A{=}0)||\mathbf{p}(Y|\mathbf{B})\right). \tag{13}$$

The condition from eqn. (13) can be equivalently stated in terms of expected loss for binary classification and regression problems; i.e., the above condition allows to derive a generalized principle of Availability Inference Restriction (see Appendix D.4 for the proof).

C Effects of user choices in PUC-compliant models

In this section, we provide a brief discussion on the effect a user's choice to provide or not provide optional information has for the decision maker and for the affected end users.

Balancing the interests of consenting users, non-consenting users and decision makers. From the user's viewpoint, we would like to outline that a user's choice to provide optional information or not may depend on different factors:

(a) **Relevance of information:** How relevant does the individual deem the information that is asked for. The user may (correctly or incorrectly) deem certain information irrelevant for the decision and therefore they may not be willing to provide information on optional features.

(b) **Sensitivity of information:** How concerned is the individual about the optional information being unintentionally leaked or intentionally passed on to a third party.

(c) **Prior beliefs and expected outcomes:** The user's mental models of the decision system and the role their information plays in the system could be essential as a user may be more inclined to provide information which they deem beneficial for their prediction.

In summary, the *utility* of an individual for providing data is composed of several factors, including sensitivity, perceived relevance and anticipated outcomes.

From the perspective of the decision maker, PUC models become increasingly more accurate with more data being voluntarily provided. Therefore, from the perspective of the decision maker, it is important that users can also benefit from providing optional information. This desiderata is captured by optimality requirement, which allows the decision maker to make the most of all voluntarily provided information and allows the users to obtain more accurate decisions when additional information is provided.

Can less information lead to more favorable predictive outcomes for users? A user's predictive outcome depends on which information was provided by the user, and the predictive outcomes do not need to be monotonic in the number of optional features being provided; i.e., *providing more information on optional features does not necessarily lead to a better outcome for the user*.

To see why this behavior is necessary and desired, we consider the two extreme cases on either side of the spectrum, where (a) optional information may not impact the prediction outcome, and where (b) predictions can only get worse when providing strictly less features. In case (a), where no changes to the predictions occur, the setup becomes trivial and results in the base feature model. If this is the goal, then the collection of any additional information is useless for the decision and one should refrain from collecting these data, directly following the principle of Data Minimization. In case (b), where users can only get worse predictions with less information, a machine learning model would always have to treat users, who did not provide information, as if the worst possible value of a feature was provided. This is to make sure that the prediction outcomes of consenting users remains higher than the outcomes of non-consenting users with identical features. In the real-world setting of college admission we considered throughout the main text, this would lead to severe penalization of users who did not share optional test score results. This could de-facto rule them out entirely. *We argue that this behavior is not desired as it implicitly forces users to provide their data*. If this is desired, then the decision maker should make this choice explicit and the feature should then be mandatory. To conclude, for the setting of optional features with a decision maker who is interested in providing users a real choice, any sensible notion of fairness must allow for differences in the outcomes depending on the provided information.

D Proofs

D.1 Counterexample: Conditional statistical parity can be inferior to the base model

Expressed in terms of expectations, the notion of conditional statistical parity $\mathbb{E}[\hat{Y}|\boldsymbol{B} = \mathbf{b}, A = 0] = \mathbb{E}[\hat{Y}|\boldsymbol{B} = \mathbf{b}, A = 1]$ requires the prediction averages conditioned on $\mathbf{b}$ to be equal among groups that provided the optional features and those that did not. We now consider a non-probabilistic prediction function $\hat{Y} = f(\mathbf{b}, a, z^*)$. Plugging in the functional form would result in the following definition: $f_{|\boldsymbol{a}=0}(\mathbf{b}, \boldsymbol{a} = 0, z^*) = \mathbb{E}_{Z^* \sim \mathbf{p}(Z^*|\mathbf{b}, \boldsymbol{a}=1)}\left[f(\mathbf{b}, \boldsymbol{a} = 1, Z^*)\right], \forall \mathbf{b}$. In the case $a = 0$, z^* is constrained to be `N/A` so we can ignore its value. The subscript is used to indicate the restriction of f on the set of points with $\boldsymbol{a}$=0. This definition constrains the output $f_{|m=0}$, when no additional features provided, to match the average output of the individuals that provided features.

We follow the requirement most accurate predictions by the sharers which requires $f_{|a=1}$ to be the best approximation of $\mathbb{E}\left[Y|\boldsymbol{B}, A = 1, Z^*\right]$. Thus, we would have to set $f_{|a=0}$ to be $f_{|a=0}(\mathbf{b}, a, z^*) = \mathbb{E}_{Z^* \sim \mathbf{p}(z^*|\mathbf{b}, a=1)}\left[\mathbb{E}\left[Y|\boldsymbol{B} = \mathbf{b}, A = 1, Z^*\right]\right] = \mathbb{E}\left[Y|B = \mathbf{b}, A = 1\right]$ when marginalizing over Z^*. Overall, this derivation results in a function f_{csp} of the following form:

$$f_{\text{csp}}(\mathbf{b}, a, z^*) = \begin{cases} \mathbb{E}[Y|\mathbf{b}, A = 1] & \text{if } a = 0 \\ \mathbb{E}[Y|\mathbf{b}, A = 1, Z^* = z^*], & \text{if } a = 1 \end{cases}. \tag{14}$$

In this section we present a simple example to show that this function f_{csp} derived from notion of conditional statistical parity may lead to an increased Mean-Squared-Error (MSE) and Binary Cross Entropy (BCE) loss compared to the base model (not

using the optional feature) even when the estimators of the conditional means are perfect. Note that in a binary space $\mathcal{Y} = \{0, 1\}$ for both losses, predicting the conditional expectation is optimal.

For the example, we take any value $\mathbf{b}$ and suppose $p(y|\mathbf{b}, A, Z)$ depends on A but that Z is useless and does not contribute any new information, i.e. $\forall z \in \mathcal{X}^z, A \in \{0, 1\} : p(y|\mathbf{b}, A, z) = p(y|\mathbf{b}, A)$. Furthermore, we set the outcome to be deterministic of A:

$$\mathbb{E}[Y|\mathbf{b}, A = 0] = 0 \tag{15}$$
$$\mathbb{E}[Y|\mathbf{b}, A = 1] = 1 \tag{16}$$
$$\mathbb{E}[Y|\mathbf{b}] = \mathbf{p}(A = 1|\mathbf{b})\mathbb{E}[Y|\mathbf{b}, A = 0] + \mathbf{p}(A = 0|\mathbf{b})\mathbb{E}[Y|\mathbf{b}, A = 1] = \mathbf{p}(A = 1|\mathbf{b}) \coloneqq \alpha. \tag{17}$$

Let $\mathbf{p}(A = 1|\mathbf{b}) = \alpha$ be in the range $0 < \alpha < 1$. The optimal base feature model g^* would predict:

$$g^*(\mathbf{b}) = \alpha, \tag{18}$$

whereas the model based on CSP is given by:

$$f_{csp}(\mathbf{b}) = 1, \tag{19}$$

independently of the realization of A (because it is not allowed to use this information). The expected MSE Loss is given by:

$$L_{base,MSE} = \mathbf{p}(A = 0|\mathbf{b})(\alpha - 0)^2 + \mathbf{p}(A = 1|\mathbf{b})(\alpha - 1)^2 \tag{20}$$
$$= (1 - \alpha)\alpha^2 + \alpha(1 - \alpha)^2 = (1 - \alpha)\alpha(\alpha + 1 - \alpha) = (1 - \alpha)\alpha, \tag{21}$$

$$L_{base,BCE} = -\mathbf{p}(A = 0|\mathbf{b}) \log(1 - \alpha) - \mathbf{p}(A = 1|\mathbf{b}) \log \alpha < \infty. \tag{22}$$

If the notion derived from Conditional statistical parity is used, we would use $f_{csp}(\mathbf{b}) = \mathbb{E}[Y|\mathbf{b}, A = 1] = 1$ to predict in both cases and obtain:

$$L_{csp,MSE} = \mathbf{p}(A = 0|\mathbf{b})(0 - 1)^2 + \mathbf{p}(A = 1|\mathbf{b})(1 - 1)^2 = p(A = 0|\mathbf{b}) = 1 - \alpha, \tag{23}$$

$$L_{csp,BCE} = -\mathbf{p}(A = 0|\mathbf{b}) \log(1 - 1) - \mathbf{p}(A = 1|\mathbf{b}) \log 1 = p(A = 0|\mathbf{b}) = \infty. \tag{24}$$

For the BCE, we already see that the loss is unbounded in the case of CSP. One can construct the same example with non-infinite losses by adding a slight probability of the other outcome, i.e., setting $\mathbb{E}[Y|\mathbf{b}, A = 1] = 1 - \epsilon$ with some small $\epsilon > 0$ and obtain an analogous result.

For the MSE, in every case with $\alpha = \mathbf{p}(A = 1|\mathbf{b}) < 1$ this results in:

$$L_{csp} = 1 - \alpha > (1 - \alpha)\alpha = L_{base}. \tag{25}$$

We have now shown that for an arbitrary $\mathbf{b}$, the loss can be higher that of the base feature model. We can complete the example to the overall loss over a distributions of $\mathbf{b}$'s by supposing $\mathbf{p}(\mathbf{B} = \mathbf{b}) = 1$, which would however be a degenerate distribution. As a broader alternative, one can assume the above for a set of $\mathbf{b} \in \mathcal{B}$ and suppose any probability distribution with support in $\mathcal{B}$, i.e., $\mathbf{p}(\mathbf{B} \notin \mathcal{B}) = 0$.

D.2 Proof: The notion of Protected User Consent is optimal in the set of predictors conforming to the two desiderata

We can consider both predictors (for the case with and without optional features) independently. On the one hand, the notion of Availability Inference Restriction demands that the base predictor $f_{|_{a=0}}(\mathbf{b}) = g(\mathbf{b})$ should not outperform the optimal base predictor $f^*_{\mathcal{L}}$ trained on the full data set,

$$\mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 0\right] \leq \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(g(\mathbf{B}), Y\right)|A = 0\right]. \tag{26}$$

This directly provides us with one predictor $g(\mathbf{b})$, that is optimal in terms of loss for these individual namely, $g \equiv f^*_{\mathcal{L}}$ where $f^*_{\mathcal{L}}(\mathbf{b}) = \mathbb{F}_{\mathcal{L}}[Y|\mathbf{b}]$

On the other hand, for the group of individuals with optional information, we face no constraints and thus use the best predictor possible, i.e.,

$$f_{|_{a=1}}(\mathbf{b}, a, z^*) = \mathbb{F}_{\mathcal{L}}[Y|\mathbf{b}, A = 1, z^*] = \underset{f(\mathbf{b},a,z^*)}{\arg\min}\, \mathbb{E}_{\mathbf{p}(Y|\mathbf{b},A=1,z^*)}\left[\mathcal{L}(f(\mathbf{b}, a, z^*), Y)\right]. \tag{27}$$

Together, this results in the given definition of PUC. □

D.3 Proof: 1D-PUC obeys Predictive Non-Degradation

For the case of optional features ($A = 1$), we have:

$$f^{\text{PUC}}_{|a=1}(\mathbf{b}, a, z^*) = \mathbb{F}_{\mathcal{L}}\left[Y|\mathbf{b}, A = 1, z^*\right] = \underset{f(\mathbf{b},a,z^*)}{\arg\min}\, \mathbb{E}_{\mathbf{p}(Y|\mathbf{b},A=1,z^*)}\left[\mathcal{L}(f(\mathbf{b}, a, z^*), Y)\right]. \tag{28}$$

As is is the optimal predictor, its loss on these samples is smaller than that of any model, including the optimal model on the base features. Therefore, for each $\mathbf{b}, z^*$, we have:

$$\mathbb{E}_{\mathbf{p}(Y|\mathbf{b},A=1,z^*)}\left[\mathcal{L}(f^{\text{PUC}}(\mathbf{b}, a{=}1, z^*), Y)\right] \leq \mathbb{E}_{\mathbf{p}(Y|\mathbf{b},A=1,z^*)}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{b}), Y\right)\right]. \tag{29}$$

Averaging over the entire class of samples with $A = 1$, we obtain:

$$\mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^{\text{PUC}}(\mathbf{B}, A, Z^*), Y\right)\middle|A = 1\right] \leq \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 1\right]. \tag{30}$$

On the other hand, the definition of PUC demands that the predictor in case $A = 0$ is equivalent to the optimal predictor on the base features. Thus they have equal loss and:

$$\mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^{\text{PUC}}(\mathbf{B}, A, Z^*), Y\right)\middle|A = 0\right] = \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 0\right]. \tag{31}$$

In total, we have

$$\begin{aligned}
\mathcal{L}^{\text{PUC}} &= \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^{\text{PUC}}(\mathbf{B}, A, Z^*), Y\right)\middle|A = 0\right]\mathbf{p}(A{=}0) + \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^{\text{PUC}}(\mathbf{B}, A, Z^*), Y\right)\middle|A = 1\right]\mathbf{p}(A{=}1) && (32)\\
&\leq \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 0\right]\mathbf{p}(A{=}0) + \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 1\right]\mathbf{p}(A{=}1) && (33)\\
&= \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)\right] = \mathcal{L}^{\text{base}}. && (34)
\end{aligned}$$

□

D.4 The Generalized Principle of Availability Inference Restriction

We can reformulate the probabilistic definition of Availability Inference Restriction in terms of loss functions, which allows for generalization. We define $\mathbf{p}_0 = \mathbf{p}(\mathbf{B}|A = 0)$. We start by the notion given in the definition:

$$\begin{aligned}
&\mathcal{D}_{KL}\left(\mathbf{p}(Y|\mathbf{B}, A{=}0)||\mathbf{p}_g(Y|\mathbf{B})\right) \geq \mathcal{D}_{KL}\left(\mathbf{p}(Y|\mathbf{B}, A{=}0)||\mathbf{p}(Y|\mathbf{B})\right) && (35)\\
&\mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\mathcal{D}_{KL}\left(\mathbf{p}(Y|\mathbf{b}, A{=}0)||\mathbf{p}_g(Y|\mathbf{b})\right) \geq \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\mathcal{D}_{KL}\left(\mathbf{p}(Y|\mathbf{b}, A{=}0)||\mathbf{p}(Y|\mathbf{b})\right). && (36)
\end{aligned}$$

The Kullback-Leibler divergence can be decomposed as $\mathcal{D}_{KL}(\mathbf{p}||\mathbf{q}) = H(\mathbf{p}) + CE(\mathbf{p}||\mathbf{q})$, which results in:

$$\begin{aligned}
&\iff \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\text{CE}\left(\mathbf{p}(Y|\mathbf{b}, A{=}0)||\mathbf{p}_g(Y|\mathbf{b})\right) + \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}H\left[Y|\mathbf{b}, A{=}0\right] && (37)\\
&\geq \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\text{CE}\left(\mathbf{p}(Y|\mathbf{b}, A{=}0)||\mathbf{p}(Y|\mathbf{b})\right) + \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}H\left[Y|\mathbf{b}, A{=}0\right] && (38)\\
&\iff \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\text{CE}\left(\mathbf{p}(Y|\mathbf{b}, A{=}0)||\mathbf{p}_g(Y|\mathbf{b})\right) \geq \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\text{CE}\left(\mathbf{p}(Y|\mathbf{b}, A{=}0)||\mathbf{p}(Y|\mathbf{b})\right) && (39)\\
&\iff \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\mathbb{E}_{Y\sim\mathbf{p}(Y|\mathbf{b},A=0)}\left[-\log\mathbf{p}_g(Y|\mathbf{b})\right] \geq \mathbb{E}_{\mathbf{b}\sim\mathbf{p}_0}\mathbb{E}_{Y\sim\mathbf{p}(Y|\mathbf{b},A=0)}\left[-\log\mathbf{p}(Y|\mathbf{b})\right] && (40)
\end{aligned}$$

The inner expectation is equivalent to the BCE loss for a specific $\mathbf{b}$. Averaged over all $\mathbf{b} \sim \mathbf{p}_0$ we obtain.

$$\Rightarrow \mathbb{E}_{\mathbf{p}}\left[\text{BCE}\left(g(\mathbf{B}), Y\right)|A = 0\right] \geq \mathbb{E}_{\mathbf{p}}\left[\text{BCE}\left(f^*_{\text{BCE}}(\mathbf{B}), Y\right)|A = 0\right]. \tag{41}$$

This notion allows for generalization by replacing BCE with some general loss function $\mathcal{L}$. Doing so results in

$$\mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(g(\mathbf{B}), Y\right)|A = 0\right] \geq \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}\left(f^*_{\mathcal{L}}(\mathbf{B}), Y\right)|A = 0\right], \tag{42}$$

the version of the desideratum of Availability Inference Restriction mentioned in the main paper. □

D.5 PUC under strategic withholding of data

To prove Theorem 2, we first note that the decision maker can only realize improvements over the base model in the setup of strategic interactions for individuals by offering them a lower premium than the prediction of the base model. Otherwise, they would strategically not provide their data.

It is only beneficial for the decision maker to do so if there exists an $y' \leq y_{\text{base}} := \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}]$ with a lower expected loss, i.e.,

$$\mathbb{E}_Y\left[\mathcal{L}(y', Y)|\mathbf{B} = \mathbf{b}, Z = z\right] \leq \mathbb{E}_Y\left[\mathcal{L}(y_{\text{base}}, Y)|\mathbf{B} = \mathbf{b}, Z = z\right] \tag{43}$$

Due to the convexity of the loss $\mathcal{L}$, this expected value will as well be convex in the prediction y' and we will also have $\mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, z] \leq \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}]$. The loss-minimal prediction would be $f(b, a = 1, z) = \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, z]$, which will not be hindered through strategic actions. This however results in a PUC-model again, as $\mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, z] = \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, A = 1, z]$, because the sharing decision does not influence the label given $\mathbf{B}, Z$. □

PUC with monotonicity constraints. A similar argument can be made when monotonicity constraints need to be enforced, i.e., the outcome can only decrease over the base model with more information provided. We can consider each optional feature value z separately for sharers. If the sample comes with a better average $\mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, z] \leq \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}]$ than the base prediction, we can confidently return this full-feature optimal prediction. In the contrary case, where $\mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}, z] > \mathbb{F}^{\mathcal{L}}[Y|\mathbf{b}]$, the best prediction that the decision maker is allowed to make is the base feature models prediction (due to the convexity of the loss function). This is equivalent to dropping the optional feature in this case and using the corresponding PUC model.

D.6 Equivalence of Expectations for the Resampling model

In this section, we show that the resampling technique proposed in this work converges to the desired outcome. Therefore, we show that in the infinite sample-limit, the optimum reached when optimizing the loss over the modified distribution corresponds to the desired PUC model.

We introduce the usual mapping $\mathcal{I}(\boldsymbol{a}) \coloneqq \{i \mid \boldsymbol{a}_i = 1, i = 1, \ldots, r\}$ to denote the set of all indices that are 1 in the vector $\boldsymbol{a}$ but also use $\mathbb{I}_S$ to denote the binary indicator vector where all components corresponding to indices in S are set to 1 and to zero otherwise, i.e., $(\mathbb{I}_S)_i = \{1 \text{ if } i \in S, \text{else } 0\}$. Note that these operations invert each other such that $\mathbb{I}_{\mathcal{I}(\boldsymbol{a})} = \boldsymbol{a}$. We can show that the optimal prediction $\hat{y} = \hat{y}(\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*)$ is given by:

$$\hat{y} = \mathbb{F}^{\mathcal{L}}_{((\mathbf{b},\boldsymbol{a},\boldsymbol{z}^*),y)\sim\overline{\mathbf{p}}}[Y|\mathbf{b}, \boldsymbol{A} = \boldsymbol{a}, \boldsymbol{Z}^* = \boldsymbol{z}^*] = \arg\min_{\hat{y}} \mathbb{E}_{((\mathbf{b},\boldsymbol{a},\boldsymbol{z}^*),y)\sim\overline{\mathbf{p}}}[\mathcal{L}(\hat{y}, y)|\mathbf{b}, \boldsymbol{A} = \boldsymbol{a}, \boldsymbol{Z}^* = \boldsymbol{z}^*] \tag{44}$$

$$= \arg\min_{\hat{y}} \sum_{\mathcal{I}(\boldsymbol{a})\subseteq S} \mathbf{p}(\boldsymbol{A}{=}\mathbb{I}_S|\mathbf{B}{=}\mathbf{b}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})}{=}\boldsymbol{z}_{\mathcal{I}(\boldsymbol{a})})\mathbb{E}_{\mathbf{p}}\left[\mathcal{L}(\hat{y}, y)\middle|\boldsymbol{A} = \mathbb{I}_S, \mathbf{B} = \mathbf{b}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})} = \boldsymbol{z}_{\mathcal{I}(\boldsymbol{a})}\right] \tag{45}$$

$$= \arg\min_{\hat{y}} \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}(\hat{y}, y)\middle|\mathcal{I}(\boldsymbol{a}) \subseteq \mathcal{I}(\boldsymbol{A}), \mathbf{B} = \mathbf{b}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})} = \boldsymbol{z}_{\mathcal{I}(\boldsymbol{a})}\right] \tag{46}$$

$$= \arg\min_{\hat{y}} \mathbb{E}_{\mathbf{p}}\left[\mathcal{L}(\hat{y}, y)\middle|\boldsymbol{A}_{\mathcal{I}(\boldsymbol{a})} = \mathbf{1}, \mathbf{B} = \mathbf{b}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})} = \boldsymbol{z}_{\mathcal{I}(\boldsymbol{a})}\right] \tag{47}$$

$$= \mathbb{F}^{\mathcal{L}}_{\mathbf{p}}\left[Y|\mathbf{B} = \mathbf{b}, \boldsymbol{A}_{\mathcal{I}(\boldsymbol{a})} = \mathbf{1}, \boldsymbol{Z}_{\mathcal{I}(\boldsymbol{a})} = \boldsymbol{z}_{\mathcal{I}(\boldsymbol{a})}\right]. \tag{48}$$

In Equation (45), we use the fact that we can express the distribution $\overline{\mathbf{p}}$ for a subset of inputs with $\boldsymbol{A} = \boldsymbol{a}$ as a mixture of $\mathbf{p}$, averaged over all subsets of inputs S with more optional features than $\boldsymbol{a}$, weighted equally but with the optional information erased. This is a result of the data augmentation procedures that defines $\overline{\mathbf{p}}$. The total weight is just a factor and does not play a role in the $\arg\min$ operation. The following steps are just reformulations of the expression. □

D.7 Proof: Convergence of the sample approximation for a finite feature space

In this section we provide a general estimation of the error of a non-parametric regressor from a finite number of samples on a finite feature space $\mathcal{X}$ (e.g., finite, discrete features) and a label space $\mathcal{Y}$ that can be either continuous or discrete. Before we can prove the main result, we establish the following lemma.

Lemma 2. *The density $\overline{\mathbf{p}}$ that is obtained from $\mathbf{p}$ by applying the augmentation strategy described in the paper (`PUCIDA`) is related to the original density through the following relation:*

$$\forall \mathbf{x} \in \mathcal{X} : \overline{\mathbf{p}}(\mathbf{x}) \geq \frac{1}{2^r}\mathbf{p}(\mathbf{x}).$$

In particular, this implies that the support of $\overline{\mathbf{p}}$ is at least as big as the support of $\mathbf{p}$.

Proof. The resampling procedure consists of two steps. First, a reweighting is done. As we state in the main text, this reweighting from $\overline{\mathbf{p}}$ can be implemented through rejection sampling with samples from $\mathbf{x} = (\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*) \sim \mathbf{p}$. Samples are passed on the the next stage with a probability of $\frac{2^{|\mathcal{I}(\boldsymbol{a})|}}{2^r}$. Using this scheme, we know that for a certain $\mathbf{x} = (\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}^*)$, the probability of the sample to be observed after applying only the reweighting step is bounded by $\frac{2^{|\mathcal{I}(\boldsymbol{a})|}}{2^r}\mathbf{p}(\mathbf{x})$. To see this, we can consider the worst case, where all other samples are passed on with probability 1 and only the considered vector $\mathbf{x}$ is downweighted by a factor of $\frac{2^{|\mathcal{I}(\boldsymbol{a})|}}{2^r}$. If the other samples are also downweighted, this is a strict lower bound. In the second step, some optional features are dropped at random with a probability of $\frac{1}{2}$. We are interested in $\overline{\mathbf{p}}(\mathbf{x})$, the probability of obtaining the exact original sample with all its optional features still present. The probability that all optional features remain present with the Bernoulli distribution used, is given by $\frac{1}{2^{|\mathcal{I}(\boldsymbol{a})|}}$. Bringing it all together we obtain:

$$\forall \mathbf{x} \in \mathcal{X} : \overline{\mathbf{p}}(\mathbf{x}) \geq \frac{2^{|\mathcal{I}(\boldsymbol{a})|}}{2^r}\frac{1}{2^{|\mathcal{I}(\boldsymbol{a})|}}\mathbf{p}(\mathbf{x}) = \frac{1}{2^r}\mathbf{p}(\mathbf{x}). \tag{49}$$

Theorem 5 (Convergence of Finite Sample Approximation)**.** *Suppose a finite feature space $\mathcal{X}$ and a numerical label space $\mathcal{Y} \subseteq \mathbb{R}$. Suppose all conditional expectations $\mu_{\overline{\mathbf{p}}}(\mathbf{x}) \coloneqq \mathbb{E}_{\overline{\mathbf{p}}}[y|\mathbf{x}]$ and the conditional variances $\sigma^2_{\overline{\mathbf{p}}}(\mathbf{x}) \coloneqq \mathrm{Var}_{\overline{\mathbf{p}}}[y|\mathbf{x}]$ exist (and thus are finite). We can estimate a (discrete) non-parametric regressor $\hat{\mu}^{\mathcal{D}} : \mathcal{X} \mapsto \mathbb{R}$ from a finite number N of independent, identically distributed observations $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1\ldots N}$ from $\overline{\mathbf{p}}$ which satisfies:*

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{p},\mathcal{D}\sim\overline{\mathbf{p}}}\left[\left(\hat{\mu}^{\mathcal{D}}(\mathbf{x}) - \mu_{\overline{\mathbf{p}}}(\mathbf{x})\right)^2\right] \leq \frac{2^r|\mathcal{X}|^2(\sigma^2_{\max} + \mu^2_{\max})}{N} + \mathcal{O}\left(\frac{1}{N^2}\right), \tag{50}$$

where $\sigma^2_{\max} \coloneqq \max_{\mathbf{x}\in\mathcal{X}} \sigma^2_{\overline{\mathbf{p}}}, (\mathbf{x})$ and $\mu^2_{\max} \coloneqq \max_{\mathbf{x}\in\mathcal{X}} \mu^2_{\overline{\mathbf{p}}}(\mathbf{x})$. The expected squared deviation to the optimal estimator converges at an order of $\mathcal{O}\left(\frac{1}{N}\right)$.

Proof. Before we proof the rate of convergence, we first define the estimator for which we establish this bound. We can draw N samples $\mathcal{D} \sim (\mathbf{x}_i, y_i) \sim \overline{\mathbf{p}}$. Then, we split these into $|\mathcal{X}|$ equal batches of size $M = \left\lfloor \frac{N}{|\mathcal{X}|} \right\rfloor$ samples. We can thus assign each possible feature value $\mathbf{x} \in \mathcal{X}$ a batch $\text{Batch}(\mathbf{x}) \subset [N]$, of samples, although the value the features $\mathbf{x}_i$ for i in the batch corresponding to $\mathbf{x}$ are still randomly distributed according to $\overline{\mathbf{p}}$. We only use M different samples to estimate each conditional mean. Denoting the true conditional mean by $\mu_{\mathbf{x}} \coloneqq \mu_{\overline{\mathbf{p}}}(\mathbf{x})$ and its estimate by $\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} \coloneqq \hat{\mu}^{\mathcal{D}}(\mathbf{x})$ for the feature $\mathbf{x} \in \mathcal{X}$, we estimate:

$$\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} = \frac{\sum_{(\mathbf{x}_i, y_i) \in \text{Batch}(\mathbf{x})} y_i \boldsymbol{\delta}_{\mathbf{x}_i = \mathbf{x}}}{1 + \sum_{(\mathbf{x}_i, y_i) \in \text{Batch}(\mathbf{x})} \boldsymbol{\delta}_{\mathbf{x}_i = \mathbf{x}}}, \tag{51}$$

where $\boldsymbol{\delta}_{\mathbf{x}_i = \mathbf{x}} = \{1 \text{ if } \mathbf{x}_i = \mathbf{x}, \text{else } 0\}$ denotes the indicator function. Depending on the number $b_{\mathbf{x}} = \sum_{(\mathbf{x}_i, y_i) \in \text{Batch}(\mathbf{x})} \boldsymbol{\delta}_{\mathbf{x}_i = \mathbf{x}}$ of samples with matching feature values that are used in the estimation of $\hat{\mu}_{\mathbf{x}}$, the estimator is slightly biased as $\mathbb{E}\left[\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} \middle| b_{\mathbf{x}} = q\right] = \frac{q\mu_{\mathbf{x}}}{q+1}$ but the bias will vanish as $b_{\mathbf{x}} \to \infty$. Note that $b_{\mathbf{x}}$ is a random variable. The variance of the estimator on iid samples is $\text{Var}_{\overline{\mathbf{p}}}\left[\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} \middle| b_{\mathbf{x}} = q\right] = \frac{q\sigma_{\mathbf{x}}^2}{(q+1)^2}$. Without loss of generality, we will suppose $\overline{p}_{\mathbf{x}} \coloneqq \overline{\mathbf{p}}(\mathbf{x}) > 0$: By Lemma 2, we obtain $\overline{\mathbf{p}}(\mathbf{x}) \geq \frac{1}{2^r}\mathbf{p}(\mathbf{x})$. Thus, $\overline{\mathbf{p}}(\mathbf{x}) = 0$ implies $\mathbf{p}(\mathbf{x}) = 0$ and the error of the estimator will not play a role in expected squared error we are interested in obtaining. By the well-known Bias-Variance decomposition, the square error of the single estimator $\mu_{\mathbf{x}}$ for a given $b_{\mathbf{x}} = q$ can be written as:

$$\mathbb{E}_{\mathcal{D} \sim \overline{\mathbf{p}}}\left[\left(\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} - \mu_{\mathbf{x}}\right)^2 \middle| b_{\mathbf{x}} = q\right] = \left(\mathbb{E}_{\overline{\mathbf{p}}}\left[\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} \middle| b_{\mathbf{x}} = q\right] - \mu_{\mathbf{x}}\right)^2 + \text{Var}_{\overline{\mathbf{p}}}\left[\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} \middle| b_{\mathbf{x}} = q\right] \tag{52}$$

$$= \left(\frac{q\mu_{\mathbf{x}}}{q+1} - \mu_{\mathbf{x}}\right)^2 + \frac{q\sigma_{\mathbf{x}}^2}{(q+1)^2} = \left(\frac{1}{q+1}\right)^2 \mu_{\mathbf{x}}^2 + \frac{q\sigma_{\mathbf{x}}^2}{(q+1)^2} \tag{53}$$

$$\leq \frac{1}{q+1}\mu_{\mathbf{x}}^2 + \frac{(q+1)\sigma_{\mathbf{x}}^2}{(q+1)^2} = \frac{1}{q+1}\left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right). \tag{54}$$

Due to the sampling procedure, the $b_{\mathbf{x}}$ are independently binomially distributed with $b_{\mathbf{x}} \sim \text{Bin}\left(M, \overline{p}_{\mathbf{x}}\right)$. Therefore, we can first aggregate the results for a single $\mathbf{x}$ and then average over the entire distribution over $\mathcal{X}$. We obtain:

$$\mathbb{E}_{\mathcal{D} \sim \overline{\mathbf{p}}}\left[\left(\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} - \mu_{\mathbf{x}}\right)^2\right] = \sum_{q=0}^{M} p(b_{\mathbf{x}} = q) \mathbb{E}_{\mathcal{D} \sim \mathbf{p}}\left[\left(\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} - \mu_{\mathbf{x}}\right)^2 \middle| b_{\mathbf{x}} = q\right] \tag{55}$$

$$\leq \sum_{q=0}^{M} \text{Bin}(q; M, \overline{p}_{\mathbf{x}}) \frac{1}{q+1}\left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right) = \left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right) \mathbb{E}_{b_{\mathbf{x}} \sim \text{Bin}(M, \overline{p}_{\mathbf{x}})}\left[\frac{1}{q+1}\right] \tag{56}$$

$$= \left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right)\left(\frac{1}{\overline{p}_{\mathbf{x}}(M+1)}\right)\left(1 - (1 - \overline{p}_{\mathbf{x}})^{M+1}\right) \tag{57}$$

$$\leq \left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right)\left(\frac{1}{\overline{p}_{\mathbf{x}}(M+1)}\right) < \left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right)\left(\frac{1}{\overline{p}_{\mathbf{x}} M}\right), \tag{58}$$

where $\text{Bin}(q; M, \overline{p}_{\mathbf{x}}) = \binom{M}{q}(\overline{p}_{\mathbf{x}})^q (1 - \overline{p}_{\mathbf{x}})^{M-q}$ is the probability given by the binomial law and the equality in Equation (57) is provided in (Cribari-Neto, Garcia, and Vasconcellos 2000, p.271). We aggregate this result to an expected value over samples from the original distribution $\mathbf{p}$. The sample $\mathbf{x} \sim \mathbf{p}$ that the estimator is evaluated on and the data set $\mathcal{D} \sim \overline{\mathbf{p}}$ are independent, and we can derive an expected error for the distribution $\mathbf{p}$ over all features $\mathbf{x}$, as:

$$\mathbb{E}_{\mathbf{x} \sim \mathbf{p}, \mathcal{D} \sim \overline{\mathbf{p}}}\left[\left(\hat{\mu}(\mathbf{x}) - \mu(\mathbf{x})\right)^2\right] = \sum_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} \mathbb{E}_{\mathcal{D} \sim \mathbf{p}}\left[\left(\hat{\mu}_{\mathbf{x}}^{\mathcal{D}} - \mu_{\mathbf{x}}\right)^2\right] \tag{59}$$

$$< \sum_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}}\left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right)\left(\frac{1}{\overline{p}_{\mathbf{x}} M}\right) \leq \sum_{\mathbf{x} \in \mathcal{X}} 2^r \overline{p}_{\mathbf{x}}\left(\mu_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}^2\right)\left(\frac{1}{\overline{p}_{\mathbf{x}} M}\right) \tag{60}$$

$$\leq \sum_{\mathbf{x} \in \mathcal{X}} \frac{2^r(\mu_{\max}^2 + \sigma_{\max}^2)}{M} \tag{61}$$

$$= |\mathcal{X}| \frac{2^r(\mu_{\max}^2 + \sigma_{\max}^2)}{M} = \frac{|\mathcal{X}|^2(2^r(\mu_{\max}^2 + \sigma_{\max}^2))}{M|\mathcal{X}|} \leq \frac{2^r|\mathcal{X}|^2(\mu_{\max}^2 + \sigma_{\max}^2)}{N - |\mathcal{X}| + 1} \tag{62}$$

$$= \frac{2^r|\mathcal{X}|^2(\sigma_{\max}^2 + \mu_{\max}^2)}{N} + \mathcal{O}\left(\frac{1}{N^2}\right), \tag{63}$$

where we use the fact that $2^r \overline{p}_{\mathbf{x}} \geq p_{\mathbf{x}}$ and the definitions of $\mu_{\max}^2, \sigma_{\max}^2$ as specified in the theorem.

□

Algorithm 1: PUC-SGD: SGD with Protected User Consent

Require: Data set $\mathcal{D}$, Loss function $\mathcal{L}$, predictor $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$
  $\mathbf{w} \leftarrow \{\text{Distribution over } \mathcal{D} \text{ with } \mathbf{w}(\mathbf{x}) \propto w(\mathbf{x})\}$
  while $r \neq 0$ **do**
    Sample batch $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(k)}, y^{(k)}) \sim \mathbf{w}$
    for $j = 1, \ldots, k$ **do** ▷ $\mathbf{x}^{(j)}=(\mathbf{b}^{(j)}, \boldsymbol{a}^{(j)}, \boldsymbol{z}^{*(j)})$
      $\mathbf{q} \leftarrow \text{Bernoulli}(0.5)$ ▷ iid. Bernoulli vector
      $\overline{\boldsymbol{a}}^{(j)} = \mathbf{q} \odot \boldsymbol{a}^{(j)}$
      $\overline{\boldsymbol{z}}_i^{*(j)} = \left\{ \boldsymbol{z}_i^{*(j)} \text{ if } \overline{\boldsymbol{a}}_i^{(j)}{=}1, \text{ else } \texttt{N/A} \right\}, i \in [r]$
      $\overline{\mathbf{x}}^{(j)} \leftarrow (\mathbf{b}^{(j)}, \overline{\boldsymbol{a}}^{(j)}, \overline{\boldsymbol{z}}^{*(j)})$
    end for
    $\boldsymbol{d\theta} \leftarrow \nabla_{\boldsymbol{\theta}} \left(\frac{1}{k} \sum_{j=1}^{k} \mathcal{L} \left(f_{\boldsymbol{\theta}}(\overline{\mathbf{x}}^{(j)}), y^{(j)} \right) \right)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \boldsymbol{d}\theta$
  end while
  return $\boldsymbol{\theta}$

D.8 Algorithms

An example of how Protected User Consent through data augmentation can be incorporated in an SGD-type algorithm is provided in Algorithm 1.

E Protected User Consent on Simulated Distributions

In this section we introduce two types of parametric data distributions with optional information that we use in our experiments with simulated data. They allow to independently control the complexity and to obtain as many samples as needed to study the convergence behavior. The first family is based on a Naive Bayes model (Appendix E.1) with binary features, whereas the second one introduced in Appendix E.2 allows for continuous features with logistic distributions.

E.1 Naïve Bayes models revisited

We can also consider a Naïve Bayes models with binary features which can possibly be unavailable as in Poole et al. (Poole, Mehr, and Wang 2020). Suppose that we have a Naive Bayes model with independent availability mechanisms, i.e., the availability of feature i is only dependent on the label y and the corresponding feature value z_i and thus $\mathbf{p}(\mathbf{b}, \boldsymbol{a}, \boldsymbol{z}, y) = (\prod_{i=1}^{n} \mathbf{p}(b_i|y)) (\prod_{i=1}^{r} \mathbf{p}(z_i|y)\mathbf{p}(a_i|z_i, y)) \mathbf{p}(y)$. A graphical representation of this model can be found in Figure 7. In this case, we can express the odds ratio as:

$$\text{odds}(Y = 1|\mathbf{b}, \boldsymbol{z}_{\mathcal{I}}^*, \boldsymbol{A}_{\mathcal{I}} = \mathbf{1}) = \frac{\mathbf{p}(Y = 1, \mathbf{b}, \boldsymbol{z}_{\mathcal{I}}, \boldsymbol{A}_{\mathcal{I}} = \mathbf{1})}{\mathbf{p}(Y = 0, \mathbf{b}, \boldsymbol{z}_{\mathcal{I}}, \boldsymbol{A}_{\mathcal{I}} = \mathbf{1})} = \tag{64}$$

$$\left(\prod_{i=1}^{n} \frac{\mathbf{p}(b_i|Y = 1)}{\mathbf{p}(b_i|Y = 0)} \right) \left(\prod_{i \in \mathcal{I}} \frac{\mathbf{p}(z_i|Y{=}1)\mathbf{p}(A_i{=}1|z_i, Y{=}1)}{\mathbf{p}(z_i|Y{=}0)\mathbf{p}(A_i{=}1|z_i, Y{=}0)} \right) \frac{\mathbf{p}(Y = 1)}{\mathbf{p}(Y = 0)}. \tag{65}$$

As we furthermore suppose the features are binary, the odds are specified through the ratios $\frac{\mathbf{p}(b_i|Y=1)}{\mathbf{p}(b_i|Y=0)}$ for $b_i \in \{0, 1\}$ and $\frac{\mathbf{p}(z_i|Y=1)}{\mathbf{p}(z_i|Y=0)} \frac{\mathbf{p}(A_i=1|z_i,Y=1)}{\mathbf{p}(A_i=1|z_i,Y=0)}$ for $z_i \in \{0, 1\}$. This requires only $2r + 2n$ parameters to be specified in total.

E.2 A parametric familiy of distributions with logistic subset models

In this section, we describe a set of conditions that can be used to construct a family of densities that will have a logistic form when applying PUC. Formally, this means that for each $\mathcal{I} \subseteq [r]$ of optional features being present, there exists a $\mathbf{w} \in \mathbb{R}^n$, $\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{I}|}$, and $s \in \mathbb{R}$ that allow to represent the odds$(Y = 1|\mathbf{b}, \boldsymbol{z}_{\mathcal{I}}^*, \boldsymbol{A}_{\mathcal{I}} = \mathbf{1})$ in the form:

$$\frac{\mathbf{p}\left(Y = 1|\mathbf{B} = \mathbf{b}, \boldsymbol{Z}_{\mathcal{I}}^* = \boldsymbol{z}_{\mathcal{I}}^*, \boldsymbol{A}_{\mathcal{I}} = \mathbf{1}\right)}{\mathbf{p}\left(Y = 0|\mathbf{B} = \mathbf{b}, \boldsymbol{Z}_{\mathcal{I}}^* = \boldsymbol{z}_{\mathcal{I}}^*, \boldsymbol{A}_{\mathcal{I}} = \mathbf{1}\right)} = \exp\left[\mathbf{w}(\mathcal{I})^\top \mathbf{b} + \boldsymbol{\beta}(\mathcal{I})^\top \boldsymbol{z}_{\mathcal{I}}^* + s(\mathcal{I})\right].$$

This allows for complex dependencies (e.g., the base feature can influence availability and value of the optional features), while also allowing to compute the ground truth PUC model relatively easy. Formally, we suggest the following assertions and show

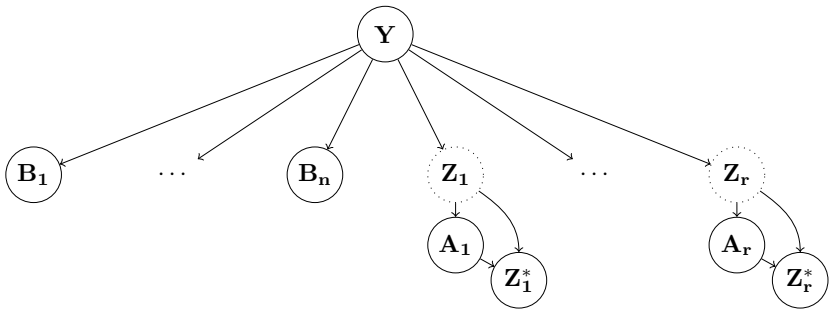


Figure 7: The Naive Bayes model with independent availability mechanisms. We observe the Label Y, the base features B_1 to B_n and the possibly unavailable features $Z_i^* = A_i \cdot Z_i$

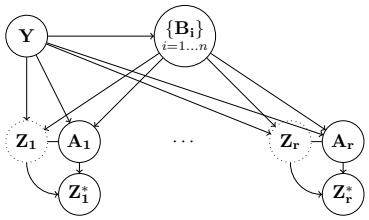


Figure 8: The relaxed graphical model with independent missingness mechanisms given the the Label Y and the base features B_1 to B_n. The observed, possibly missing features are $Z_i^* = M_i \cdot Z_i$.

that they will result in logistic models for each set of features present:

C1: Base model is logistic:	$\mathbf{p}(Y = 1\|\mathbf{b}) = \sigma(\mathbf{w}^\top \mathbf{b} + t)$
C2: Availability is cond. independent	$\forall \mathcal{I} \subseteq [r] : \mathbf{p}(\boldsymbol{A}_\mathcal{I} = \mathbf{1}\|\mathbf{b}, y) = \prod_{i \in \mathcal{I}} \mathbf{p}(A_i = 1\|\mathbf{b}, y)$
C3: Mut. independence when present:	$\forall \mathcal{I} \subseteq [r] : \mathbf{p}(\boldsymbol{z}_\mathcal{I}\|\mathbf{b}, y, \boldsymbol{A}_\mathcal{I} = \mathbf{1}) = \prod_{i \in \mathcal{I}} \mathbf{p}(z_i\|\mathbf{b}, y, A_i{=}1)$
C4: Availability is sigmoidal:	$\mathbf{p}(A_i = 1\|\mathbf{b}, Y = 1) = N(\mathbf{b})\sigma\left(\mathbf{u}_i^\top \mathbf{b} + \lambda_i\right)$
	$\mathbf{p}(A_i = 1\|\mathbf{b}, Y = 0) = N(\mathbf{b}) - \mathbf{p}(A_i = 1\|\mathbf{b}, Y = 1)$
C5: Base-dependent Normal distributions:	$\mathbf{p}(z_i\|\mathbf{b}, y, A_i = 1) \sim \mathcal{N}\left(\mathbf{v}_i^\top \mathbf{b} + \tau_i(y), \eta^2\right).$

Intuitively, after ensuring that the base feature model has a logistic form (C1), the next two assumptions follow directly from the graphical dependency model (C2, C3), see Figure 8. C4 suggests the availability should sigmoidally depend on the base features with a different offset for each class. The last condition (C5) allows the z_i to depend on the base features $\mathbf{b}$ with the same $\mathbf{v}_i$ for both classes y. However, a different offset by the coefficient τ_i can be added for each class. The entire distribution can be specified through the parameters $\mathbf{w}$, t, and $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, λ_i, $\tau_i(0)$, $\tau_i(1)$ and s_i for each missing feature in $i = 1 \dots r$.

In this special case, we can show that each of the models required will have the form of logistic regression again. We start by determining some density ratios that will arise later:

$$\frac{\mathbf{p}(A_i = 1|\mathbf{b}, Y{=}1)}{\mathbf{p}(A_i = 1|\mathbf{b}, Y{=}0)} = \frac{N(\mathbf{b})\sigma\left(\mathbf{u}_i^\top \mathbf{b} + \lambda_i\right)}{N(\mathbf{b})\left(1 - \sigma\left(\mathbf{u}_i^\top \mathbf{b} + \lambda_i\right)\right)} = \exp\left(\mathbf{u}_i^\top \mathbf{b} + \lambda_i\right), \tag{66}$$

where the identity $\frac{\sigma(x)}{1-\sigma(x)} = \exp(x)$ was used. Furthermore,

$$\frac{\mathbf{p}(z_i|\mathbf{b}, Y=1, A_i=1)}{\mathbf{p}(z_i|\mathbf{b}, Y=0, A_i=1)} = \frac{\exp\left(-\frac{\left(z_i-\mathbf{v}_i^\top\mathbf{b}-\tau_{i1}\right)^2}{2\eta^2}\right)}{\exp\left(-\frac{\left(z_i-\mathbf{v}_i^\top\mathbf{b}-\tau_{i0}\right)^2}{2\eta^2}\right)} \tag{67}$$

$$= \exp\left[-\frac{\left(z_i-\mathbf{v}_i^\top\mathbf{b}-\tau_{i1}\right)^2-\left(z_i-\mathbf{v}_i^\top\mathbf{b}-\tau_{i0}\right)^2}{2\eta^2}\right] \tag{68}$$

$$= \exp\left[\frac{\left(z_i-\mathbf{v}_i^\top\mathbf{b}\right)^2-2\left(z_i-\mathbf{v}_i^\top\mathbf{b}\right)\tau_{i1}+\tau_{i1}^2-\left(z_i-\mathbf{v}_i^\top\mathbf{b}\right)^2+2\left(z_i-\mathbf{v}_i^\top\mathbf{b}\right)\tau_{i0}-\tau_{i0}^2}{-2\eta^2}\right] \tag{69}$$

$$= \exp\left[\frac{2\left(\tau_{i1}-\tau_{i0}\right)\left(z_i-\mathbf{v}_i^\top\mathbf{b}\right)+\tau_{i0}^2-\tau_{i1}^2}{2\eta^2}\right] \tag{70}$$

$$= \exp\left[\underbrace{\eta^{-2}\left(\tau_{i1}-\tau_{i0}\right)}_{\beta_i} z_i - \underbrace{\eta^{-2}\left(\tau_{i1}-\tau_{i0}\right)\mathbf{v}_i^\top}_{\boldsymbol{\gamma}_i^\top}\mathbf{b} + \underbrace{\frac{1}{2}\eta^{-2}\left(\tau_{i0}^2-\tau_{i1}^2\right)}_{\theta_i}\right]. \tag{71}$$

Let $\mathcal{I} \subseteq [r]$ be the set index set of of present features once again. We can insert the previous results and obtain:

$$\text{odds}(Y=1|\mathbf{b}, \boldsymbol{Z}_\mathcal{I}, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1}) = \frac{\mathbf{p}(Y=1, \mathbf{b}, \boldsymbol{Z}_\mathcal{I}, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1})}{\mathbf{p}(Y=0, \mathbf{b}, \boldsymbol{Z}_\mathcal{I}, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1})} \tag{72}$$

$$= \frac{\mathbf{p}(Y=1|\mathbf{b})}{\mathbf{p}(Y=0|\mathbf{b})}\frac{\mathbf{p}(\boldsymbol{A}_\mathcal{I}=\mathbf{1}|\mathbf{b}, Y{=}1)}{\mathbf{p}(\boldsymbol{A}_\mathcal{I}=\mathbf{1}|\mathbf{b}, Y{=}0)}\frac{\mathbf{p}(\boldsymbol{Z}_\mathcal{I}|\mathbf{b}, Y{=}1, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1})}{\mathbf{p}(\boldsymbol{Z}_\mathcal{I}|\mathbf{b}, Y{=}0, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1})} \tag{73}$$

$$= \frac{\mathbf{p}(Y=1|\mathbf{b})}{\mathbf{p}(Y=0|\mathbf{b})}\left(\prod_{i\in\mathcal{I}}\frac{\mathbf{p}(z_i|\mathbf{b}, Y{=}1, A_i{=}1)\mathbf{p}(A_i{=}1|\mathbf{b}, Y{=}1)}{\mathbf{p}(z_i|\mathbf{b}, Y{=}0, A_i{=}1)\mathbf{p}(A_i{=}1|\mathbf{b}, Y{=}0)}\right) \tag{74}$$

$$= \exp(\mathbf{w}^\top\mathbf{b} + t + \sum_{i\in\mathcal{I}}\underbrace{(\mathbf{u}_i-\boldsymbol{\gamma}_i)^\top}_{\boldsymbol{\omega}_i^\top}\mathbf{b} + \beta_i z_i + \underbrace{\lambda_i+\theta_i}_{s_i}). \tag{75}$$

As this derivation shows, each subset model will again be of the logistic form. On a sidenote, the probability of a true model with no fairness constraints can be estimated as:

$$\text{odds}(Y=1|\mathbf{b}, \boldsymbol{Z}, \boldsymbol{A}) = \frac{\mathbf{p}(Y=1, \mathbf{b}, \boldsymbol{Z}_\mathcal{I}, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1}), \boldsymbol{A}_{\overline{\mathcal{I}}}{=}\mathbf{0})}{\mathbf{p}(Y=0, \mathbf{b}, \boldsymbol{Z}_\mathcal{I}, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1}, \boldsymbol{A}_{\overline{\mathcal{I}}}{=}\mathbf{0})} \tag{76}$$

$$= \frac{\mathbf{p}(Y=1|\mathbf{b})}{\mathbf{p}(Y=0|\mathbf{b})}\frac{\mathbf{p}(\boldsymbol{A}_\mathcal{I}=\mathbf{1}|\mathbf{b}, Y{=}1)}{\mathbf{p}(\boldsymbol{A}_\mathcal{I}=\mathbf{1}|\mathbf{b}, Y{=}0)}\frac{\mathbf{p}(\boldsymbol{Z}_\mathcal{I}|\mathbf{b}, Y{=}1, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1})}{\mathbf{p}(\boldsymbol{Z}_\mathcal{I}|\mathbf{b}, Y{=}0, \boldsymbol{A}_\mathcal{I}{=}\mathbf{1})}\frac{\mathbf{p}(\boldsymbol{A}_{\overline{\mathcal{I}}}{=}\mathbf{0}|\mathbf{b}, Y{=}1)}{\mathbf{p}(\boldsymbol{A}_{\overline{\mathcal{I}}}{=}\mathbf{0}|\mathbf{b}, Y{=}0)}. \tag{77}$$

F Additional Experimental Results and Details

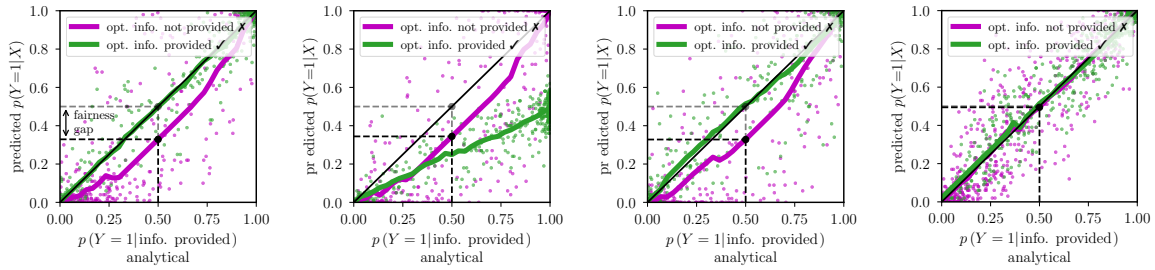
F.1 Comparing PUC to existing fairness notions

In this section, we visually show the effect of not compensating for information contained in the decision to share data. We refer Figure 9, were we compute probabilities for positive outcomes for a standard model ("fairness through unawareness") and other fairness-constrained models. The figure shows that all models apart from PUC, are not calibrated *with respect to the data explicitly provided*. The data set used to create the Figure was sampled according to the logistic family described in Appendix E.2. The feature value distributions follow a logistic form. There were two base features and one optional feature. The availability and the values of this feature was dependent on the label and the value of the base features as described in the mentioned section. Specifically, the following parameters were used to instantiate the logistic family described in Appendix E.2:

base features	n=2, $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, 5\boldsymbol{I})$, $\boldsymbol{w}{=}(-1.5, 1.0)^\top$, $t{=}0$
opt. feature 1	$\boldsymbol{u}_1 = (0.8, 0.4)^\top$, $\boldsymbol{v}_1 = (0, 1)^\top$, $\lambda_1{=}0.7$, $\tau_1(0){=}-0.25$, $\tau_1(1){=}0.25$

The models used in these experiments were `sklearn` RandomForests with default parameters. To incorporate the Fairness constraints of Statistical Parity and Equalized Odds, we leverage the `fairlearn`[4] library (Bird et al. 2020), which implements the ExponentiatedGradient algorithm by Agarwal et al. (2018). Although this algorithms only returns an approximate solution, we verified that the corresponding fairness gaps for Statistical Parity and Equalized Odds were substantially improved.

[4] https://fairlearn.org/



(a) Fairness through unawareness

(b) Statistical parity

(c) Equalized odds

(d) Protected User Consent (ours)

Figure 9: **Standard models treat users who do not share optional information not according to the data they provided**. In this work, users can provide information on optional features and only the provided information should be used in the decision making. We show calibration curves for a model without fairness considerations (a) and with common fairness constraints enforcing statistical parity (b) and equalized odds (c) with respect to a model that uses only the explicity provided information (base feature model in case of no optional information, full feature model in case of optional information). The first three models can penalize users not sharing the optional information (fairness gap in left panel), whereas a model trained with Protected User Consent through `PUCIDA` (d) exhibits no systematic bias. Models are probabilistic Random Forests trained on a synthetic data set (see Appendix F.1).

F.2 Data Sets and Preprocessing

The diabetes data set[5] was collected by the National Institute of Diabetes and Digestive and Kidney Diseases. It contains diagnostic measurements of female patients that are at least 21 years old. The target variable "Outcome" describes whether or not a person has diabetes.

The COMPAS data set[6] was originally collected by ProPublica and contains features describing criminal defendants in Broward County, Florida. It also contains their respective recividism score provided by the COMPAS algorithm and whether or not they reoffended within the following two years. For our analysis, we only kept features relevant for the prediction of recividism within the next two years and dropped irrelevant features such as name or date. Furthermore, we turned the categorical features race, sex and charge degree into numerical features by encoding the categories with integers.

The UCI adult data set[7] is one of the most poplar tabular data sets and has appeared in over 300 publications (Ding et al. 2021). The goal is to predict whether an individuals yearly income is above 50k$ (worth of 1994).

The California Housing data set[8] contains infomration and average prices of properties in certain areas in the state of California, USA. The regression target is to predict the value of a property. Because the income values range over several values of magnitudes, we apply log normalization to the label.

The ACSIncome data set ("income") is derived from US census data in the work of Ding et al. (2021). Code to download it is available online[9]. As for Adult, the goal is to predict an inviduals yearly income. It features are similar from the one used in the Adult data set, however the exact incomes of each person are reported, and the data set can therefore be used in a regression setting. Because the income values range over several values of magnitudes, we apply log normalization to the labels.

The Health Insurance ("insurance") dataset[10] contains insurance data from individuals. It is a regression dataset, where inferences about the number of hours worked are to be made (whrswk, hours worked per week). We use the experience (years of potential work experience) as optional feature in the task.

We furthermore use two datasets with natural missing features. The UCI horse colic dataset[11] ("colic") is a database of leasion surgeries on horses and contains a number of health attributes such as temperatures, pulse, respiratory rate and others. The target feature describes the outcome of the pathology. We use the feature abdominocentesis appearance as optional feature, which describes the appearance of fluid that is obtained from the abdominal cavity. This information is not available for each horse in the database and thus comes with natural missingness.

[5] https://www.kaggle.com/s/mathchi/diabetes-data-set
[6] https://www.kaggle.com/s/danofer/compass
[7] https://archive.ics.uci.edu/ml/datasets/Adult/
[8] https://www.kaggle.com/datasets/camnugent/california-housing-prices
[9] https://github.com/zykls/folktables/tree/main/folktables
[10] https://api.openml.org/d/44993
[11] https://archive.ics.uci.edu/ml/datasets/Horse+Colic

The water treatment dataset[12] contains features describing the operational state of a water treatment plant, which is to be classified as either postive or negative. We use the feature RD-DBO-P ("oxygen demand") as optional feature, which describes the Biological demand of oxygen in primary settler and comes with missing values.

Across all data sets, multi-value categorical features were one-hot encoded. We provide an overview of the characteristics of the different data sets in Table 4.

Data Sets	Label	Num. features	Num. samples (N)
diabetes	Outcome	8	768
compas	two_year_recid	9	7192
adult	ZFYA	5	21791
california housing	med_house_val	9	20640
income	income	10	19567
insurance	whrswk	11	22272
water	binaryClass	36	527
colic	pathology_cp_data	26	368

Table 4: Characteristics of the data sets studied in this work.

Stochastic Availability: We make values available by the following scheme over continuous features $z_i \in \mathcal{X}^z$:

$$\mathbf{p}(A_i = 0|z_i) = \text{sigmoid}\left(\lambda_i(z_i - \bar{z_i})\right) = \frac{1}{1 + \exp(-\lambda_i(z_i - \bar{z_i}))}, \tag{78}$$

where we denote the empirical feature mean by $\bar{z_i}$ and $\lambda_i \in \mathbb{R}$ denotes a parameter that specifies how quickly the probability of unavailability ($A_1 = 0$) increases with higher feature values (for positive values of λ_i). For negative values of λ_i, values of the feature that are lower than the mean are more likely to be unavailable. We chose λ_i such that values which were negatively influcing the prediction were more likely to be missing. We show the probabilities curves used of the feature distribution with the corresponding values of λ_i in Figure 10.

Adversarial Availability: We also experiment with adversarial sharing decision as discussed in the paper. To this end, we first train a full feature model (with no missing data) and a base feature model. We then modify the dataset and drop all optional feature values where the full feature model would lead to a lower regression score or chance of the positive outcome and retrain the corresponding classifiers on this dataset. As a final check, we replace all `PUCIDA` prediction that are higher than the base model's predictions by the base model's prediction to arrive at the aformentioned bonus system.

Models. We use standard models from the `sklearn` library (Pedregosa et al. 2011). Across all experiments, we used these models with the following parameters:

model	parameters
RandomForestClassifier / RandomForestRegressor	default parameters
ExtraTreesClassifier / ExtraTressRegressor	min_samples_split=10
GradientBoostingClassifier / GradientBoostingRegressor	min_samples_split=10
DecisionTreeClassifier / DecisionTreeRegressor	min_samples_split=10
MLPClassifier / MLPRegressor	hidden_layer_sizes= [30,40], max_iter=500

If not stated otherwise, we report averages over 5 runs with a random 80/20 test split. Code to reproduce experiments is provided in the supplementary material and will be publicly released in case of acceptance.

F.3 Experiment 1: Protecting User consent on real-world data sets

This section provides additional results for Experiment 1 (Table 2) showing that Availability Inference Restriction is violated.

Ablation studies To test the robustness of the results shown in Table 2, we performed three ablation studies. For all alternative parameters tested, the results are not qualitatively different from the original ones.

Imputation Values. In Table 2, imputed data points are replaced by zeros. Alternatively, one could also use the mean or the median of the voluntary feature as imputation values, which does not lead to substantial changes as expemplarily shown for classification datasets in Table 8. We conclude that it is hard to stop Penalization through simple imputation. Note: Our current implementation of the data augmentation strategy is implicitly converts missing values to zero for all missing values, so the results are the same as in the main paper for `PUCIDA`.

Random forest hyperparameters. In Table 2, the default parameters of random forest are used (min_samples_split=2, n_estimators=100, max_depth=None). The ablation studies with different hyperparameters are shown in Tables 9 –11.

[12] https://api.openml.org/d/940

Different models. As an alternative to random forest, we test gradient boosting models (see Table 12) and ExtRaTrees by Geurts, Ernst, and Wehenkel (2006) (see Table 13). While the extent of change differs to some extent, for every model and hyperparameter configuration, the full feature model uses the information in the sharing decision and the individuals that do not have feature values are rated worse. Occasionally, PUC models can be non-significantly better than base models, but this is due to statistical errors (as indicated through the standard deviations).

task	data	opt. feature	Base feature model	PUC	Full feature model
C	diab.	Glucose	24.75% ±1.78	20.22% ±2.01	19.90% ±2.35
C	compas	#priors	42.02% ±0.20	36.89% ±0.42	36.81% ±0.51
C	adult	edu-num	18.75% ±0.08	17.95% ±0.07	17.85% ±0.12
R	income	WKHP	63.56 ±1.08	54.66 ±0.83	56.22 ±1.13
R	calif.	m_income	12.76 ±0.11	8.51 ±0.16	8.60 ±0.17
R	insurance	experience	245.00 ±0.47	223.53 ±0.45	230.95 ±0.34

Table 5: **Performance for sharers is maintained with `PUCIDA`.** For the setup corresponding to Table 2, we montior performance measures for the subgroup of sharers. We report missclassification rate (1-Acc) for classification task and MSE ($\times 100$) for regression tasks. We show that the performance in this group is close to the unconstrained model, an indication that their optional information is used.

task	data	opt. feature	base model	imputed	PUCIDA
C	diab.	Glucose	29.30% ±0.62	25.83% ±0.41	26.95% ±0.82
C	compas	#priors	42.89% ±0.10	38.51% ±0.59	39.55% ±0.37
C	adult	edu-num	16.05% ±0.03	15.38% ±0.11	15.62% ±0.09
R	income	WKHP	85.09 ±0.12	79.76 ±0.47	81.52 ±0.28
R	calif.	m_income	13.98 ±0.06	11.90 ±0.15	12.55 ±0.05
R	insurance	experience	262.43 ±0.21	249.31 ±0.27	254.84 ±0.13

Table 6: Costs for PUC with non-adversarial sharing decisions. Otherwise the setup is equivalent to Table 4a.

task	data	opt. feature	base model	imputed	PUCIDA
Non-Adversarial Sharing					
C	diab.	Glucose	23.96 ±0.11	20.35 ±0.43	21.85 ±0.69
C	compas	#priors	40.85 ±0.06	34.88 ±0.54	36.85 ±0.21
C	adult	edu-num	11.90 ±0.03	11.18 ±0.07	11.10 ±0.05
C	water	oxygen. dem.	3.97 ±0.42	3.47 ±0.25	3.19 ±0.25
C	colic	abdom. app.	12.07 ±0.31	9.08 ±0.30	9.13 ±0.28
Adversarial Sharing					
C	diab.	Glucose	23.96 ±0.11	18.25 ±0.85	19.91 ±0.46
C	compas	#priors	40.85 ±0.06	32.03 ±0.35	34.86 ±0.26
C	adult	edu-num	11.90 ±0.03	10.26 ±0.05	10.57 ±0.03

Table 7: Costs for PUC when using $100\times$(1-ROCScores) as cost functions for the classification models instead of accuray. Setup as in Table 4a.

F.4 Experiment 2: Validating Non-Degradation and costs of fairness with respect to optional information

This section contains additional details on the experiments leading up to Table 3.

Single optional feature. We first investigate the performance of the models in the setup corresponding to Table 2, i.e., with only a single optional feature. In Table 6 we show the cost setup when non-stategic sharing decisions are taking, which leads to qualitatively equivalent results as in the main paper. Table 7 shows the results for the classification models when using the area under the 1-ROC-curve as a cost function.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	35.00% ±1.57	32.24% ±1.67	34.67% ±1.07
C	compas	#priors	44.55% ±0.45	41.84% ±0.96	44.56% ±0.51
C	adult	edu-num	13.40% ±0.13	13.02% ±0.27	13.37% ±0.20
R	income	WKHP	109.09 ±1.05	107.80 ±1.09	110.12 ±1.27
R	calif.	m_income	17.83 ±0.25	16.41 ±0.34	19.04 ±0.18
R	insurance	experience	282.99 ±0.78	278.27 ±0.46	284.88 ±0.93

(a) Corresponding to Table 1, costs, mean imputation.

task	data	optional	Base feature model	Full feature model pred.	Full feature model change	PUCIDA pred.	PUCIDA change
C	diab.	Glucose	64.17%	51.17%	-13.00% ±3.51	63.12%	-1.05% ±2.92
C	compas	#priors	51.39%	33.77%	-17.63% ±0.84	51.18%	-0.21% ±0.14
C	adult	edu-num	13.77%	11.35%	-2.42% ±0.16	13.77%	0.01% ±0.03
R	income	WKHP	100.0%	81.5%	-18.5% ±0.48	101.4%	1.4% ±0.18
R	insurance	experience	100.0%	94.9%	-5.1% ±0.10	100.1%	0.1% ±0.05
R	calif.	m_income	100.0%	95.3%	-4.7% ±0.28	104.2%	4.2% ±0.42

(b) Corresponding to Table 2, absolute predictions, mean imputation.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	35.00% ±1.57	32.22% ±1.18	34.67% ±1.07
C	compas	#priors	44.55% ±0.45	41.70% ±0.97	44.56% ±0.51
C	adult	edu-num	13.40% ±0.13	12.99% ±0.22	13.37% ±0.20
R	income	WKHP	109.09 ±1.05	107.70 ±1.18	110.12 ±1.27
R	calif.	m_income	17.83 ±0.25	16.28 ±0.31	19.04 ±0.18
R	insurance	experience	282.99 ±0.78	278.29 ±0.64	284.88 ±0.93

(c) Corresponding to Table 1, costs, median imputation.

task	data	optional	Base feature model	Full feature model pred.	Full feature model change	PUCIDA pred.	PUCIDA change
C	diab.	Glucose	64.17%	50.83%	-13.34% ±4.10	63.12%	-1.05% ±2.92
C	compas	#priors	51.39%	33.53%	-17.86% ±0.92	51.18%	-0.21% ±0.14
C	adult	edu-num	13.77%	11.29%	-2.48% ±0.17	13.77%	0.01% ±0.03
R	income	WKHP	100.0%	81.9%	-18.1% ±0.60	101.4%	1.4% ±0.18
R	insurance	experience	100.0%	94.9%	-5.1% ±0.09	100.1%	0.1% ±0.05
R	calif.	m_income	100.0%	94.9%	-5.1% ±0.52	104.2%	4.2% ±0.42

(d) Corresponding to Table 2, absolute predictions, median imputation.

Table 8: Same setup as Table 2 using *mean imputation* (upper line) and *median imputation* (lower line). The differencees between the two imputation techniques are minimal.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	34.38% ±1.60	33.52% ±1.30	34.25% ±1.01
C	compas	#priors	42.53% ±0.40	39.21% ±0.60	42.92% ±0.43
C	adult	edu-num	12.04% ±0.11	11.75% ±0.19	11.99% ±0.19
R	income	WKHP	104.62 ±0.56	103.15 ±0.80	105.85 ±0.60
R	calif.	m_income	17.84 ±0.23	16.15 ±0.50	19.11 ±0.46
R	insurance	experience	260.05 ±0.29	256.41 ±0.19	262.31 ±0.54

(a) Corresponding to Table 1 (costs).

task	data	optional	Base feature model	Full feature model pred.	Full feature model change	PUCIDA pred.	PUCIDA change
C	diab.	Glucose	64.86%	52.13%	-12.73% ±2.15	66.11%	1.25% ±1.89
C	compas	#priors	51.89%	29.92%	-21.97% ±0.97	52.11%	0.22% ±0.57
C	adult	edu-num	12.08%	9.49%	-2.59% ±0.06	12.18%	0.10% ±0.09
R	income	WKHP	100.0%	81.4%	-18.6% ±0.36	101.3%	1.3% ±0.31
R	insurance	experience	100.0%	94.8%	-5.2% ±0.06	100.2%	0.2% ±0.07
R	calif.	m_income	100.0%	94.6%	-5.4% ±1.00	104.1%	4.1% ±0.75

(b) Corresponding to Table 2 (absolute predictions).

Table 9: **Availability Inference Restriction is violated by full feature models**. Same setup as Table 2 using a *Random Forest model* with $min\_samples\_split = 10$.

Having verified these results for a single feature, we now continue with the more challenging setup of multiple optionality.

Introducing multiple optionality. For the real data experiment, we apply the following preprocessing steps to induce stochastic availability:

- We identify the most discriminative numerical features by dropping each feature from the data set and reporting the decline in predictive performance of a model trained without the feature with respect to a model trained on all features. We rank the features starting with the one resulting in the highest performance loss.
- We select the r most discriminative features, such that on average, each subset of missing pattern has at least 150 samples out of the initial data set size of N to be fitted with, i.e.,

$$r = \inf \left\{ r' \in \mathbb{N} : \frac{N}{2^r} > 150 \right\}.$$

- We do not consider numerical features were the relation to the label is not clear (i.e., is there a positive or negative correlation). The optional features are listed in Table 17.
- We independently induce stochastic availability into each feature using the sigmoidal strategy. We use a $\lambda_i = \pm \frac{1}{\sqrt{\mathrm{Var}\,[f_i]}}$, which is effectively equivalent to applying a sigmoid over normalized feature values. The signs are determined by the context such that negative indicators are more likely to be not provided and are also reported in the Table 17.

We show the corresponding results of Table 3b using 1-ROC as cost function in Table 14. We provide ablations with the two other models in Table 15 and Table 16

F.5 Experiment 3: Convergence to analytical PUC

In this section, we provide additional details regarding the experiment where we study the gaps to analytical Protected User Consent on our simulated data sets.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	36.49% ±0.45	34.27% ±0.99	37.87% ±0.95
C	compas	#priors	43.99% ±0.70	37.34% ±0.45	50.43% ±0.96
C	adult	edu-num	13.29% ±0.23	13.96% ±0.13	13.18% ±0.10
R	income	WKHP	117.24 ±0.73	115.76 ±0.79	123.08 ±1.01
R	calif.	m_income	26.08 ±0.10	20.29 ±0.21	25.51 ±0.12
R	insurance	experience	251.99 ±0.13	251.21 ±0.12	258.73 ±0.16

(a) Corresponding to Table 1 (costs).

				Full feature model		PUCIDA	
task	data	optional	Base feature model	pred.	change	pred.	change
C	diab.	Glucose	73.66%	58.45%	-15.22% ±4.23	79.07%	5.41% ±2.30
C	compas	#priors	65.82%	27.97%	-37.85% ±4.26	79.98%	14.16% ±0.75
C	adult	edu-num	2.34%	1.54%	-0.80% ±0.41	2.54%	0.20% ±0.27
R	income	WKHP	100.0%	75.5%	-24.5% ±0.78	108.6%	8.6% ±0.38
R	insurance	experience	100.0%	94.6%	-5.4% ±0.09	103.5%	3.5% ±0.04
R	calif.	m_income	100.0%	83.9%	-16.1% ±0.30	99.4%	-0.6% ±0.21

(b) Corresponding to Table 2 (absolute predictions).

Table 10: **Availability Inference Restriction is violated by full feature models**. Same setup as Table 2 using a *Random Forest model* with $max\_depth = 4$.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	34.07% ±0.87	33.61% ±0.43	33.58% ±0.92
C	compas	#priors	44.64% ±0.20	41.40% ±0.61	44.78% ±0.42
C	adult	edu-num	13.31% ±0.06	12.83% ±0.14	13.31% ±0.05
R	income	WKHP	107.98 ±0.37	106.71 ±0.32	109.29 ±0.54
R	calif.	m_income	17.70 ±0.10	16.11 ±0.31	18.93 ±0.20
R	insurance	experience	281.96 ±0.12	277.03 ±0.51	283.79 ±0.19

(a) Corresponding to Table 1 (costs).

				Full feature model		PUCIDA	
task	data	optional	Base feature model	pred.	change	pred.	change
C	diab.	Glucose	63.30%	51.63%	-11.67% ±1.20	63.21%	-0.09% ±1.14
C	compas	#priors	51.31%	32.62%	-18.69% ±1.27	51.36%	0.05% ±0.43
C	adult	edu-num	13.93%	11.47%	-2.46% ±0.24	13.91%	-0.02% ±0.09
R	income	WKHP	100.0%	81.4%	-18.6% ±0.46	101.3%	1.3% ±0.12
R	insurance	experience	100.0%	94.8%	-5.2% ±0.08	100.1%	0.1% ±0.02
R	calif.	m_income	100.0%	94.1%	-5.9% ±0.86	103.8%	3.8% ±0.48

(b) Corresponding to Table 2 (absolute predictions).

Table 11: **Availability Inference Restriction is violated by full feature models**. Same setup as Table 2 using a *Random Forest model* with $n\_estimators = 500$.

Synthetic data sets We initially conduct a synthetic data experiment to verify our theory.

First Synthetic Data Set. For the data set used in Figure 5 and Figure 4, we create binary features according to the Naive Bayes scheme described in Appendix E.1. The probabilities of each feature pointing to the corresponding class were drawn randomly, we made the three features with the highest discriminatory power optional. We then drew probabilities of the feature values being missing also at random. FOr this example, the missingness did not depend on the feature value, but only on the label. The resulting parameters are given in Table 18 for the sake of completeness.

Second Synthetic Data Set. We create a more complicated data set with continuous features as describeb by the familiy in Appendix E.2. We create two normally distributed base features and three optional features to test interesting dependency combinations by using the parameters in Table 19. This distribution includes cases where:

- the availability distribution depends on the base features ($\boldsymbol{u} \neq \mathbf{0}$, feature 1)
- the availability distribution depends on the class value ($\lambda \neq 0$, feature 1, feature 2)
- the feature value depends on the base features and the class value ($\boldsymbol{v} \neq \mathbf{0}$, $\tau(0) \neq \tau(1)$, feature 1, feature 2, feature 3)

We draw increasing numbers of samples from the known distribution and fit the corresponding estimators. The test set on which the PUC-Gap^2 is estimated on 5000 independently drawn samples. For Figure 4, we use 50000 samples to train each model.

Additional Results We also conduct the approximation experiment on the more complicated continuous data set. The results can be found in Figure 11. Note that on this continuous data sets the models will not perfectly converge. However, we show that the PUC-Gap is in range of the irreducible, random estimation error, by computing the average squared estimation error without PUC on the unfair data set and adding the ranges of this error to the plot.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	34.07% ±0.87	33.61% ±0.43	33.58% ±0.92
C	compas	#priors	44.64% ±0.20	41.40% ±0.61	44.78% ±0.42
C	adult	edu-num	13.31% ±0.06	12.83% ±0.14	13.31% ±0.05
R	income	WKHP	107.98 ±0.37	106.71 ±0.32	109.29 ±0.54
R	calif.	m_income	17.70 ±0.10	16.11 ±0.31	18.93 ±0.20
R	insurance	experience	281.96 ±0.12	277.03 ±0.51	283.79 ±0.19

(a) Corresponding to Table 1 (costs).

				Full feature model		PUCIDA	
task	data	optional	Base feature model	pred.	change	pred.	change
C	diab.	Glucose	62.80%	52.96%	-9.84% ±2.46	63.75%	0.95% ±1.35
C	compas	#priors	62.82%	21.29%	-41.54% ±1.69	64.14%	1.32% ±0.22
C	adult	edu-num	9.73%	5.98%	-3.76% ±0.10	9.31%	-0.43% ±0.07
R	income	WKHP	100.0%	79.6%	-20.4% ±0.13	100.0%	0.0% ±0.09
R	insurance	experience	100.0%	94.4%	-5.6% ±0.05	101.2%	1.2% ±0.02
R	calif.	m_income	100.0%	91.1%	-8.9% ±0.36	102.9%	2.9% ±0.75

(b) Corresponding to Table 2 (absolute predictions).

Table 12: **Availability Inference Restriction is violated by full feature models**. Same setup as Table 2 using a *Gradient Boosting model*.

C: (1-Acc)×100, R: MSE×100

task	data	opt. feature	base model	Full feature model	PUCIDA
C	diab.	Glucose	35.27% ±0.89	33.20% ±1.46	35.68% ±1.54
C	compas	#priors	42.22% ±0.41	36.69% ±0.29	42.66% ±0.56
C	adult	edu-num	11.25% ±0.09	11.37% ±0.07	11.27% ±0.12
R	income	WKHP	104.79 ±0.72	100.90 ±0.75	106.02 ±0.66
R	calif.	m_income	16.99 ±0.10	15.50 ±0.22	17.74 ±0.34
R	insurance	experience	243.33 ±0.12	242.30 ±0.12	246.22 ±0.12

(a) Corresponding to Table 1 (costs).

				Full feature model		PUCIDA	
task	data	optional	Base feature model	pred.	change	pred.	change
C	diab.	Glucose	71.27%	48.73%	-22.54% ±5.84	68.78%	-2.49% ±1.27
C	compas	#priors	53.53%	29.53%	-24.00% ±1.16	53.39%	-0.14% ±0.18
C	adult	edu-num	12.27%	9.58%	-2.69% ±0.27	12.33%	0.06% ±0.06
R	income	WKHP	100.0%	80.1%	-19.9% ±0.60	101.0%	1.0% ±0.08
R	insurance	experience	100.0%	94.6%	-5.4% ±0.07	100.1%	0.1% ±0.02
R	calif.	m_income	100.0%	90.9%	-9.1% ±0.55	104.0%	4.0% ±0.28

(b) Corresponding to Table 2 (absolute predictions).

Table 13: **Availability Inference Restriction is violated by full feature models**. Same setup as Table 2 using a *Extra Trees model*

		Fair models				Full feature model
task	data (# opt.)	Base feature model	PUCIDA (f)	PUCIDA (e)	(×)	zero-imputed
C	diab. (2)	73.14 ±2.59	77.13 ±3.67	**77.22** ±3.92	2.3	78.42 ±2.61
C	compas (5)	61.32 ±0.92	61.90 ±0.96	**62.32** ±0.86	7.6	62.69 ±1.58
C	adult (5)	84.90 ±0.46	90.39 ±0.35	**89.68** ±0.33	7.4	90.57 ±0.21

Table 14: **PUC-compliant models improve predictive performance**. Same setup as in to Table 3b, but in this case we use *ROC-AUC as the performance metric*. A higher ROC-AUC is preferable.

		Fair models				Full feature model
task	data (# opt.)	Base feature model	PUCIDA (f)	PUCIDA (e)	(×)	zero-imputed
C	diab. (2)	29.87 ±2.25	28.70 ±2.37	**28.18** ±2.74	2.3	27.14 ±2.67
C	compas (5)	40.78 ±0.63	**37.71** ±0.63	37.79 ±0.90	7.6	35.62 ±0.60
C	adult (5)	17.84 ±0.41	13.43 ±0.49	**13.43** ±0.48	7.4	13.31 ±0.45
R	calif. (4)	11.01 ±1.89	9.45 ±0.19	**9.32** ±0.34	5.1	9.04 ±0.17
R	income (3)	46.31 ±2.02	45.00 ±2.42	**44.28** ±1.86	3.4	41.89 ±1.52
R	insurance (3)	230.00 ±0.72	212.30 ±1.73	**211.27** ±2.13	3.2	210.72 ±1.55

Table 15: **PUC-compliant models improve predictive performance**. Same setup as in to Table 3b, but in this case we use *Gradient Boosted Decision Trees*.

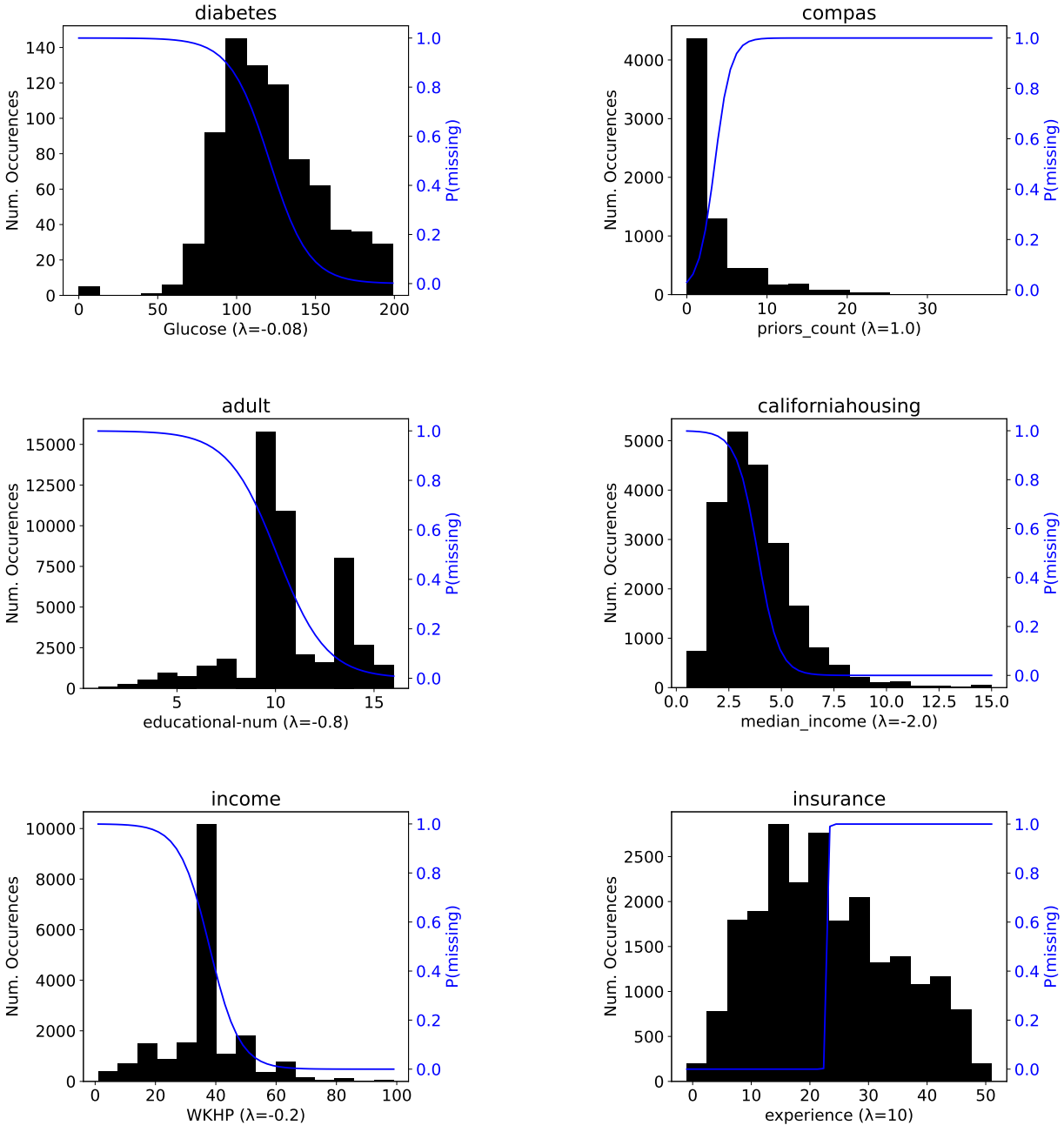


Figure 10: Value distribution of the respective optional features per data set and corresponding function $\mathbf{p}(A_i = 0|z_i)$ with parameter λ used to introduce stochastic availability.

		Fair models				Full feature model
task	data (# opt.)	Base feature model	PUCIDA (f)	PUCIDA (e)	(×)	zero-imputed
C	diab. (2)	27.79 ±4.22	28.18 ±2.48	**27.92** ±2.63	2.3	26.88 ±3.50
C	compas (5)	40.83 ±0.53	39.82 ±0.75	**39.33** ±1.38	7.7	39.74 ±1.39
C	adult (5)	18.00 ±0.37	**15.21** ±0.52	15.31 ±0.51	7.4	15.15 ±0.37
R	calif. (4)	5.83 ±0.27	9.21 ±0.64	**7.86** ±0.27	5.0	7.01 ±0.23
R	income (3)	48.99 ±1.60	47.32 ±1.87	**46.98** ±1.85	3.4	43.78 ±1.83
R	insurance (3)	251.47 ±2.57	**238.28** ±1.18	249.41 ±2.21	3.2	226.85 ±1.75

Table 16: **PUC-compliant models improve predictive performance**. Same setup as in to Table 3b, but in this case we use *Extra Trees*.

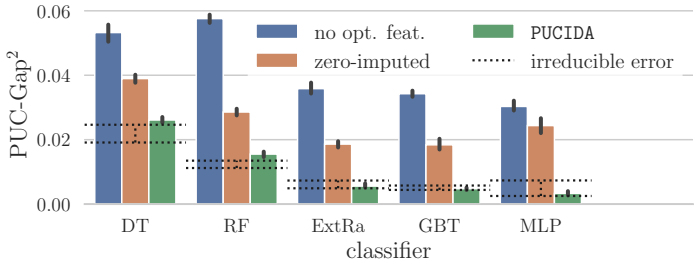


Figure 11: PUCIDA converges independently of the ML model on the second simulated data set with optional features. The fairness gaps are close to the irreducible model estimation error (due to imperfect models on this continuous data set) when applying our technique across a variety of common models on the continuous simulated data set.

data set	optional features
insurance	experience (−), kidslt6 (−), kids618 (−)
adult	age(−) , educational-num (−), hours-per-week (−), capital-gain (−), capital-loss (−)
compas	priors_count (+), age (−), c_days_from_compas (−), c_charge_degree (+), juv_misd_count (+)
diabetes	Glucose (+), age (+)
california housing	housing_median_age (+), population (−), households (−), median_income (−)
income	AGEP (−), SCHL (−), WKHP (−)

Table 17: Features made optional in the experiment with multiple optional features. Direction: (+) means higher values more likely to be unavailable, (−) indicates lower values to be more likely to be unavailable. The direction was chosen such that feature values that lead to more negative outcomes tend to be undisclosed more frequently.

feature	$p(x_i = 1\|y = 0)$	$p(x_i = 1\|y = 1)$	$p(a_i = 0\|y = 0)$	$p(a_i = 0\|y = 1)$
1	0.090	0.141	–	–
2	0.915	0.930	–	–
3	0.225	0.020	–	–
4	0.771	0.377	–	–
5	0.202	0.347	–	–
6	0.968	0.322	0.920	0.345
7	0.874	0.239	0.647	0.294
8	0.723	0.159	0.508	0.207

Table 18: Parametric distribution parameters used in the first synthetic data set. Features are all binary. Features 1–5 are base feature which are always available. Features 6 –8 are unavailable with a certain probability given the class label.

base features	n=2, $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, 5\boldsymbol{I})$, $\boldsymbol{w}$=$(-1.5, 1.0)^\top$, t=0
opt. feature 1	$\boldsymbol{u}_1 = (0.8, 0.4)^\top$, $\boldsymbol{v}_1 = (0, 1)^\top$, λ_1=0.7, $\tau_1(0)$=-0.25, $\tau_1(1)$=0.25
opt. feature 2	$\boldsymbol{u}_2 = \mathbf{0}$, $\boldsymbol{v}_2 = (0, -0.15)^\top$, λ_2=1.0, $\tau_2(0)$=0.4, $\tau_2(1)$=-0.4
opt. feature 3	$\boldsymbol{u}_3 = \mathbf{0}$, $\boldsymbol{v}_3 = (0.1, 0.2)^\top$, λ_3=0.0, $\tau_3(0)$=-0.2, $\tau_3(1)$=0.2

Table 19: Parametric distribution parameters used in the synthetic data experiment. The used density covers all possible dependencies between availability, feature values and the base features that are allowed by the graphical model.

3.5 Calibrating Privacy to Realistic Threat Models

Publication 5

Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci: Gaussian Membership Inference Privacy. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Author Contributions. Martin Pawelczyk suggested the initial idea of developing a privacy notion that allows for higher utility to Gjergji Kasneci and me. In close collaboration with all authors, I then started to develop the initial formalization to theoretically calculate Membership Inference risk of a mean operation, which was later condensed in the DP-SGD algorithm for Gaussian Membership Inference Privacy. Martin Pawelczyk suggested the Gaussian approximation, which made the hypothesis test solvable. I contributed the main part of the experiments with support from Martin Pawelczyk. Gjergji Kasneci supported the research process in all stages and contributed substantial improvements to the final manuscript.

Summary. When privacy is concerned during the model training phase, Differential Privacy (DP) has become the workhorse of privacy-preserving ML. However, recent works provide evidence that the guarantees imposed through DP are derived from an unrealistically strong attack model. Specifically, DP supposes a worst-case scenario, including an attacker who can modify the entire dataset at their will and covers edge cases such as empty datasets. Such complete access may not be realistic in practical scenarios and results in a higher loss in predictive performance than necessary. In this work, we seek a privacy notion that follows from realistic threats such as Membership Inference Attacks (MIAs) and increases possible predictive performance over DP. We define f-Membership Inference Privacy (f-MIP), which constrains the trade-off curve between false positives and false negatives for MIAs to be bounded by a function f. We then perform a novel privacy analysis for DP-SGD in terms of membership inference privacy. Our technical derivation frames MIAs as a hypothesis test and leverages a novel Gaussian approximation of the gradient distribution for large batch sizes. We denote the specialization of f-MIP using the specific μ-Gaussian trade-off function (Dong et al., 2022) for f as μ-Gaussian Membership Inference Privacy. Here, the positive real number μ controls the permissible MIA risk. Our analysis allows us to compute μ for realistic models trained with SGD. Our analysis highlights that even without noise, some base level of protection against MIAs is achieved through the averaging operation, which can be increased by performing DP-SGD, which includes cropping and adding additional noise. Usually, we

find that less noise is required to ensure GMIP than DP. We also propose the Gradient Likelihood Ratio attack (GLiR), a white-box membership inference attack that is optimal for a single SGD step.

3.5.1 Discussion

In this work, we argue that privacy should protect against the relevant threat models introduced in the related work section (Section 2.2.1). By MIA protection, we cover the most user-critical threats, such as data reconstruction in the untargeted case or with partial knowledge (e.g., reconstructing only some attributes of a given sample). We therefore believe that in many cases, protection against MIAs aligns very well with the user interest. We formally show that DP is more restrictive than needed to achieve protection against MIAs and leads to a more severe performance reduction than required. Drastically speaking, it may lead to incorrect or arbitrary decisions for some individuals (Kulynych et al., 2023), subjecting users to an additional inaccuracy risk. We propose Gaussian Membership Inference Privacy (GMIP) as a tool to address this misalignment, which defines privacy directly via the maximum allowed trade-off curve of MIAs and can be implemented with DP-SGP at lower noise levels. Our finding that some base level of protection against MIAs is already present even without adding noise to SGD, highlights that privacy protection may even be possible with no performance degradation at all for some cases.

As a word of caution, we would like to emphasize that there are practical scenarios where stronger attackers than in the MIA threat model are plausible. For instance, if the data is crawled from untrusted sources, it might very well be possible for an attacker to insert some malicious datapoints (“poisons”) into the dataset, but this is not always the case. Generally speaking, the paper highlights that privacy definitions should be derived from realistic threats faced in the applications. While DP is seen as the standard to privacy problems, we believe that there is space for alternatives to DP with relaxed threat models such as GMIP or other relaxations proposed in Triastcyn and Faltings (2020); Kaissis et al. (2023). ML security practitioners should carefully assess which threat model is realistic in their use case. This may benefit both the end user and the model owner through more accurate decisions.

Gaussian Membership Inference Privacy

Tobias Leemann[*]
University of Tübingen
Technical University of Munich

Martin Pawelczyk[*]
Harvard University

Gjergji Kasneci
Technical University of Munich

Abstract

We propose a novel and practical privacy notion called f-Membership Inference Privacy (f-MIP), which explicitly considers the capabilities of realistic adversaries under the membership inference attack threat model. Consequently, f-MIP offers interpretable privacy guarantees and improved utility (e.g., better classification accuracy). In particular, we derive a parametric family of f-MIP guarantees that we refer to as μ-Gaussian Membership Inference Privacy (μ-GMIP) by theoretically analyzing likelihood ratio-based membership inference attacks on stochastic gradient descent (SGD). Our analysis highlights that models trained with standard SGD already offer an elementary level of MIP. Additionally, we show how f-MIP can be amplified by adding noise to gradient updates. Our analysis further yields an analytical membership inference attack that offers two distinct advantages over previous approaches. First, unlike existing state-of-the-art attacks that require training hundreds of shadow models, our attack does not require *any* shadow model. Second, our analytical attack enables straightforward auditing of our privacy notion f-MIP. Finally, we quantify how various hyperparameters (e.g., batch size, number of model parameters) and specific data characteristics determine an attacker's ability to accurately infer a point's membership in the training set. We demonstrate the effectiveness of our method on models trained on vision and tabular datasets.

1 Introduction

Machine learning (ML) has seen a surge in popularity and effectiveness, leading to its widespread application across various domains. However, some of these domains, such as finance and healthcare, deal with sensitive data that cannot be publicly shared due to ethical or regulatory concerns. Therefore, ensuring data privacy becomes crucial at every stage of the ML process, including model development and deployment. In particular, the trained model itself [5, 31] or explanations computed to make the model more interpretable [29, 32] may leak information about the training data if appropriate measures are not taken. For example, this is a problem for recent generative Diffusion Models [7] and Large Language models, where the data leakage seems to be amplified by model size [6].

Differential privacy (DP) [14] is widely acknowledged as the benchmark for ensuring provable privacy in academia and industry [10]. DP utilizes randomized algorithms during training and guarantees that the output of the algorithm will not be significantly influenced by the inclusion or exclusion of any individual sample in the dataset. This provides information-theoretic protection against the maximum amount of information that an attacker can extract about any specific sample in the dataset, even when an attacker has full access to and full knowledge about the predictive model.

While DP is an appealing technique for ensuring privacy, DP's broad theoretical guarantees often come at the expense of a significant loss in utility for many ML algorithms. This utility loss cannot be further reduced by applying savvier algorithms: Recent work [26, 27] confirms that an attacker can

[*]Equal contribution. Corresponding authors: tobias.leemann@uni-tuebingen.de and martin.pawelczyk.1@gmail.com.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

be implemented whose empirical capacity to differentiate between neighboring datasets D and D' when having access to privatized models matches the theoretical upper bound. This finding suggests that to improve a model's utility, we need to take a step back and inspect the premises underlying DP. For example, previous work has shown that privacy attacks are much weaker when one imposes additional realistic restrictions on the attacker's capabilities [26].

In light of these findings, we revisit the DP threat model and identify three characteristics of an attacker that might be overly restrictive in practice. First, DP grants the attacker full control over the dataset used in training including the capacity to poison all samples in the dataset. For instance, DP's protection includes pathological cases such as an empty dataset and a dataset with a single, adversarial instance [27]. Second, in many applications, it is more likely that the attacker only has access to an API to obtain model predictions [13, 31] or to model gradients [20]. Finally, one may want to protect typical samples from the data distribution. As argued by Triastcyn & Faltings [38], it may be over-constraining to protect images of dogs in a model that is conceived and trained with images of cars.

Such more realistic attackers have been studied in the extensive literature on Membership Inference (MI) attacks (e.g., [5, 41]), where the attacker attempts to determine whether a sample from the data distribution was part of the training dataset. Under the MI threat model, Carlini et al. [5] observe that ML models with very lax ($\epsilon > 5000$) or no ($\epsilon = \infty$) DP-guarantees still provide some defense against membership inference attacks [5, 41]. Hence, we hypothesize that standard ML models trained with low or no noise injection may already offer some level of protection against realistic threats such as MI, despite resulting in very large provable DP bounds.

To build a solid groundwork for our analysis, we present a hypothesis testing interpretation of MI attacks. We then derive f-Membership Inference Privacy (f-MIP), which bounds the trade-off between an MI attacker's false positive rate (i.e., FPR, type I errors) and false negative rate (i.e., FNR, type II errors) in the hypothesis testing problem by some function f. We then analyze the privacy leakage of a gradient update step in stochastic gradient descent (SGD) and derive the first analytically optimal attack based on a likelihood ratio test. However, for f-MIP to cover practical scenarios, post-processing and composition operations need to be equipped with tractable privacy guarantees as well. Using f-MIP's handy composition properties, we analyze full model training via SGD and derive explicit f-MIP guarantees. We further extend our analysis by adding carefully calibrated noise to the SGD updates to show that f-MIP may be guaranteed without any noise or with less noise than the same parametric level of f-DP [11], leading to a smaller loss of utility.

Our analysis comes with a variety of novel insights: We confirm our hypothesis that, unlike for DP, no noise ($\tau^2 = 0$) needs to be added during SGD to guarantee f-MIP. Specifically, we prove that the trade-off curves of a single SGD step converge to the family of Gaussian trade-offs identified by Dong et al. [11] and result in the more specific μ-Gaussian Membership Inference Privacy (μ-GMIP). The main contributions this research offers to the literature on privacy preserving ML include:

1. **Interpretable and practical privacy notion**: We suggest the novel privacy notion of f-MIP that addresses the realistic threat of MI attacks. f-MIP considers the MI attacker's full trade-off curve between false positives and false negatives. Unlike competing notions, f-MIP offers appealing composition and post-processing properties.
2. **Comprehensive theoretical analysis**: We provide (tight) upper bounds on any attacker's ability to run successful MI attacks, i.e., we bound any MI attacker's ability to identify whether points belong to the training set when ML models are trained via gradient updates.
3. **Verification and auditing through novel attacks**: As a side product of our theoretical analysis, which leverages the Neyman-Pearson lemma, we propose a novel set of attacks for auditing privacy leakages. An important advantage of our analytical Gradient Likelihood-Ratio (GLiR) attack is its computational efficiency. Unlike existing attacks that rely on training hundreds of shadow models to approximate the likelihood ratio, our attack does not require any additional training steps.
4. **Privacy amplification through noise addition**: Finally, our analysis shows how one can use noisy SGD (also known as Differentially Private SGD [1]) to reach f-MIP while maintaining worst-case DP guarantees. Thereby our work establishes a theoretical connection between f-MIP and f-DP [11], which allows to translate an f-DP guarantee into an f-MIP guarantee and vice versa.

2 Related Work

Privacy notions. DP and its variants provide robust, information-theoretic privacy guarantees by ensuring that the probability distribution of an algorithm's output remains stable even when one sample of the input dataset is changed [14]. For instance, a DP algorithm is ε-DP if the probability of the algorithm outputting a particular subset E for a dataset S is not much higher than the probability of outputting E for a dataset S_0 that differs from S in only one element. DP has several appealing features, such as the ability to combine DP algorithms without sacrificing guarantees.

A few recent works have proposed to carefully relax the attacker's capabilities in order to achieve higher utility from private predictions [4, 13, 17, 38]. For example, Dwork & Feldman [13] suggest the notion of "privacy-preserving prediction" to make private model predictions through an API interface. Their work focuses on PAC learning guarantees of any class of Boolean functions. Similarly, Triastcyn & Faltings [38] suggest "Bayesian DP", which is primarily based on the definition of DP, but restricts the points in which the datasets S and S_0 may differ to those sampled from the data distribution. In contrast, Izzo et al. [17] introduces a notion based on MI attacks, where their approach guarantees that an adversary $\mathcal{A}$ does not gain a significant advantage in terms of accuracy when distinguishing whether an element $\boldsymbol{x}$ was in the training data set compared to just guessing the most likely option. However, they only constrain the accuracy of the attacker, while we argue that it is essential to bound the entire trade-off curve, particularly in the low FPR regime, to prevent certain re-identification of a few individuals [5]. Our work leverages a hypothesis testing formulation that covers the entire trade-off curve thereby offering protection also to the most vulnerable individuals. Additionally, our privacy notion maintains desirable properties such as composition and privacy amplification through subsampling, which previous notions did not consider.

Privacy Attacks on ML Models. Our work is also related to auditing privacy leakages through a common class of attacks called MI attacks. These attacks determine if a given instance is present in the training data of a particular model [5, 7, 9, 15, 23, 29, 30, 31, 32, 35, 36, 40, 41]. Compared to these works, our work suggests a new much stronger class of MI attacks that is analytically derived and uses information from model gradients. An important advantage of our analytically derived attack is its computational efficiency, as it eliminates the need to train any additional shadow models.

3 Preliminaries

The classical notion of (ε, δ)-differential privacy [14] is the current workhorse of private ML and can be described as follows: An algorithm is DP if for any two neighboring datasets S, S' (that differ by one instance) and any subset of possible outputs, the ratio of the probabilities that the algorithm's output lies in the subset for inputs S, S' is bounded by a constant factor. DP is a rigid guarantee, that covers *every* pair of datasets S and S', including pathologically crafted datasets (for instance, Nasr et al. [27] use an empty dataset) that might be unrealistic in practice. For this reason, we consider a different attack model in this work: The MI game [41]. This attack mechanism on ML models follows the goal of inferring an individual's membership in the training set of a learned ML model. We will formulate this problem using the language of hypothesis testing and trade-off functions, a concept from hypothesis testing theory [11]. We will close this section by giving several useful properties of trade-off functions which we leverage in our main theoretical results presented in Sections 4 and 5.

3.1 Membership Inference Attacks

The overarching goal of privacy-preserving machine learning lies in protecting personal data. To this end, we will show that an alternative notion of privacy can be defined through the success of a MI attack which attempts to infer whether a given instance was present in the training set or not. Following Yeom et al. [41] we define the standard MI experiment as follows:

Definition 3.1 (Membership Inference Experiment [41])**.** *Let $\mathcal{A}$ be an attacker, A be a learning algorithm, N be a positive integer, and $\mathcal{D}$ be a distribution over data points $\boldsymbol{x} \in D$, where the vector $\boldsymbol{x}$ may also be a tuple of data and labels. The MI experiment proceeds as follows: The model and data owner $\mathcal{O}$ samples $S \sim \mathcal{D}_N$ (i.e, sample n points i.i.d. from $\mathcal{D}$) and trains $A_S = A(S)$. They choose $b \in \{0, 1\}$ uniformly at random and draw $\boldsymbol{x}' \sim \mathcal{D}$ if $b = 0$, or $\boldsymbol{x}' \sim S$ if $b = 1$. Finally, the attacker is successful if $\mathcal{A}(\boldsymbol{x}', A_S, N, \mathcal{D}) = b$. $\mathcal{A}$ must output either 0 or 1.*

We note that the membership inference threat model features several key differences to the threat model underlying DP, which are listed in Table 1. Most notably, in MI attacks, the datasets are sampled from the distribution $\mathcal{D}$, whereas DP protects all datasets. This corresponds to granting the attacker the capacity of full dataset manipulation. Therefore, the MI attack threat model is sensible in cases where the attacker cannot manipulate the dataset through injection of malicious samples also called "canaries". This may be realistic for financial and healthcare applications, where the data is often collected from actual events (e.g., past trades) or only a handful of people (trusted hospital staff) have access to the records. In such scenarios, it might be overly restrictive to protect against worst-case canary attacks as attackers cannot freely inject arbitrary records into the training datasets. Furthermore, MI attacks are handy as a fundamental ingredient in crafting data extraction attacks [6]. Hence we expect a privacy notion based on the MI threat model to offer protection against a broader class of reconstruction attacks. Finally, being an established threat in the literature [5, 9, 31, 40, 41], MI can be audited through a variety of existing attacks.

	f-DP threat model	f-MIP threat model (this work)
Goal	Distinguish between S and S' for *any* S, S' that differ in at most one instance.	Distinguish whether $\boldsymbol{x}' \in S$ (training data set) or not.
Dataset access	Attacker has full data access. For example, the attacker can poison or adversarially construct datasets on which ML models could be trained; e.g., $S = \{\}$ and $S' = \{10^6\}$.	Attacker has no access to the training data set; i.e., the model owner privately trains their model free of adversarially poisoned samples.
Protected Instances	The instance in which S and S' differ is arbitrary. This includes OOD samples and extreme outliers.	The sample $\boldsymbol{x}'$ for which membership is to be inferred is drawn from the data distribution $\mathcal{D}$. Therefore, MI is concerned with typical samples that can occur in practice.
Best used	When the specific attack model is unknown. Offers a form of general protection.	When dataset access (e.g. canary injection) of an attacker can be ruled out and the main attack goal lies in revealing private training data (e.g., membership inference, data reconstruction).
Model knowledge	The attacker knows the model architecture and has full access to the model in form of its parameters, hyperparameters and its model outputs.	

Table 1: Comparing the threat models underlying f-DP and f-MIP.

3.2 Membership Inference Privacy as a Hypothesis Testing Problem

While DP has been studied through the perspective of a hypothesis testing formulation for a while [3, 11, 19, 39], we adapt this route to formulate membership inference attacks. To this end, consider the following hypothesis test:

$$\mathrm{H}_0 : \boldsymbol{x}' \notin S \text{ vs. } \mathrm{H}_1 : \boldsymbol{x}' \in S. \tag{1}$$

Rejecting the null hypothesis corresponds to detecting the presence of the individual $\boldsymbol{x}'$ in S, whereas failing to reject the null hypothesis means inferring that $\boldsymbol{x}'$ was not part of the dataset S. The formulation in (1) is a natural vehicle to think about any attacker's capabilities in detecting members of a train set in terms of false positive and true positive rates. The motivation behind these measures is that the attacker wants to reliably identify the subset of data points belonging to the training set (i.e., true positives) while incurring as few false positive errors as possible [5]. In other words, the attacker wants to maximize their true positive rate at any chosen and ideally low false positive rate (e.g., 0.001). From this perspective, the formulation in (1) allows to define membership inference privacy via trade-off functions f which exactly characterize the relation of false negative rates (i.e., 1-TPR) and false positive rates that an optimal attacker can achieve.

Definition 3.2. *(Trade-off function [11]) For any two probability distributions P and Q on the same space, denote the trade-off function Test$(P; Q) : [0; 1] \to [0; 1]$*

$$\mathrm{Test}(P; Q)(\alpha) = \inf \{\mathit{FNR} \mid \mathit{FPR} = \alpha\}, \tag{2}$$

where the infimum is taken over all (measurable) rejection rules ("tests") which lead to a FPR of α between distributions P, Q.

Not every function makes for a valid trade-off function. Instead, trade-off functions possess certain characteristics that are handy in their analysis.

Definition 3.3 (Characterization of trade-off functions [11])**.** *A function $f : [0,1] \to [0,1]$ is a trade-off function if f is convex, continuous at zero, non-increasing, and $f(r) \leq 1 - r$ for $r \in [0,1]$.*

We additionally introduce a semi-order on the space of trade-off functions to make statements on the hardness of different trade-offs in relation to each other.

Definition 3.4 (Comparing trade-offs)**.** *A trade-off function f is uniformly at least as hard as another trade-off function g, if $f(r) \geq g(r)$ for all $0 \leq r \leq 1$. We write $f \geq g$.*

If $\text{Test}(P; Q) \geq \text{Test}(P'; Q')$, testing P vs Q is uniformly at least as hard as testing P' vs Q'. Intuitively, this means that for a given FPR α, the best test possible test on $(P; Q)$ will result in an equal or higher FNR than the best test on $(P'; Q')$.

3.3 Noisy Stochastic Gradient Descent (Noisy SGD)

Most recent large-scale ML models are trained via stochastic gradient descent (SGD). Noisy SGD (also known as DP-SGD) is a variant of classical SGD that comes with privacy guarantees. We consider the algorithm as in the work by Abadi et al. [1], which we restate for convenience in Appendix A. While its characteristics with respect to DP have been extensively studied, we take a fundamentally different perspective in this work and study the capabilities of this algorithm to protect against membership inference attacks. In summary, the algorithm consists of three fundamental steps: *gradient clipping* (i.e., $\boldsymbol{\theta}_i \coloneqq \boldsymbol{g}(\boldsymbol{x}_i, y_i) \cdot \max(1, C/\|\boldsymbol{g}(\boldsymbol{x}_i, y_i)\|)$ where $\boldsymbol{g}(\boldsymbol{x}_i, y_i) = \nabla \mathcal{L}(\boldsymbol{x}_i, y_i)$ is the gradient with respect to the loss function $\mathcal{L}$), *aggregation* (i.e., $\boldsymbol{m} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\theta}_i$) and *adding Gaussian noise* (i.e., $\tilde{\boldsymbol{m}} = \boldsymbol{m} + Y$ where $Y \sim \mathcal{N}(\boldsymbol{0}, \tau^2 \boldsymbol{I})$ with variance parameter τ^2). To obtain privacy bounds for this algorithm, we study MI attacks for means of random variables. This allows us to bound the MI vulnerability of SGD.

4 Navigating Between Membership Inference Privacy and DP

In this section, we formally define our privacy notion f-MIP. To this end, it will be handy to view MI attacks as hypothesis tests.

4.1 Membership Inference Attacks from a Hypothesis Testing Perspective

Initially, we define the following distributions of the algorithm's output

$$A_0 = A(\boldsymbol{X} \cup \{\boldsymbol{x}\}) \text{ with } \boldsymbol{X} \sim \mathcal{D}^{n-1}, \boldsymbol{x} \sim \mathcal{D} \text{ and } A_1(\boldsymbol{x}') = A(\boldsymbol{X} \cup \{\boldsymbol{x}'\}) \text{ with } \boldsymbol{X} \sim \mathcal{D}^{n-1}, \quad (3)$$

where we denote other randomly sampled instances that go into the algorithm by $\boldsymbol{X} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_{n-1}\}$. Here A_0 represents the output distribution under the null hypothesis (H_0) where the sample $\boldsymbol{x}'$ is not part of the training dataset. On the other hand, A_1 is the output distribution under the alternative hypothesis (H_1) where $\boldsymbol{x}'$ was part of the training dataset. The output contains randomness due to the instances drawn from the distribution $\mathcal{D}$ and due to potential inherent randomness in A.

We observe that the distribution A_1 depends on the sample $\boldsymbol{x}'$ which is known to the attacker. The attacker will have access to samples for which A_0 and $A_1(\boldsymbol{x}')$ are simpler to distinguish and others where the distinction is harder. To reason about the characteristics of such a stochastically composed test, we define a composition operator that defines an optimal test in such a setup. To obtain a global FPR of α, an attacker can target different FPRs $\bar{\alpha}(\boldsymbol{x}')$ for each specific test. We need to consider the optimum over all possible ways of choosing $\bar{\alpha}(\boldsymbol{x}')$, which we refer to as *test-specific FPR function*, giving rise to the following definition.

Definition 4.1 (Stochastic composition of trade-off functions)**.** *Let $\mathcal{F}$ be a family of trade-off functions, $h : D \subset \mathbb{R}^d \to \mathcal{F}$ be a function that maps an instance of the data domain to a corresponding trade-off function, and $\mathcal{D}$ be a probability distribution on D. The set of valid test-specific FPR functions $\bar{\alpha} : D \to [0,1]$ that result in a global FPR of $\alpha \in [0,1]$ given the distribution $\mathcal{D}$ is defined through*

$$\mathcal{E}(\alpha, \mathcal{D}) = \{\bar{\alpha} : D \to [0,1] \mid \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{D}} [\bar{\alpha}(\boldsymbol{x}')] = \alpha\}. \quad (4)$$

For a given test-specific FPR function, $\bar{\alpha}$ the global false negative rate (type II error) β is given by

$$\beta_h(\bar{\alpha}) = \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{D}} \left[h(\boldsymbol{x})(\bar{\alpha}(\boldsymbol{x})) \right], \tag{5}$$

where $\bar{\alpha}(\boldsymbol{x})$ is the argument to the trade-off function $h(\boldsymbol{x}) \in \mathcal{F}$. For a global $\alpha \in [0, 1]$ the stochastic composition of these trade-functions is defined as

$$\left(\bigotimes_{\boldsymbol{x} \sim \mathcal{D}} h(\boldsymbol{x}) \right)(\alpha) = \min_{\bar{\alpha} \in \mathcal{E}(\alpha, \mathcal{D})} \{ \beta_h(\bar{\alpha}) \}, \tag{6}$$

(supposing the minimum exists), the smallest global false negative rate possible at a global FPR of α.

This definition specifies the trade-off function $\bigotimes_{\boldsymbol{x} \sim \mathcal{D}} h(\boldsymbol{x}) : [0, 1] \to [0, 1]$ of a stochastic composition of several trade-offs. While it is reminiscent of the "most powerful test" (MPT) [28], there are several differences to the MPT that are important in our work. Most prominently, a straightforward construction of the MPT to MI problems does not work since the adversary does not only run one hypothesis test to guess whether one sample belongs to the training data set or not; instead, the adversary draws multiple samples and runs sample-dependent and (potentially) different hypotheses tests for each drawn sample. This is necessary due to the form of the alternative hypotheses in the formulation of the test in (3), which depends on the sample $\boldsymbol{x}'$. We therefore require a tool to compose the results from different hypothesis tests. Finally, we prove that the trade-off of the stochastic composition has the properties of a trade-off function (see App. D.1):

Theorem 4.1 (Stochastic composition of trade-off functions)**.** *The stochastic composition $\bigotimes_{\boldsymbol{x} \sim \mathcal{D}} h(\boldsymbol{x})$ of trade-off functions $h(\boldsymbol{x})$ maintains the characteristics of a trade-off function, i.e., it is convex, non-increasing, $\left(\bigotimes_{\boldsymbol{x} \sim \mathcal{D}} h(\boldsymbol{x}) \right)(r) \leq 1 - r$ for all $r \in [0, 1]$, and it is continuous at zero.*

4.2 f-Membership Inference Privacy (f-MIP)

This rigorous definition of the stochastic composition operator allows us to define membership inference privacy from a hypothesis testing perspective.

Definition 4.2 (f-Membership Inference Privacy)**.** *Let f be a trade-off function. An algorithm[1] $A : D^n \to \mathbb{R}^d$ is said to be f-membership inference private (f-MIP) with respect to a data distribution $\mathcal{D}$ if*

$$\bigotimes_{\boldsymbol{x}' \sim \mathcal{D}} \text{Test}\left(A_0; A_1(\boldsymbol{x}')\right) \geq f, \tag{7}$$

where $\boldsymbol{x}' \sim \mathcal{D}$ and $\bigotimes$ denotes the stochastic composition built from individual trade-off functions of the MI hypotheses tests for random draws of $\boldsymbol{x}'$.

In this definition, both sides are functions dependent on the false positive rate α. A prominent special case of a trade-off function is the Gaussian trade-off, which stems from testing one-dimensional normal distributions of unit variance that are spaced apart by $\mu \in \mathbb{R}_{\geq 0}$. Therefore, defining the following special case of f-MIP will be useful.

Definition 4.3 (μ-Gaussian Membership Inference Privacy)**.** *Let Φ be the cumulative distribution function (CDF) of a standard normal distribution. Define $g_\mu(\alpha) := \Phi(\Phi^{-1}(1 - \alpha) - \mu)$ to be the trade-off function derived from testing two Gaussians; one with mean 0 and one with mean μ. An algorithm A is μ-Gaussian Membership Inference private (μ-GMIP) with privacy parameter μ if it is g_μ-MIP, i.e., it is MI private with trade-off function g_μ.*

Remark 4.1. *DP can also be defined via the Gaussian trade-off function, which results in μ-Gaussian Differential Privacy (μ-GDP, [11]). While the trade-off curves for both μ-GDP and μ-GMIP have the same parametric form, they have different interpretations: μ-GDP describes the trade-off function that an attacker with complete knowledge (left column in Table 1) could achieve while μ-GMIP describes the trade-off function that an attacker with MI attack capability can achieve (right column in Table 1). In the next section we will quantify their connection further.*

[1] When using the term "algorithm", we also include randomized mappings.

4.3 Relating f-MIP and f-DP

We close this section by providing first results regarding the relation between f-DP and f-MIP. As expected, f-DP is strictly stronger than f-MIP, which can be condensed in the following result:

Theorem 4.2 (f-DP implies f-MIP)**.** *Let an algorithm $A : D^n \to \mathbb{R}^d$ be f-differentially private [11]. Then, algorithm A will also be f-membership inference private.*

We proof this result in Appendix F.1. This theorem suggests one intuitive, simple and yet actionable approach to guarantee Membership Inference Privacy. This approach involves the use of DP learning algorithms such as DP-SGD [1], which train models using noised gradients. However, as we will see in the next section, using noise levels to guarantee f-DP is usually suboptimal to guarantee f-MIP.

5 Implementing f-MIP through Noisy SGD

We would now like to obtain a practical learning algorithm that comes with f-MIP guarantees. As the dependency between the final model parameters and the input data is usually hard to characterize, we follow the common approach and trace the information flow from the data to the model parameters through the training process of stochastic gradient descent [1, 33]. Since the gradient updates are the only path where information flows from the data into the model, it suffices to privatize this step.

5.1 f-MIP for One Step of Noisy SGD

We start by considering a single SGD step. Following prior work [1, 33], we make the standard assumption that only the mean over the individual gradients $\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}_i$ is used to update the model (or is published directly) where $\boldsymbol{\theta}_i \in \mathbb{R}^d$ is a sample gradient. Consistent with the definition of the membership inference game, the attacker tries to predict whether a specific gradient $\boldsymbol{\theta}'$ was part of the set $\{\boldsymbol{\theta}_i\}_i$ that was used to compute the model's mean gradient $\boldsymbol{m}$ or not. We are interested in determining the shape of the attacker's trade-off function. For the sake of conciseness, we directly consider one step of noisy SGD (i.e., one averaging operation with additional noising, see Algorithm 2 from the Appendix), which subsumes a result for the case without noise by setting $\tau^2 = 0$. We establish the following theorem using the Central Limit Theorem (CLT) for means of adequately large batches of n sample gradients, which is proven in Appendix E.

Theorem 5.1 (One-step noisy SGD is f-membership inference private)**.** *Denote the cumulative distribution function of the non-central chi-squared distribution with d degrees of freedom and non-centrality parameter γ by $F_{\chi^2_d(\gamma)}$. Let the gradients $\boldsymbol{\theta}' \in \mathbb{R}^d$ of the test points follow a distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, let $K \geq \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\theta}'\|_2^2$ and define $n_{effective} = n + \frac{\tau^2 n^2}{C^2}$. For sufficiently large batch sizes n, one step of noisy SGD is f-membership inference private with trade-off function given by:*

$$\beta(\alpha) \approx 1 - F_{\chi^2_d((n_{effective}-1)K)}\left(\frac{n_{effective}}{n_{effective}-1} F^{-1}_{\chi^2_d(n_{effective}K)}(\alpha)\right). \tag{8}$$

The larger the number of parameters d and the batch size n grow, the more the trade-off curve approaches the μ-GMIP curve, which we show next (see Figure 1).

Corollary 5.1 (One step noisy SGD is approx. μ-GMIP)**.** *For large d, n, noisy SGD is approximately $g_{\mu_{Step}}$-GMIP. In particular, $\beta(\alpha) \approx \Phi(\Phi^{-1}(1-\alpha) - \mu_{Step})$ with privacy parameter:*

$$\mu_{step} = \frac{d + (2n_{effective} - 1)K}{n_{effective}\sqrt{2d + 4n_{effective}K}}. \tag{9}$$

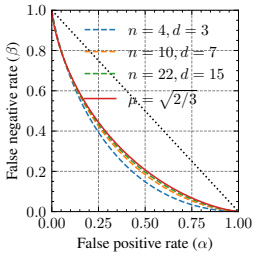


Figure 1: **Trade-off function convergence.** The trade-off function from Theorem 5.1 converges to the one from Corollary 5.1 where τ^2=0 and K=d.

This result is striking in its generality as it also covers models trained without additional noise or gradient cropping ($n_{\text{effective}} = n$ in that case). Unlike for DP, even standard models trained with non-noisy SGD offer an elementary level of MIP. Our result further explicitly quantifies four factors

that lead to attack success: the batch size n, the number of parameters d, the strength of the noise τ^2 and the worst-case data-dependent *gradient susceptibility* $\|\mathbf{\Sigma}^{-1/2}\boldsymbol{\theta}'\|_2^2$. The closeness of the trade-off function to the diagonal, which is equivalent to the attacker randomly guessing whether a gradient $\boldsymbol{\theta}'$ was part of the training data or not, is majorly determined by the ratio of d to n. The higher the value of d relative to n, the easier it becomes for the attacker to identify training data points. Furthermore, a higher gradient susceptibility K, which measures the atypicality of a gradient with respect to the gradient distribution, increases the likelihood of MI attacks succeeding in identifying training data membership. It is worth noting that if we do not restrict the gradient distribution or its support, then there might always exist gradient samples that significantly distort the mean, revealing their membership in the training dataset. This phenomenon is akin to the δ parameter in DP, which also allows exceptions for highly improbable events.

Remark 5.1 (Magnitude of μ_{Step})**.** *When the dimensions of the uncorrelated components in $\mathbf{\Sigma}^{-\frac{1}{2}}\boldsymbol{\theta}'$ are also independent, we expect K to follow a χ^2-distribution with d degrees of freedom and thus $K \in \mathcal{O}(d)$. In the standard SGD-regime ($\tau^2 = 0$) with $d, n \gg 1$, we obtain $\mu \in \mathcal{O}\left(\sqrt{d/n}\right)$.*

Remark 5.2 (On Optimality)**.** *The dependency on d when $\tau^2 > 0$ is a consequence of our intentionally broad proving strategy. Our proof approach consists of two key steps: First, we establish the optimal LRT under general gradient distributions, without adding noise or imposing any cropping constraints (See Appendix E.1). This initial step serves as the foundation for our subsequent analysis and is (1) as general as possible covering all distributions with finite variance and is (2) optimal in the sense of the Neyman-Pearson Lemma, i.e., it cannot be improved. This means that our result covers all models trained with standard SGD ($\tau^2 = 0$ and $C = \infty$) and is remarkable in its generality as it is the first to suggest clear conditions when adding noise is not required to reach f-MIP. Second, we specialize our findings to cropped random variables with added noise (See Appendix E.2). This analysis could potentially be improved by considering individual gradient dimensions independently.*

5.2 Composition and Subsampling

In the previous section, we have derived the trade-off function for a single step of SGD. Since SGD is run over multiple rounds, we require an understanding of how the individual trade-off functions can be composed when a sequence of f-MIP operations is conducted, and a random subset of the entire data distribution is used as an input for the privatized algorithm. The next lemma provides such a result for μ-GMIP and follows from a result that holds for hypotheses tests between Gaussian random variables due to Dong et al. [11] (see Appendix D.3 for details and more results).

Lemma 5.1 (Asymptotic convergence of infinite DP-SGD)**.** *Let n be the batch size in SGD, and N be the entire size of the dataset. If a single SGD-Step is at least as hard as μ_{step}-GMIP with respect to the samples that were part of the batch and $\frac{n\sqrt{t}}{N} \to c$ as $\lim_{t\to\infty}$ (the batch size is gradually decreased), then the noisy SGD algorithm will be μ-GMIP with*

$$\mu = \sqrt{2}c\sqrt{\exp(\mu_{step}^2)\Phi\left(1.5\mu_{step}\right) + 3\Phi\left(-0.5\mu_{step}\right) - 2}. \tag{10}$$

Note that this result also provides a (loose) bound for the case where exactly T iterations are run with a batch size of n' with $c = \frac{n'\sqrt{T}}{N}$ (through using $n(t) = n'$ if $t \leq T$, else $n(t) = \frac{n'\sqrt{T}}{\sqrt{t}}$). With this result in place, we can defend against MI attacks using the standard noisy SGD algorithm.

6 Experimental Evaluation

Datasets and Models. We use three datasets that were previously used in works on privacy risks of ML models [32]: The CIFAR-10 dataset which consists of 60k small images [21], the Purchase tabular classification dataset [25] and the Adult income classification dataset from the UCI machine learning repository [12]. Following prior work by Abadi et al. [1], we use a model pretrained on CIFAR-100 and finetune the last layer on CIFAR-10 using a ResNet-56 model for this task [16] where the number of fine-tuned parameters equals $d = 650$. We follow a similar strategy on the Purchase dataset, where we use a three-layer neural network. For finetuning, we use the 20 most common classes and $d = 2580$ parameters while the model is pretrained on 80 classes. On the adult dataset, we use a two-layer network with 512 random features in the first layer trained from scratch on the

dataset such that $d = 1026$. We refer to Appendix C.1 for additional training details. We release our code online.[2]

6.1 Gradient Attacks Based on the Analytical LRT

To confirm our theoretical analysis for one step of SGD and its composition, we implement the gradient attack based on the likelihood ratio test derived in the proof of Theorem 5.1. We provide a sketch of the implementation in Algorithm 1 and additional details in Appendix C.3. An essential requirement in the construction of the empirical test is the estimation of the true gradient mean $\boldsymbol{\mu}$ and the true inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ since these quantities are essential parts of both the test statistic $\hat{S}$ and the true gradient susceptibility term $\hat{K}$ needed for the analytical attack. The attacker uses their access to the gradient distribution (which is standard for membership inference attacks [5, 29] and realistic in federated learning scenarios [20]), to estimate the distribution parameters. In practice, however, the empirical estimates of $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}^{-1}$ and thus $\hat{K}$ will be noisy and therefore we do not expect that the empirical trade-off curves match the analytical curves exactly.

Algorithm 1: Gradient Likelihood Ratio (GLiR) Attack

Require: Training data distribution $\mathcal{D}$, batch size n, number of parameters d, query point $\boldsymbol{x} \in D$, averaged gradients of each batch $\boldsymbol{m}_t \in \mathbb{R}^d$ for training steps $t = 1, \ldots, T$, parameter gradient computation function $\nabla_{\boldsymbol{w}_t}\mathcal{L} : D \to \mathbb{R}^d$ of training, threshold η

$p_{\text{total}} \leftarrow 0$
for $t = 1, \ldots, T$ **do**
  $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\} \sim D^m$ ▷ Sample background data
  $\boldsymbol{g}_i = \nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{b}_i), i = 1...m$ ▷ Compute background gradients
  $\hat{\boldsymbol{\Sigma}} = \text{Cov}\{\boldsymbol{g}_1, ..., \boldsymbol{g}_m\} \in \mathbb{R}^{d\times d}$ ▷ Approximate covariance $\boldsymbol{\Sigma}$
  $\hat{\boldsymbol{\mu}} = \text{Mean}\{\boldsymbol{g}_1, ..., \boldsymbol{g}_m\} \in \mathbb{R}^d$ ▷ Approximate mean $\boldsymbol{\mu}$
  $\boldsymbol{\theta} = \nabla_{w_t}\mathcal{L}(\boldsymbol{x})$ ▷ Compute gradients for the query point
  $\hat{S} = (n-1)\,(\boldsymbol{m}_t - \boldsymbol{\theta})^\top \hat{\boldsymbol{\Sigma}}^{-1}\,(\boldsymbol{m}_t - \boldsymbol{\theta})$ ▷ Compute test statistic
  $\hat{K} = \|\hat{\boldsymbol{\Sigma}}^{-1/2}(\boldsymbol{\theta} - \hat{\boldsymbol{\mu}})\|_2^2$ ▷ Estimate gradient susceptibility
  $p_{\text{step}} = \log \text{F}^{-1}_{\chi^2_d(n\hat{K})}(\hat{S})$ ▷ Compute $\log p$-value under H_0
  $p_{\text{total}} \leftarrow p_{\text{total}} + p_{\text{step}}$
end for
return `Train` **if** $p_{\text{total}} < \eta$, **else** `Test`

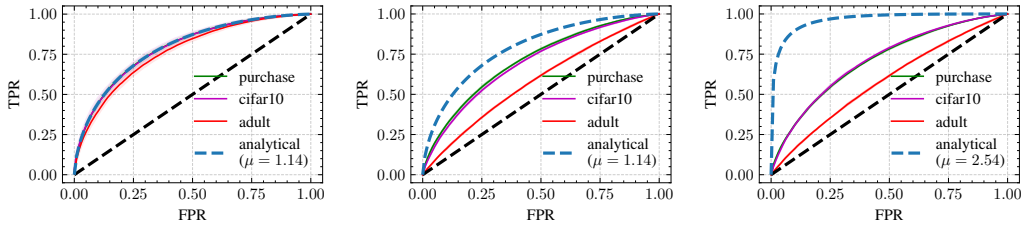
Using our novel Gradient Likelihood Ratio (GLiR) attack we can audit our derived guarantees and their utility. First, we audit our one-step guarantees from Theorem 5.1. To compare the models, we adapt the batch size n such that all models reach the same level of μ-GMIP. In Figure 2a, we use a simulated gradient distribution with known parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}^{-1}$ and d. In this case, we can estimate K accurately and observe that our bounds are tight when the distribution parameters and thus the respective gradient susceptibilities can be computed accurately. We provide additional ablation studies that gauge the approximation quality of with small values for d, n and different simulated distributions in Appendix C.2. When the parameters are unknown and we have to estimate the parameters, our attacks become weaker and do not match the analytical prediction (see Figure 2b).

We also audit our composition guarantees. We do five SGD-steps in Figure 2c. While there is a small gain in attack performance on the CIFAR-10 dataset (e.g., at FPR=0.25), the attack performance on the other datasets remains largely unaffected. This mismatch occurs since the theoretical analysis is based on the premise that the attacker gains access to independently sampled gradient means for each step to separate training and non-training points, but in practice we do not gain much new information as the model updates are not statistically independent and too incremental to change the gradient means significantly between two subsequent steps. Therefore, a practical attacker does not gain much additional information through performing several steps instead of one. Future work is required to model these dependencies and potentially arrive at a tighter composition result under incremental parameter updates. We provide results for additional existing membership inference attacks, for instance the recent loss-based likelihood-ratio attack by Carlini et al. [5] in Appendix C.4, which all show weaker success rates than the gradient-based attack that proved most powerful in our setting.

6.2 Comparing Model Utility under μ-GDP and μ-GMIP

Here we compare the utility under our privacy notion to the utility under differential privacy. We sample 20 different privacy levels ranging from $\mu \in [0.4, ..., 50]$ and calibrate the noise in the SGD iteration to reach the desired value of μ. We can do so both for μ-GMIP using the result in Equation (10) and using the result by Dong et al. [11, Corollary 4] for μ-GDP, which result in the same

[2] https://github.com/tleemann/gaussian_mip

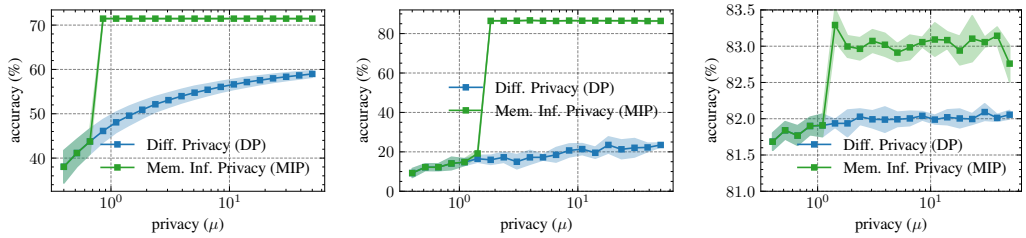


(a) Single step of simulated gradient distribution with known parameters.

(b) Single step with real model gradients and estimated parameters.

(c) As in (b), but now composition of 5 steps for real model gradients.

Figure 2: **Auditing f-MIP with our gradient attack (GLiR) when $\tau^2 = 0$.** We show trade-off curves when the gradient distribution is known (a) and when the gradients are obtained from a trained model that was finetuned on various data sets (b, c). The analytical solutions are computed with a value of $K = d$ and using the composition result for k steps in Appendix D.3 for (c).



(a) CIFAR-10

(b) Purchase

(c) Adult

Figure 3: **Utility of DP versus MIP.** Model performance on three datasets across different privacy levels μ (small μ denotes high privacy) using the notions of μ-Gaussian Differential Privacy (parametric form of f-DP, [11]) and μ-Gaussian Membership Inference Privacy (parametric form of f-MIP, ours) on three datasets. GMIP usually allows for substantially increased accuracy over the corresponding GDP guarantee with the same attack success rates controlled by μ. However, the attacker under GMIP runs membership inference (MI) attacks while GDP allows for a wider set of privacy threat models. For more details on differences in the underlying threat models see Table 1.

attack success rates while μ-GDP allows for stronger privacy threat models. Due to Theorem 4.2, we never need to add more noise for μ-GMIP than for μ-DP. Further details are provided in Appendix C.1. Figure 3 shows a comparison of the accuracy that the models obtain. We observe that the model under GMIP results in significantly higher accuracy for most values of μ. As $\mu \to 0$ both privacy notions require excessive amounts of noise such that the utility decreases towards the random guessing accuracy. On the other hand, for higher values of μ, there is no need to add any noise to the gradient to obtain μ-GMIP, allowing to obtain the full utility of the unconstrained model. This indicates that useful GMIP-bounds do not necessarily require noise. For instance, on the CIFAR-10 model, no noise is required for $\mu \geq 0.86$ which is a reasonable privacy level [11]. Overall, these results highlight that useful and interpretable privacy guarantees can often be obtained without sacrificing utility.

7 Conclusion and Future Work

In the present work, we derived the general notion of f-Membership Inference Privacy (f-MIP) by taking a hypothesis testing perspective on membership inference attacks. We then studied the noisy SGD algorithm as a model-agnostic tool to implement f-Membership Inference Privacy, while maintaining Differential Privacy (DP) as a worst-case guarantee. Our analysis revealed that significantly less noise may be required to obtain f-MIP compared to DP resulting in increased utility. Future work is required to better model the dependencies when composing subsequent SGD steps which could lead to improved bounds in practice. Furthermore, our analysis shows that when the capacity of the attacker is further restricted, e.g., to API access of predictions, there remains a gap between our theoretical bounds and loss-based membership inference attacks that can be implemented for real models. More work is required to either produce more sophisticated attacks or derive theoretical bounds for even less powerful attackers to close this gap.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016. 2, 5, 7, 8, 14, 16

[2] Andrew, G., Kairouz, P., Oh, S., Oprea, A., McMahan, H. B., and Suriyakumar, V. One-shot empirical privacy estimation for federated learning. *arXiv preprint arXiv:2302.03098*, 2023. 14

[3] Balle, B., Barthe, G., Gaboardi, M., Hsu, J., and Sato, T. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 2496–2506. PMLR, 2020. 4

[4] Bassily, R., Thakkar, O., and Guha Thakurta, A. Model-agnostic private learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. 3

[5] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. *arXiv preprint arXiv:2112.03570*, 2021. 1, 2, 3, 4, 9, 14, 15, 17, 18

[6] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021. 1, 4, 16

[7] Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023. 1, 3, 14

[8] Chen, D., Yu, N., Zhang, Y., and Fritz, M. Gan-leaks: A taxonomy of membership inference attacks against generative models. In Ligatti, J., Ou, X., Katz, J., and Vigna, G. (eds.), *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pp. 343–362. ACM, 2020. doi: 10.1145/3372297.3417238. URL https://doi.org/10.1145/3372297.3417238. 14

[9] Choquette-Choo, C. A., Tramèr, F., Carlini, N., and Papernot, N. Label-only membership inference attacks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume abs/2007.14321, 2020. 3, 4, 14

[10] Cummings, R., Desfontaines, D., Evans, D., Geambasu, R., Jagielski, M., Huang, Y., Kairouz, P., Kamath, G., Oh, S., Ohrimenko, O., et al. Challenges towards the next frontier in privacy. *arXiv preprint arXiv:2304.06929*, 2023. 1

[11] Dong, J., Roth, A., and Su, W. J. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022. 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 16, 18, 20, 21, 33, 34

[12] Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml. 8

[13] Dwork, C. and Feldman, V. Privacy-preserving prediction. In *Conference On Learning Theory*, pp. 1693–1702. PMLR, 2018. 2, 3

[14] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006. 1, 3

[15] Haim, N., Vardi, G., Yehudai, G., Shamir, O., et al. Reconstructing training data from trained neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3, 14, 16

[16] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 8

[17] Izzo, Z., Yoon, J., Arik, S. O., and Zou, J. Provable membership inference privacy. *arXiv preprint arXiv:2211.06582*, 2022. 3

[18] Jagielski, M., Ullman, J., and Oprea, A. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 22205–22216, 2020. 14

[19] Kairouz, P., Oh, S., and Viswanath, P. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, pp. 1376–1385, 2015. 4

[20] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 2, 9, 15

[21] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009. 8

[22] Lehmann, E. L., Romano, J. P., and Casella, G. *Testing statistical hypotheses*, volume 3. Springer, 2005. 24

[23] Long, Y., Bindschaedler, V., Wang, L., Bu, D., Wang, X., Tang, H., Gunter, C. A., and Chen, K. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018. 3, 14

[24] Maddock, S., Sablayrolles, A., and Stock, P. Canife: Crafting canaries for empirical privacy measurement in federated learning. In *International Conference on Learning Representations (ICLR)*, 2023. 14

[25] Nasr, M., Shokri, R., and Houmansadr, A. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 634–646, 2018. 8

[26] Nasr, M., Songi, S., Thakurta, A., Papernot, N., and Carlin, N. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pp. 866–882. IEEE, 2021. 1, 2, 14

[27] Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., and Terzis, A. Tight auditing of differentially private machine learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 1631–1648. USENIX Association, 2023. 1, 2, 3, 14

[28] Neyman, J. and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933. 6

[29] Pawelczyk, M., Lakkaraju, H., and Neel, S. On the Privacy Risks of Algorithmic Recourse. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023. 1, 3, 9, 14, 15, 17

[30] Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jegou, H. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. 3, 14

[31] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017. 1, 2, 3, 4, 14

[32] Shokri, R., Strobel, M., and Zick, Y. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 231–241, 2021. 1, 3, 8, 14, 15

[33] Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pp. 245–248. IEEE, 2013. 7, 16

[34] Steinke, T., Nasr, M., and Jagielski, M. Privacy auditing with one (1) training run. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 14

[35] Tan, J., Mason, B., Javadi, H., and Baraniuk, R. Parameters or privacy: A provable tradeoff between overparameterization and membership inference. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:17488–17500, 2022. 3, 14

[36] Tan, J., LeJeune, D., Mason, B., Javadi, H., and Baraniuk, R. G. A blessing of dimensionality in membership inference through regularization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 10968–10993. PMLR, 2023. 3, 14

[37] Thudi, A., Shumailov, I., Boenisch, F., and Papernot, N. Bounding membership inference. *arXiv preprint arXiv:2202.12232*, 2022. 14

[38] Triastcyn, A. and Faltings, B. Bayesian differential privacy for machine learning. In *International Conference on Machine Learning (ICML)*, pp. 9583–9592. PMLR, 2020. 2, 3

[39] Wasserman, L. and Zhou, S. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010. 4

[40] Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022. 3, 4, 14

[41] Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018. 2, 3, 4, 14, 15

[42] Zanella-Béguelin, S., Wutschitz, L., Tople, S., Salem, A., Rühle, V., Paverd, A., Naseri, M., Köpf, B., and Jones, D. Bayesian estimation of differential privacy. In *International Conference on Machine Learning (ICML)*. PMLR, 2023. 14

A Algorithms

Reviewing Noisy SGD. Noisy SGD, also known as DP-SGD when appropriately parameterized, is the most prevalent algorithm to train differentiable machine learning models subject to DP privacy constraints. In the main text, we have shown that this algorithm, when appropriately parameterized, can be used to train f-MIP models, too. Since our gradient attack relies on the inner workings of the algorithm, we review it here for the reader's convenience. DP-SGD works by clipping the individual gradients in each batch, taking the mean over the these gradients in a batch, and finally adding noise of magnitude τ to them. This process is then iterated over T epochs. Pseudo code is shown in Algorithm 2.

We note that the parametrization of the noise level is different across recent works. While τ corresponds to the noise added to the entire batch, other works such as [1, 11] use different parameters to characterize the noise level. For instance, Dong et al. [11] add noise of magnitude $\tau^2 = \frac{4\sigma^2 C^2}{n^2}$ and use σ to characterize the noise level.

Algorithm 2 Noisy Stochastic Gradient Descent (Noisy SGD)

Require: Training data $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, loss function $\mathcal{L}$, learning rate η, batch size n, number of iterations T, gradient norm bound $C \in \mathbb{R}_+$, noise scale $\tau \in \mathbb{R}_+$

Initialize model parameters $\boldsymbol{\theta}$ randomly
for $t = 1, \ldots, T$ **do**
  Sample a batch B_t of size n uniformly at random from D
  Compute gradients
  For each $(\boldsymbol{x}_i, y_i) \in B_t$ compute $\boldsymbol{g}(\boldsymbol{x}_i, y_i) = \nabla \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}_i, y_i)$
  Clip gradients (to have norm at most C)
  $\boldsymbol{g}(\boldsymbol{x}_i, y_i) \leftarrow \boldsymbol{g}(\boldsymbol{x}_i, y_i) \cdot \max\left(1, \frac{C}{\|\boldsymbol{g}(\boldsymbol{x}_i, y_i)\|}\right), (\boldsymbol{x}_i, y_i) \in B_t$
  Aggregate and noise gradients
  $\tilde{\boldsymbol{g}} \leftarrow \left(\frac{1}{n} \sum_i \boldsymbol{g}(\boldsymbol{x}_i, y_i)\right) + \mathcal{N}\left(\boldsymbol{0}, \tau^2 \boldsymbol{I}\right)$
  Update parameters
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \tilde{\boldsymbol{g}}$
end for
Return $\boldsymbol{\theta}$

B Additional Information on Related Work

Privacy Auditing. Our research is also linked to the literature focusing on the validation of theoretical privacy guarantees, also known as privacy auditing. This literature involves the assessment of privacy breaches in private algorithms [18]. Typically, privacy auditing entails the utilization of membership inference attacks [31, 41], wherein the effectiveness of the attack is translated into an empirical approximation of the privacy level, denoted as $\hat{\varepsilon}$. While most existing privacy auditing methods, such as those by Jagielski et al. [18], Nasr et al. [26, 27] and Zanella-Béguelin et al. [42] are computationally extensive as they require multiple shadow models to be fitted to conduct privacy audits, more recent works suggest privacy audits based on a single model fit [2, 24, 34]. Similar to this recent line of work, our privacy notion can be audited by a single model fit. However, our work differs in the sense that our auditing algorithm precisely evaluates our suggested privacy notion f-MIP.

Extended comparison to privacy attacks. There is a long line of prior work developing [5, 7, 15, 23, 29, 30, 31] or analyzing [35, 36, 37] privacy attacks on machine learning models. A common class of attacks called *membership inference attacks* focus on determining if a given instance is present in the training data of a particular model [5, 8, 9, 29, 30, 31, 32, 40, 41]. Most of these attacks typically exploit the differences in the distribution of model confidence on the true label (or the loss) between the instances that are in the training set and those that are not [5, 30, 31, 40]. For example, Shokri et al. [31] proposed a loss-based membership inference attack which determines if an instance is in the training set by testing if the loss of the model for that instance is less than a specific threshold. Other membership inference attacks are also predominantly loss-based attacks where the calibration of the threshold varies from one proposed attack to the other [5, 30, 40]. Some

Info	`Loss` [41]	`CFD` [29]	`Loss LRT` [5]	`CFD LRT` [29]	`Gradient LRT`
Query access to $f_{\hat{\theta}}$	✓	×	✓	×	×
Query access to $\nabla f_{\hat{\theta}}$	×	×	×	×	✓
Query access to $\mathcal{R}$	×	✓	×	✓	×
Known loss function	✓	×	✓	×	×
Access to $\mathcal{D}^N$	×	×	✓	✓	✓
Access to true labels	✓	×	✓	×	×
Analytical	×	×	×	×	✓
Shadow models	×	×	✓	✓	×

Table 2: Summarizing the assumptions underlying the different MI attacks. The recourse based attacks do not require access to the true labels nor do they need to know the correct loss functions, but they additionally require access to a recourse generating API $\mathcal{R}$. To the best our knowledge, our gradient attack is the only one for which analytical results exist.

works leverage different information that goes beyond the loss functions to do membership inference attacks. For instance, Shokri et al. [32] and Pawelczyk et al. [29] leverage model explanations to orchestrate membership inference attacks.

Comparison to existing attacks. In Table 2, we summarize the assumptions underlying different membership inference attacks. Note that our attack does not require the training of multiple shadow models on data from the data distribution $\mathcal{D}^N$. Instead, we derive the distributions of the LRT test statistic under the null and alternative hypotheses in closed form (see Appendix E), which drops the requirement of training (appropriately parameterized) shadow models to approximate these two distributions. These shadow models can be trained since the attacker is allowed access to the general data distribution $\mathcal{D}$. Similar to other LRT attacks, our attack also requires access to $\mathcal{D}$ to approximate the parameters $\mathbf{\Sigma}$, $\boldsymbol{\mu}$ and K required for the construction and verification of our likelihood ratio based attack. As opposed to other attacks, our attack is based on the requirement that the attacker has access to model gradients which is a realistic assumption in many federated learning scenarios [20]. Appendix C.3 summarizes our gradient based LRT attack in more detail.

Type	Dataset	# Samples (N)	# Parameters (d)	Batch size (n)	Epochs	C	τ^2	Architecture
I	CIFAR-10	500	650	500	5	10.0	0.0	ResNet56
T	Purchase	1970	2580	1970	5	10.0	0.0	3 layer DNN
T	Adult	790	1026	790	5	10.0	0.0	Random feature NN

Table 3: The parameters for the verification experiment are chosen so that the analytical privacy levels from Figure 2 are $\mu_{\text{step}} = 1.13$ and $\mu = 2.54$, respectively. Note that "I" denotes image and "T" denotes tabular.

Type	Dataset	# Samples (N)	# Parameters (d, =K)	Batch size (n)	Epochs	C	Architecture
I	CIFAR-10	48000	650	400	10	500.0	ResNet56
T	Purchase	54855	2580	795	3	2000.0	3 layer DNN
T	Adult	43000	1026	1000	20	800.0	Random feature NN

Table 4: Parameters for the utility experiment from Figure 3. Note that "I" denotes image and "T" denotes tabular. The dataset size were chosen to make them divisible by the batch size. The required noise τ^2 is determined by the required privacy level μ.

Comparing the threat models underlying f-DP and f-MIP. We note that the underlying threat model in membership inference (MI) attacks features several key differences to the threat model underlying DP, which controls an attacker's capacity to distinguish *any* two neighboring datasets D and D'.

First, in MI attacks, the datasets are sampled from the distribution $\mathcal{D}$, whereas DP protects all datasets which corresponds to granting the attacker the capacity of full dataset manipulation. Thereby the MI attack model is sensible in cases where the attacker cannot manipulate the dataset through injection of malicious samples ("canaries"). Instead, the notion of MI attacks is more realistic in cases when an attacker only has API access or access to the trained model but cannot interfere during training.

$\mu =$	0.40	0.52	0.66	0.86	1.11	1.43	1.84	2.37	3.05	3.94	5.08	6.55	8.44	10.88	14.03	18.09	23.33	30.08	38.78	50.00
CIFAR-10 (MIP)	2.84	2.44	2.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CIFAR-10 (DP)	2.84	2.44	2.13	1.89	1.70	1.55	1.42	1.32	1.24	1.17	1.11	1.06	1.02	0.98	0.94	0.91	0.88	0.85	0.83	0.81
Purchase (MIP)	4.72	4.14	3.68	3.32	3.04	2.81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Purchase (DP)	4.72	4.14	3.68	3.32	3.04	*2.81*	2.62	2.46	2.32	2.21	2.11	2.02	1.94	1.87	1.81	1.75	1.70	1.65	1.61	1.57
Adult (MIP)	3.38	2.77	2.30	1.93	1.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Adult (DP)	3.38	2.77	2.30	1.93	1.65	1.43	1.26	1.13	1.02	0.94	0.87	0.81	0.77	0.73	0.69	0.66	0.63	0.61	0.59	0.57

Table 5: Values of τ obtained for the Utility Experiment. We observe that for the higher values of μ there is often no need to add any noise to the gradients to obtain MIP, whereas substantial noise still needs to be added in the case of DP, which results in the reduced utility observed in Figure 3.

Typically membership inference attacks are a fundamental ingredient in crafting data extraction attacks [6], and hence we expect a privacy notion based on the membership inference threat model to be broadly applicable. Second, the samples that are protected under this notion are also drawn from the distribution. Consequently, MI primarily protects typical samples. In most cases, the distribution covers the data that the model is conceived to handle in practice, such that protecting against extreme outliers may be overconstraining. Finally, the goals of the attackers in both threat models are different. Instead of being able to tell apart two datasets, the MI attacker is interested in inferring whether a given sample was part of the model's training set. As the sample is already known and only a binary response is required, this goal is weaker than other types of attacks such as full reconstruction attacks [6, 15]. Therefore, the MI threat model covers many goals of realistic attackers. For the reader's convenience, we replicate the tabular overview over these key differences from Table 1 below.

C Experimental Details

C.1 Hyperparameters

In this section, we summarize the parameter settings for our experiments. In Table 3, we provide details on the verification experiment shown in Figure 2. In Table 4, we summarize the hyperparameters used in our utility experiment shown in Figure 3.

To compute the analytical privacy levels in Figure 2, we use Corollary 5.1 with $K = d$ and $\tau = 0$, resulting in $\mu_{\text{step}} = \sqrt{\frac{2d}{2n+1}} \approx 1.14$ with the batch sizes and model parameters in Table 3. For the last Figure, we use the result shown in Appendix D.3, indicating that the combined privacy level when performing k steps of SGD without subsampling and individual privacy level μ_{step}, is given by $\mu = \sqrt{k}\mu_{\text{step}} = \sqrt{5} \cdot \mu_{\text{step}} \approx 2.54$ (calculation was performed prior to rounding).

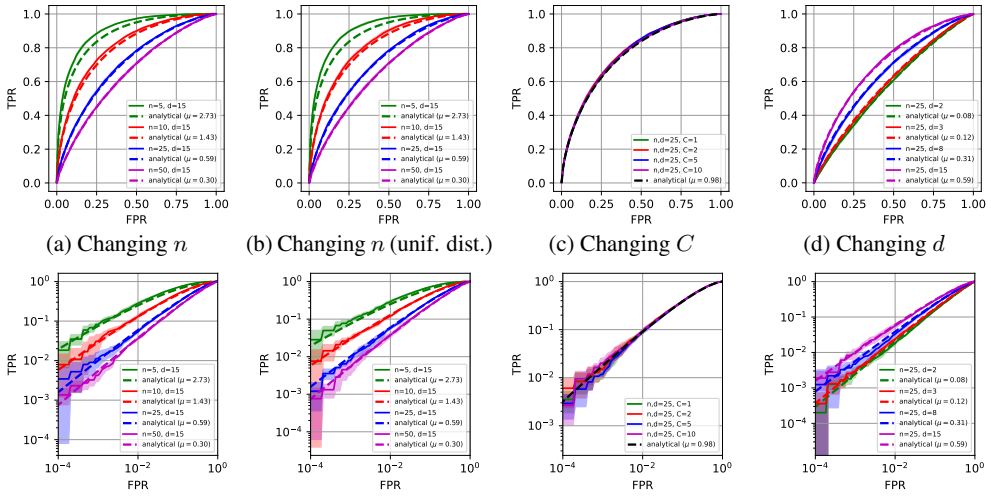
For the utility experiment, we chose the noise level τ according to Equation (10) when we use MIP. For Differential Privacy, we use the result by Dong et al. [11, Corollary 4]. However our τ has the following relation to the σ by Dong et al., $\tau = \frac{2C}{n}\sigma$ so that we plug in $\sigma = \frac{n}{2C}\tau$ in Corollary 4 of Dong et al. [11]. We solve both Equation (10) and Corollary 4 of [11] numerically to obtain the level τ_{MIP} required to obtain μ-GMIP and τ_{DP} for μ-GDP for 20 values of μ between 0.4 and 50 that are linearly spaced in logspace. Due to Theorem 4.2, we never need to add more noise for μ-GMIP than for μ-GDP. Therefore, we set $\tau_{\text{MIP}} \leftarrow \min\{\tau_{\text{MIP}}, \tau_{\text{DP}}\}$, i.e., we take the minimum of the noise levels required for GDP, GMIP when we would like to guarantee GMIP. We obtain the values given in Table 5.

C.2 Ablation studies

We show results of several ablation studies in Figure 4. The correspond to the verification setup used in Figure 2a with simulated gradients and confirm that the approximations made in our analysis even hold for very small values of $d = 2, n = 5$ or $C = 1$ and regardless of the gradients' distribution. We provide log-log scale plots for the setups corresponding to Figure 2 in Figure 5.

C.3 Gradient Likelihood Ratio (GLiR) Attack

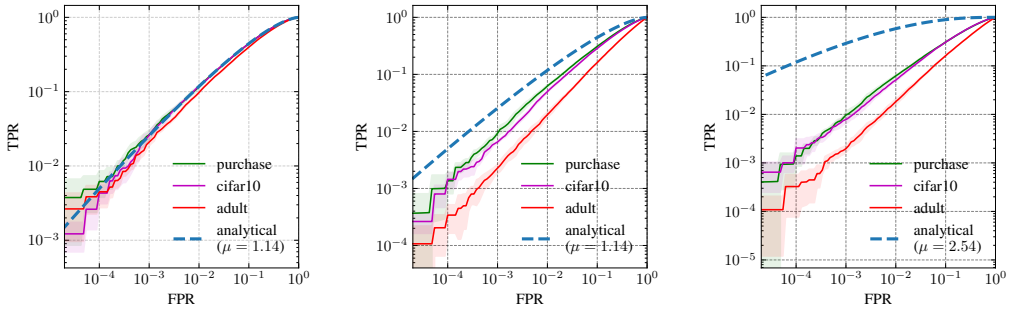
We follow the common approach and trace the information flow from the data through the training process of stochastic gradient descent [1, 33]. We follow [1, 33] and make the standard assumption that only the mean over the individual gradients $\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\theta}_i$, where $\boldsymbol{\theta}_i \in \mathbb{R}^d$ is a sample gradient is used to update the model (or is published directly). Consistent with the definition of the



(a) Changing n (b) Changing n (unif. dist.) (c) Changing C (d) Changing d

(e) Changing n (f) Changing n (unif. dist.) (g) Changing C (h) Changing d

Figure 4: **Ablation studies on approximation quality; Log-log plots in bottom row. (a, e)**: Decreasing the batch size n starts showing an effect when the batch size becomes as low as $n = 5$, which is a batch size rarely used in practice. **(b, f)**: If we change the gradient distribution to be Uniform, there is no significant difference. This is expected as the CLT also holds for means of variables with bounded support. **(c, g)**: Changing the cropping threshold C has no effect on the empirical predictions ($\mathbb{E}[\|\theta\|] = 5$ in this example) **(d, h)**: Changing the gradient dimension d only has a minor effect when $d = 2$, which is a gradient dimension unlikely to be used in practice.



(a) Single step of simulated gradient distribution with known parameters. (b) Single step with real model gradients and estimated parameters. (c) As in (b), but now composition of 5 steps for real model gradients.

Figure 5: **Observed trade-off curves for the gradient attacks when $\tau^2 = 0$, Loglog-Scale**. We show trade-off curves when the gradient distribution is known (left) and when the gradients are obtained from a trained model that was finetuned on various data sets (center, right). The analytical solutions are computed with a value of $K = d$.

membership inference game, the attacker now tries to predict whether a specific gradient $\boldsymbol{\theta}'$ was part of the set $\{\boldsymbol{\theta}_i\}_i$ that was used to compute the mean gradient $\boldsymbol{m}$ or not.

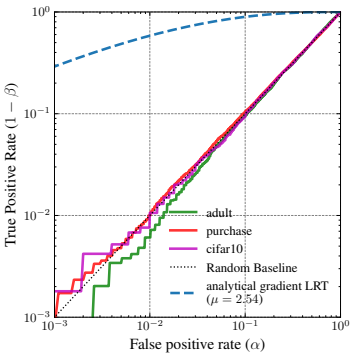
An important requirement in the construction of the gradient likelihood ratio (GLiR) attack is the estimation of the true gradient mean $\boldsymbol{\mu}$ and the true inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ since these quantities are essential parts of both the test statistic $S = (\boldsymbol{m} - \boldsymbol{\theta}')^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{m} - \boldsymbol{\theta}')$ and the true gradient susceptibility term K (see Proof of Theorem 5.1). Here, we briefly summarize the attack algorithm (see Algorithm 1 for pseudo code):

1. The attacker uses their access to the data distribution, which is standard for membership inference attacks (see e.g., [5, 29]), to obtain estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$, which we refer to as $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$ where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the true means and covariances of the gradient distributions.

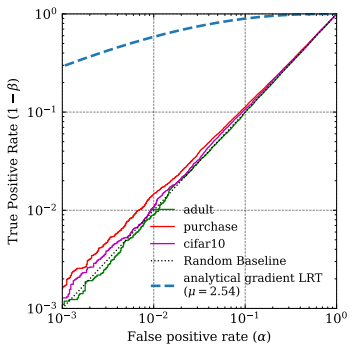
2. Given a gradient $\boldsymbol{\theta}'$, the attacker uses $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$, and estimates $\hat{K}$.
3. Given a gradient $\boldsymbol{\theta}'$, under the hypothesis that $\boldsymbol{\theta}'$ is part of the test set, the attacker uses $\hat{K}$, $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$ to computes the quantiles of the non-central chi-squared distribution and compares them to the test statistic S, resulting in p-values.
4. This procedure is repeated for several steps. The p-values can be aggregated through different means, where one strategy would be a simple multiplication of p-values as when assuming independence (corresponding to a sum of the $\log p$-values). However the threshold would have to be adjusted to compensate for multiple testing.
5. Finally, the attacker uses the p-values to determine whether a given gradient $\boldsymbol{\theta}'$ was part of the training set or not. The full trade-off curve can then be obtained by varying the thresholds over the p-values.

C.4 Additional Membership Inference attacks

We provide the same plots as in Figure 2 using a loglog-scale in Figure 5 to show that our bounds also hold for the low FPR regime. Further, we can directly compare the analytical LRT gradient based attacks with the empirical loss based LRT attacks [5]: we see that, at the false positive rate of 10^{-2}, the loss-based attacks work substantially less reliably than our proposed analytical gradient based attacks (compare Figures 5c and 6a). The loss-based LRT attacks do not work at all when the model is trained for 5 steps only while our gradient attacks work up to 10 times more reliably. In Figure 6b, we plot loss-based attacks on models that were trained for more steps than in Figure 6a. In particular, these models were trained using the same number of epochs than the models from the utility experiment of Figure 3. Now, we see that the loss-based attacks slowly start working.



(a) Loss LRT attacks on the same models as in the verification experiment from Figures 2 and 5.



(b) Loss LRT attacks on the same models as in the utility experiment from Figure 3.

Figure 6: **Observed trade-off curves for the empirical loss based LRT attacks by Carlini et al. [5] when $\tau^2 = 0$, Loglog-Scale**. We compare the analytical trade-off curves from the gradient attack to the trade-off curves obtained from the empirical loss-based LRT attacks. The analytical solutions are computed as in Figure 5.

D Proof of Theorem 4.1 and Results for General Hypothesis Test Calculus [11]

D.1 Properties of the stochastic composition operator (Theorem 4.1)

Theorem 4.1 (Stochastic composition of trade-off functions). *The stochastic composition $\bigotimes_{\boldsymbol{x}\sim\mathcal{D}} h(\boldsymbol{x})$ of trade-off functions $h(\boldsymbol{x})$ maintains the characteristics of a trade-off function, i.e., (1) it is convex, (2) non-increasing, (3) $\left(\bigotimes_{\boldsymbol{x}\sim\mathcal{D}} h(\boldsymbol{x})\right)(r) \leq 1 - r$ for all $r \in [0,1]$, and (4) it is continuous at $r = 0$.*

Proof. **(1) convexity.** We start by proving convexity (1). Let $0 \leq a \leq b \leq 1$ and let $\lambda \in [0,1]$ and define $H(r) \coloneqq \left(\bigotimes_{\boldsymbol{x}\sim\mathcal{D}} h(\boldsymbol{x})\right)(r)$ for brevity.

We have

$$H(a) = \min_{\bar{\alpha} \in \mathcal{E}(a,\mathcal{D})} \{\beta_h(\bar{\alpha})\} \tag{11}$$

and we denote the test-specific FPR function (TS-FPR) that reaches this minimum by $\bar{\alpha}_a(\boldsymbol{x}) \in \mathcal{E}(a, \mathcal{D})$ such that

$$H(a) = \beta_h(\bar{\alpha}_a) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[h(\boldsymbol{x})(\bar{\alpha}_a(\boldsymbol{x}))\right]. \tag{12}$$

We can do the same for b and find a TS-FPR function $\bar{\alpha}_b(\boldsymbol{x}) \in \mathcal{E}(b, \mathcal{D})$ such that

$$H(b) = \beta_h(\bar{\alpha}_b) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[h(\boldsymbol{x})(\bar{\alpha}_b(\boldsymbol{x}))\right]. \tag{13}$$

For any $\lambda \in [0, 1]$, we can define the convex combination of the TS-FPR functions $\bar{\alpha}_{\lambda,a,b}(\boldsymbol{x}) = \lambda\bar{\alpha}_a(\boldsymbol{x}) + (1-\lambda)\bar{\alpha}_b(\boldsymbol{x})$ and see that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\lambda\bar{\alpha}_a(\boldsymbol{x}) + (1-\lambda)\bar{\alpha}_b(\boldsymbol{x})\right] &= \lambda\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\bar{\alpha}_a(\boldsymbol{x})\right] + (1-\lambda)\left[\bar{\alpha}_b(\boldsymbol{x})\right] &(14)\\ &= \lambda a + (1-\lambda)b &(15)\end{aligned}$$

which implies that

$$\bar{\alpha}_{\lambda,a,b} \in \mathcal{E}(\lambda a + (1-\lambda)b, \mathcal{D}), \tag{16}$$

i.e., $\bar{\alpha}_{\lambda,a,b}$ is a valid TS-FPR function for a global type 1 error of $\lambda a + (1-\lambda)b$. We can now chose the function $\bar{\alpha}_{\lambda,a,b}$ to bound the minimum which allows to complete the proof

$$\begin{aligned}H(\lambda a + (1-\lambda)b) &= \min_{\bar{\alpha} \in \mathcal{E}(\lambda a+(1-\lambda)b,\mathcal{D})} \{\beta_h(\bar{\alpha})\} &(17)\\ &\leq \beta_h(\bar{\alpha}_{\lambda,a,b}) &(18)\\ &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[h(\boldsymbol{x})\left(\lambda\bar{\alpha}_a(\boldsymbol{x}) + (1-\lambda)\bar{\alpha}_b(\boldsymbol{x})\right)\right] &(19)\\ &\leq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\lambda h(\boldsymbol{x})\left(\bar{\alpha}_a(\boldsymbol{x})\right) + (1-\lambda)h(\boldsymbol{x})\left(\bar{\alpha}_b(\boldsymbol{x})\right)\right] &(20)\\ &= \lambda\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[h(\boldsymbol{x})\left(\bar{\alpha}_a(\boldsymbol{x})\right)\right] + (1-\lambda)\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[h(\boldsymbol{x})\left(\bar{\alpha}_b(\boldsymbol{x})\right)\right] &(21)\\ &= \lambda H(a) + (1-\lambda)H(b). &(22)\end{aligned}$$

In this derivation we use the convexity of the trade-off function $h(\boldsymbol{x})$ to arrive at Equation (20).

(3) upper bounded by $H(r) \leq 1 - r$. We prove property (3) next. For $r \in [0, 1]$, we have

$$H(r) = \min_{\bar{\alpha} \in \mathcal{E}(r,\mathcal{D})} \{\beta_h(\bar{\alpha})\} \tag{23}$$

where we can bound

$$\beta_h(\bar{\alpha}) = \mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[h(\boldsymbol{x})(\bar{\alpha}(\boldsymbol{x}))\right] \leq \mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[1 - \bar{\alpha}(\boldsymbol{x})\right] = 1 - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\bar{\alpha}(\boldsymbol{x})\right] = 1 - r, \tag{24}$$

where we use the fact that $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\bar{\alpha}(\boldsymbol{x})\right] = r$ for $\bar{\alpha} \in \mathcal{E}(r, \mathcal{D})$. Therefore,

$$H(r) = \min_{\bar{\alpha} \in \mathcal{E}(r,\mathcal{D})} \{\beta_h(\bar{\alpha})\} \leq 1 - r. \tag{25}$$

(2) non-increasing. Take any two points $0 \leq a < b \leq 1$. We can establish that $H(1) = 0$ by verifying that $H(r) \geq 0$ and the upper bound property (3). As $a < b \leq 1$, we can express

$$b = \lambda a + (1-\lambda)\cdot 1, \tag{26}$$

for some $\lambda \in [0, 1)$. From the convexity property of H, we infer that

$$H(b) = H(\lambda a + (1-\lambda)\cdot 1) \leq \lambda H(a) + (1-\lambda)H(1) = \lambda H(a) \leq H(a), \tag{27}$$

or $H(a) \geq H(b)$. We therefore conclude that H is non-increasing.

(4) Continuity at $r = 0$**.** We will use the common ϵ,δ-criterion of continuity. Thus, we will have to show that for every $\epsilon > 0$, there exists a $\delta > 0$ such that for all $|x - 0| < \delta$ (as the support of $H(x)$ is [0,1], this means $x < \delta$), we have $|H(\delta) - H(0)| < \epsilon$.

We first denote the value of the composed trade-off at 0 by $H_0 \coloneqq H(0)$ and $h_0(\boldsymbol{x}) \coloneqq h(\boldsymbol{x})(0)$. Denote the TS-FPR function that takes the minimum again by $\bar{\alpha}$. Because $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\bar{\alpha}\right] = 0$ together with $\bar{\alpha} \geq 0$ implies $\bar{\alpha} = 0$ almost everywhere, we can equivalently express $H_0 = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[h_0(\boldsymbol{x})\right]$.

Now let $1 \geq \epsilon > 0$ (for $\epsilon > 1$, we can choose $\delta = 1$) and let $\epsilon' \coloneqq \epsilon/4$. We first note that the individual trade-off functions $h(\boldsymbol{x})$ are increasing and continuous at $r{=}0$ themselves, which means that for every $h(\boldsymbol{x}), \epsilon_0$ there exists a $d(\boldsymbol{x}, \epsilon_0) > 0$ such that for all $y < d(\boldsymbol{x}, \epsilon_0)$ we have $h(\boldsymbol{x})(0) - h(\boldsymbol{x})(y) = h_0(\boldsymbol{x}) - h(\boldsymbol{x})(y) < \epsilon_0$. Absolute values are not required because $h(\boldsymbol{x})$ is monotonously decreasing. As $\boldsymbol{x}$ is stochastic, these d's also follow a certain distribution.

We now choose $\delta(\epsilon) = d_{\epsilon'}$ such that

$$P_{\boldsymbol{x}\sim\mathcal{D}}\left(h_0(\boldsymbol{x}) - h(\boldsymbol{x})(d_{\epsilon'}/\epsilon') \geq \epsilon'\right) < \epsilon'. \tag{28}$$

This means that the probability that the change in the trade-off function when going from 0 to $d_{\epsilon'}/\epsilon'$ will exceed ϵ', is bounded by ϵ'. We can find such a $d_{\epsilon'} > 0$ for every $\epsilon' > 0$ by conducting the following steps: Finding a $d(\boldsymbol{x}, \epsilon')$ for each $\boldsymbol{x}$. This is possible due to the continuity of the individual trade-offs. However, the values of $d(\boldsymbol{x}, \epsilon')$ can grow arbitrarily large or small for some $\boldsymbol{x}$. Therefore, we select the $1-\epsilon'$-quantile $q_{1-\epsilon'}$ of the distribution of $d(\boldsymbol{x}, \epsilon')$ for $\boldsymbol{x} \sim \mathcal{D}$ for which we certainly have $0 < q_{1-\epsilon'} < 1$. We then choose $d_{\epsilon'} = q_{1-\epsilon'}\epsilon' > 0$. Having found such value $d_{\epsilon'}$ with the characteristic in Equation (28) implies

$$\mathbb{E}_{\boldsymbol{x}}\left[H_0 - h(\boldsymbol{x})(d_{\epsilon'}/\epsilon')\right] = \mathbb{E}_{\boldsymbol{x}}\left[h_0(\boldsymbol{x}) - h(\boldsymbol{x})(d_{\epsilon'}/\epsilon')\right] < 2\epsilon', \tag{29}$$

as with a probability smaller than ϵ', the change can only be bounded by 1, for the other values, it is bounded by ϵ'. We now bound $H_0 - H(y)$ for $y < d_{\epsilon'}$ to show that $H_0 - H(y) < \epsilon$. First we note that to obtain a global true positive rate of $d_{\epsilon'}$, $P(\bar{\alpha} > d_{\epsilon'}/\epsilon') \leq \epsilon'$:

$$\begin{aligned}
H_0 - H(y) \leq & H_0 - H(d_{\epsilon'}) && (30)\\
\leq & P(\bar{\alpha} \leq d_{\epsilon'}/\epsilon)\mathbb{E}_{\boldsymbol{x}}[h_0(\boldsymbol{x}) - h(\boldsymbol{x})(\bar{\alpha}(\boldsymbol{x}))|\bar{\alpha} \leq d_{\epsilon'}/\epsilon] + && (31)\\
& P(\bar{\alpha} > d_{\epsilon'}/\epsilon)\mathbb{E}_{\boldsymbol{x}}[h_0(\boldsymbol{x}) - h(\boldsymbol{x})(\bar{\alpha}(\boldsymbol{x}))|\bar{\alpha} > d_{\epsilon'}/\epsilon] && (32)\\
\leq & 1 \cdot \mathbb{E}_{\boldsymbol{x}}[H_0 - h(\boldsymbol{x})(\bar{\alpha}(\boldsymbol{x}))|\bar{\alpha} \leq d_{\epsilon'}/\epsilon] + \epsilon' && (33)
\end{aligned}$$

We also note that:

$$\begin{aligned}
& \mathbb{E}_{\boldsymbol{x}}[h_0(\boldsymbol{x}) - h(\boldsymbol{x})(d_{\epsilon'}/\epsilon)|\bar{\alpha} \leq d_{\epsilon'}/\epsilon] && (34)\\
= & \frac{\mathbb{E}_{\boldsymbol{x}}[h_0(\boldsymbol{x}) - h(\boldsymbol{x})(d_{\epsilon'}/\epsilon)] - P(\bar{\alpha} > d_{\epsilon'}/\epsilon)\mathbb{E}_{\boldsymbol{x}}[h_0(\boldsymbol{x}) - h(\boldsymbol{x})(d_{\epsilon'}/\epsilon)|\bar{\alpha} \leq d_{\epsilon'}/\epsilon]}{P(\bar{\alpha} \leq d_{\epsilon'}/\epsilon)} && (35)\\
< & \frac{2\epsilon'}{1-\epsilon'} < 2\epsilon'\frac{4}{3} < 3\epsilon'. && (36)
\end{aligned}$$

In total we arrive at $H(0) - H(y) \leq H(0) - H(d_{\epsilon'}) < 3\epsilon' + \epsilon' = 4\epsilon' = \epsilon$. □

D.2 A composition lemma for individual tests

We repeat a lemma from Dong et al. [11].

Definition D.1. *The tensor product of two trade-off functions $f = \mathit{Test}(P; Q)$ and $g = \mathit{Test}(P'; Q')$ where P, P', Q, Q' are distributions is defined as*

$$f \otimes g \coloneqq \mathit{Test}(P \times P'; Q \times Q'). \tag{37}$$

Thus, the trade-off function f of a test that is composed of independent dimension-wise tests with trade-off functions $f_1, \ldots, f_n$ can be written as $f = f_1 \otimes f_2 \otimes \ldots \otimes f_n$. We reiterate the following result:

Lemma D.1. *If there are two trade-off functions $f_1 \geq f_2$, i.e., the test f_1 is uniformly at least as hard as f_2, for any other trade-off function g:*

$$f_1 \otimes g \geq f_2 \otimes g. \tag{38}$$

Thus, by making an individual test harder, the functional composition (the tensor product, see below) will also be uniformly harder or maintain its hardness. This lemma corresponds to Lemma C.2. of Dong et al. [11, Appendix C] where the corresponding proof can be found.

D.3 Composition results for f-MIP derived from results for f-DP

The result given in Lemma 5.1 follows from Theorem 11 and Lemma 3 in Dong et al. [11], which provide functional composition results (we refer to the subsequent execution of two algorithms as *functional* composition and the to the stochastic selection of a test as in Definition 4.1 as *stochastic* composition) for general hypotheses tests. In particular, the following result [11, Theorem 4] can be restated:

Theorem D.1. *Let $A_i : D \times D_1 \times \ldots \times D_{i-1} \to D_i$ be a series of f_i-DP algorithms ("mechansisms" in [11]) for all inputs $x \in D, y_1 \in D_1, \ldots, y_{i-1} \in D_{i-1}$, for $i = 1, \ldots, r$. Then the r-fold composed mechanism $M : D \to D_1 \times \ldots \times D_r$, defined as $M = (A_1(x), A_2(x, A_1(x)), \ldots, A_r(x, ...))$ is $f_1 \otimes \ldots \otimes f_r$-DP.*

This and other composition results from Dong et al. also apply to the MIP bounds derived for our analysis of DP-SGD and for a stricter from of f-MIP, where additionally each Test$(A_0, A_1(\boldsymbol{x}')) > f$ (in Definition 4.2) is bounded by f for each $\boldsymbol{x}'$. We can then replace f-DP with f-MIP in the results.

Proof Scheme. This can be seen as follows: Suppose we have several steps $i = 1, \ldots, k$ each being naturally f_i-MIP as each of the tests in Definition 4.2 is bounded by f_i for each $\boldsymbol{x}'$. We can apply the functional composition results from Dong et al. for each $\boldsymbol{x}'$ independently, as they hold for hypotheses tests in general. We therefore bound the functional composition for each individual $\boldsymbol{x}'$ through the composition result FuncComp$(f_1, \ldots, f_k)$. When we finally perform the stochastic composition to obtain f-MIP, we use the result stated below (Theorem D.2) which tells us that if each individual test in the stochastic composition is bounded through some trade-off, this in an upper bound on the entire stochastic composition as well. In our case each individual test is bounded by FuncComp$(f_1, \ldots, f_k)$, which will thus bound the stochastic composition result and the level of f-MIP as well.

In our analysis of SGD, such a worst-case exists and boils down to the test for the highest value of K considered (as this results in the highest μ). Using these insights, the results by Dong et al. [11] can generally be transferred without further ramifications. Instead of using a value of $\frac{1}{\sigma}$ for the privacy level of each step, we plug in μ_{step} and arrive at Lemma 5.1.

Example: Composing μ-GMIP algorithms. As an alternative example for a composition results, we note that Dong et al. [11, Corollary 2] provide a result which explicitly states that the r-fold composition of steps which are μ_i-GDP each is $\sqrt{\mu_1 + \ldots + \mu_r}$-GDP. This result also holds for μ-GMIP as well, if the stochastic composition operator can again be bounded by the trade-off g_{μ_i} for every $\boldsymbol{x}'$, such as in the SGD case we study in this work. We use this result to arrive at the privacy level shown in Figure 2c, where we conduct k steps of SGD, resulting in a combined privacy level of $\mu = \sqrt{k}\mu_{\text{step}}$.

We now consider the concluding remark. We first note that it is possible to make the functional composition harder by using smaller batch sizes than allowed in the theorem (see Dong et al. [11, Theorem 9, Fact 1] to verify that the trade-off function will be at least as hard when the subsampling ratio is smaller). This is what we do for the steps up to T. For the subsequent steps $t > T$, we would be allowed so use the same mechanism with μ_{step}-MIP but at a smaller batch size (this would usually require additional noise to be added). However, we do not perform these steps, thereby again making the functional composition of the tests harder.

D.4 Worst-case bound for stochastic composition

Theorem D.2 (Worst-case bounds for stochastic composition)**.** *Let $h : \mathcal{X} \to \mathcal{F}$ denote a mapping from the input space to a set of trade-off functions $\mathcal{F}$. Suppose there is a trade-off function f^* such that every other trade-off function $f \in \mathcal{F}$ is uniformly at least as hard as f^*,*

$$f \geq f^*, \forall f \in \mathcal{F}. \tag{39}$$

Then, the stochastic composition of trade-off functions will also be uniformly at least as hard as f^*, i.e.

$$\left(\bigotimes_{\boldsymbol{x} \sim \mathcal{D}} h(\boldsymbol{x}) \right) \geq f^* \tag{40}$$

regardless of the choice of h or the distribution $\mathcal{D}$.

Proof. Denote $H(r) \coloneqq \left(\bigotimes_{\boldsymbol{x}\sim\mathcal{D}} h(\boldsymbol{x})\right)(r)$ again for brevity.

$$H(r) = \min_{\bar{\alpha}\in\mathcal{E}(r,\mathcal{D})} \{\beta_h(\bar{\alpha})\} \tag{41}$$

where for every $\bar{\alpha} \in \mathcal{E}(r, \mathcal{D})$ we can bound

$$\beta_h(\bar{\alpha}) = \mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[h(\boldsymbol{x})(\bar{\alpha}(\boldsymbol{x}))\right] \geq \mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[f^*\left(\bar{\alpha}(\boldsymbol{x})\right)\right] \geq f^*\left(\mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[\bar{\alpha}(\boldsymbol{x})\right]\right) = f^*(r). \tag{42}$$

We use the Jensens inequality to derive that $\mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[f^*\left(\bar{\alpha}(\boldsymbol{x})\right)\right] \geq f^*\left(\mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}}\left[\bar{\alpha}(\boldsymbol{x})\right]\right)$ because f^* is convex. Therefore, we conclude that for every $r \in [0, 1]$

$$H(r) = \min_{\bar{\alpha}\in\mathcal{E}(r,\mathcal{D})} \{\beta_h(\bar{\alpha})\} \geq f^*(r). \tag{43}$$

□

E Proof of Theorem 5.1 and Corollary 5.1

For the sake of better readability, we summarize our setting before we proceed with the formal proof:

- $\Theta = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n], \boldsymbol{\theta}_i \sim P$, where $\boldsymbol{\theta}_i \in \mathbb{R}^d$. P can be any distribution with finite mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ (in our application, θ_i are gradients of the samples)
- The sample mean $\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\theta}_i$ is published
- $b \in \{0, 1\}$ is drawn uniformly at random. If $b = 0, \boldsymbol{x}' \sim \Theta$, if $b = 1$, $\boldsymbol{x}' \sim P$. $\boldsymbol{x}'$ is published.
- The attacker $\mathcal{A}(\boldsymbol{m}, \boldsymbol{x}') = b'$ attempts to predict the value of b, i.e., whether $\boldsymbol{x}'$ was in the training set or not.

The following hypothesis test succinctly summarizes the attacker's problem in the this setting[3]:

$$H_0 : \boldsymbol{x}' \text{ was drawn from } \boldsymbol{\Theta} \qquad H_1 : \boldsymbol{x}' \text{ was drawn from } P. \tag{44}$$

Based on this testing setup, the attacker constructs an attack based on a likelihood ratio test. Next, we summarize the individual proof steps before we give the formal proof:

1. Derive the distributions of $\boldsymbol{m}$ under the null and the alternative hypothesis for a given $\boldsymbol{x}'$.
2. Given these two distributions, we can setup the likelihood ratio, which will yield the test statistic S. This statistic will be optimal due to the Neyman-Pearson fundamental testing lemma;
3. Given the test statistic S and the distribution under the null hypothesis, we derive the rejection region of the likelihood ratio test for a given level α;
4. Finally, we derive the false negative rate $\beta(\alpha)$ of the likelihood ratio test. This trade-off function depends on the CDF and inverse CDF of non-central χ^2 distribution;
5. In a final step, we provide approximations to the trade-off function for large d.
6. Steps 1-5 initially prove f-membership inference privacy of a single SGD step when we add Gaussian noise with covariance $\hat{\tau}^2\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is the covariance of the gradients that we would like to privatize (see Appendix E.1). We then use this result to bound the privacy of SGD with unit noise in Appendix E.2. There we show that the result from Appendix E.1 can be used in combination with a scaled noise level to guarantee membership inference privacy when we add Gaussian noise with covariance $\tau^2\boldsymbol{I}$.

[3]Note that the hypotheses are interchanged with respect to the main paper here. Following the remarks after Corollary 2 of Dong et al., the trade-off function is inverted by when interchanging H_0, H_1. To arrive at the trade-off in the main paper, we will later invert the trade-off function derived here.

E.1 Proof of Theorem 5.1 and Corollary 5.1 with data dependent noise

In this section, we consider the effect that averaging with Gaussian noise has on membership inference privacy. We first add Gaussian noise with covariance $\hat{\tau}^2\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is the covariance of the gradients that we would like to privatize. In the next section, we will consider the case of independent unit noise $\tau^2\boldsymbol{I}$, which can be derived from the result presented here. The proof in this subsection follows the steps outlined in the previous section.

Proof. **Step 1: Deriving the distributions of $\boldsymbol{m}$ under H_0 and H_1.** First, we derive the distributions of $\boldsymbol{m}$ for both cases of interest. We suppose that the number of averaged samples is sufficiently large such that we can apply the Central Limit Theorem. Note that this does not restrict the form of the distribution P, besides having finite variance. Below, we start with the distribution of $\boldsymbol{m}$ under H_0 (with no additional noise yet):

$$\boldsymbol{m} \sim \mathcal{N}\left(\frac{1}{n}\boldsymbol{x}' + \frac{n-1}{n}\boldsymbol{\mu}, \frac{(n-1)}{n^2}\boldsymbol{\Sigma}\right) = \mathcal{N}\left(\boldsymbol{\mu} + \frac{1}{n}(\boldsymbol{x}' - \boldsymbol{\mu}), \frac{(n-1)}{n^2}\boldsymbol{\Sigma}\right). \tag{45}$$

Moreover, under the alternative hypothesis H_1, we have:

$$\boldsymbol{m} \sim \mathcal{N}\left(\boldsymbol{\mu}, \frac{1}{n}\boldsymbol{\Sigma}\right). \tag{46}$$

Instead of testing the distributions of $\boldsymbol{m}$ we directly, we can equivalently test $\boldsymbol{m} - \boldsymbol{x}'$ by subtracting $\boldsymbol{x}'$ from both means. Adding Gaussian noise $Y \sim \mathcal{N}(\boldsymbol{0}, \hat{\tau}^2\boldsymbol{\Sigma})$ under both hypotheses results in the test that we provide below:

$$\bigotimes_{\boldsymbol{x}' \sim \mathcal{D}} \text{Test}\left[\mathcal{N}\left(\frac{n-1}{n}\boldsymbol{\mu}, \frac{(n-1)}{n^2}\boldsymbol{\Sigma}\right) - \frac{n-1}{n}\boldsymbol{x}' + Y, \mathcal{N}\left(\boldsymbol{\mu}, \frac{1}{n}\boldsymbol{\Sigma}\right) - \boldsymbol{x}' + Y\right]. \tag{47}$$

Again, we now consider the test for a fixed $\boldsymbol{x}'$. If we can show that there is one $\boldsymbol{x}'$ that makes the test harder than any other $\boldsymbol{x}' \in \mathcal{X}$, we can apply Theorem D.2 and show that the composed test is uniformly at least as hard as for $\boldsymbol{x}'$.

We conduct the following reformulations (note that the hardness of a test remains unaffected by by invertible transforms, e.g., linear transforms):

$$\text{Test}\left[\mathcal{N}\left(\frac{n-1}{n}\boldsymbol{\mu}, \frac{(n-1)}{n^2}\boldsymbol{\Sigma}\right) - \frac{n-1}{n}\boldsymbol{x}' + Y, \mathcal{N}\left(\boldsymbol{\mu}, \frac{1}{n}\boldsymbol{\Sigma}\right) - \boldsymbol{x}' + Y\right] \tag{48}$$

$$\iff \text{Test}\left[\mathcal{N}\left(\frac{n-1}{n}\left(\boldsymbol{\mu} - \boldsymbol{x}'\right), \frac{(n-1)}{n^2}\boldsymbol{\Sigma} + \hat{\tau}^2\boldsymbol{\Sigma}\right), \mathcal{N}\left(\boldsymbol{\mu} - \boldsymbol{x}', \frac{1}{n}\boldsymbol{\Sigma} + \hat{\tau}^2\boldsymbol{\Sigma}\right)\right] \tag{49}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n}\left(\boldsymbol{\mu} - \boldsymbol{x}'\right), \left(\frac{(n-1)}{n^2} + \hat{\tau}^2\right)\boldsymbol{\Sigma}\right), \mathcal{N}\left(\boldsymbol{0}, \left(\frac{1}{n} + \hat{\tau}^2\right)\boldsymbol{\Sigma}\right)\right] \tag{50}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n\sqrt{n^{-1} + \hat{\tau}^2}}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\boldsymbol{\mu} - \boldsymbol{x}'\right), \left(\frac{n-1}{n^2} + \hat{\tau}^2\right)\left(\frac{1}{n} + \hat{\tau}^2\right)^{-1}\boldsymbol{I}\right), \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\right)\right] \tag{51}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n\sqrt{n^{-1} + \hat{\tau}^2}}\tilde{\boldsymbol{\delta}}, \frac{n(1 + \hat{\tau}^2 n) - 1}{n(1 + \hat{\tau}^2 n)}\boldsymbol{I}\right), \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\right)\right], \tag{52}$$

where $\tilde{\boldsymbol{\delta}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\boldsymbol{\mu} - \boldsymbol{x}'\right) = \tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}'$ and the tilde indicates the corresponding quantities transformed by $\Sigma^{-\frac{1}{2}}$. For instance, to arrive at Equation (50) and Equation (51) the random variable $\boldsymbol{m} - \boldsymbol{x}'$ is transformed by subtracting $(\boldsymbol{\mu} - \boldsymbol{x}')$ and multiplied by $\frac{1}{\sqrt{n^{-1} + \hat{\tau}^2}}\boldsymbol{\Sigma}^{-\frac{1}{2}} = \sqrt{\frac{n^2}{n + n^2\hat{\tau}^2}}\boldsymbol{\Sigma}^{-\frac{1}{2}}$, respectively. This yields the following transformed likelihood ratio test for the transformed random variable

$$Q \coloneqq \sqrt{\frac{n^2}{n + n^2\hat{\tau}^2}}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left((\boldsymbol{m} - \boldsymbol{x}') - (\boldsymbol{\mu} - \boldsymbol{x}')\right) = \sqrt{\frac{n^2}{n + n^2\hat{\tau}^2}}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\boldsymbol{m} - \boldsymbol{\mu}\right) \tag{53}$$

$$= \sqrt{\frac{n^2}{n + n^2\hat{\tau}^2}}(\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}). \tag{54}$$

Step 2: Deriving the test statistic. By the Neyman-Pearson Lemma [22, Theorem 3.2.1.], conducting the likelihood ratio test will be most powerful test at a given false positive rate. The corresponding likelihood ratio is given as follows:

$$\text{LR} = \frac{p_0(Q)}{p_1(Q)} = \frac{\mathcal{N}\left(Q; \boldsymbol{\mu}_1, \sigma_1^2 \boldsymbol{I}\right)}{\mathcal{N}\left(Q; \boldsymbol{\mu}_2, \sigma_2^2 \boldsymbol{I}\right)} \tag{55}$$

$$= \frac{\mathcal{N}\left(Q; -\frac{1}{\sqrt{n+n^2\hat{\tau}^2}} \boldsymbol{\Sigma}^{-\frac{1}{2}} \left(\boldsymbol{\mu} - \boldsymbol{x}'\right), \frac{n+n^2\hat{\tau}^2-1}{n+n^2\hat{\tau}^2} \boldsymbol{I}\right)}{\mathcal{N}\left(Q; \boldsymbol{0}, \boldsymbol{I}\right)} \tag{56}$$

$$= c_2 \mathcal{N}\left(Q; \boldsymbol{d}, \boldsymbol{D}\right). \tag{57}$$

We can use identities for the ratio of two normal distributions (with $\boldsymbol{\mu}_1 = -\frac{1}{\sqrt{n(1+n\hat{\tau}^2)}}\tilde{\boldsymbol{\delta}}, \boldsymbol{\mu}_2 = \boldsymbol{0}, \boldsymbol{\Sigma}_1 = \frac{n(1+\hat{\tau}^2 n)-1}{n(1+\hat{\tau}^2 n)} \boldsymbol{I}, \boldsymbol{\Sigma}_2 = \boldsymbol{I}$) and obtain

$$\boldsymbol{D} = (\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})^{-1} = \left(\frac{n(1+\hat{\tau}^2 n) - 1}{n(1+\hat{\tau}^2 n)} - \frac{n(1+\hat{\tau}^2 n)}{n(1+\hat{\tau}^2 n)}\right)^{-1} \boldsymbol{I} = (n + \hat{\tau}^2 n^2 - 1)\boldsymbol{I} \tag{58}$$

and

$$\boldsymbol{d} = \boldsymbol{D}\left(\boldsymbol{\Sigma}_1^{-1}\mu_1 - \boldsymbol{\Sigma}_2^{-1}\mu_2\right) = (n + \hat{\tau}^2 n^2 - 1)\frac{n(1+\hat{\tau}^2 n)}{n(1+\hat{\tau}^2 n) - 1}\left(-\frac{1}{\sqrt{n+\hat{\tau}^2 n^2}}\tilde{\boldsymbol{\delta}}\right) \tag{59}$$

$$= -\sqrt{n+\hat{\tau}^2 n^2}\tilde{\boldsymbol{\delta}} = -\sqrt{n+\hat{\tau}^2 n^2}\left(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}'\right). \tag{60}$$

For a Gaussian likelihood ratio of the form in Equation (57), i.e., $S = c_2 \exp\left(-\frac{1}{2}(\boldsymbol{s}' - \boldsymbol{d})^\top \boldsymbol{D}^{-1}(\boldsymbol{s}' - \boldsymbol{d})\right)$, where $\boldsymbol{s}' = \sqrt{\frac{n^2}{n+n^2\hat{\tau}^2}}(\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}})$ it suffices to use the inner argument as a test statistic, as exp is an invertible transform. Therefore, we can use the following as a test statistic:

$$S = \sqrt{\frac{n^2}{n+n^2\hat{\tau}^2}}\left((\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) + \frac{n+\hat{\tau}^2 n^2}{n}\tilde{\boldsymbol{\delta}}\right)^\top \frac{1}{(n+n^2\hat{\tau}^2) - 1}\boldsymbol{I}\sqrt{\frac{n^2}{n+n^2\hat{\tau}^2}}\left((\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) + \frac{n+\hat{\tau}^2 n^2}{n}\tilde{\boldsymbol{\delta}}\right) \tag{61}$$

$$= \frac{n^2}{((n+n^2\hat{\tau}^2) - 1)(n+n^2\hat{\tau}^2)}\left((\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) + \frac{n+\hat{\tau}^2 n^2}{n}\tilde{\boldsymbol{\delta}}\right)^\top \left((\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) + \frac{n+\hat{\tau}^2 n^2}{n}\tilde{\boldsymbol{\delta}}\right), \tag{62}$$

for which we can derive the closed-form distributions under both the null and alternative hypotheses.

Step 3: Deriving the distributions of S under H_0 and H_1. From above, we know that, under the respective hypotheses we have:

$$H_0 : \sqrt{\frac{n^2}{n+n^2\hat{\tau}^2}}(\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) \sim \mathcal{N}\Big(-\frac{1}{\sqrt{n+n^2\hat{\tau}^2}}\tilde{\boldsymbol{\delta}}, \frac{n+n^2\hat{\tau}^2-1}{n+n^2\hat{\tau}^2}\mathbf{I}\Big) \tag{63}$$

$$H_1 : \sqrt{\frac{n^2}{n+n^2\hat{\tau}^2}}(\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) \sim \mathcal{N}\big(\mathbf{0}, \mathbf{I}\big) \tag{64}$$

and

$$H_0 : (\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) \sim \mathcal{N}\Big(-\frac{1}{n}\tilde{\boldsymbol{\delta}}, \frac{n+n^2\hat{\tau}^2-1}{n^2}\mathbf{I}\Big) \tag{65}$$

$$H_1 : (\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) \sim \mathcal{N}\big(\mathbf{0}, \frac{n+n^2\hat{\tau}^2}{n^2}\mathbf{I}\big). \tag{66}$$

and hence, for $\tilde{\boldsymbol{l}} = (\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) + \frac{n+n^2\hat{\tau}^2}{n}\tilde{\boldsymbol{\delta}} = (\tilde{\boldsymbol{m}} - \tilde{\boldsymbol{\mu}}) + \frac{n+n^2\hat{\tau}^2}{n}(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}')$ the distributions are given by:

$$H_0 : \tilde{\boldsymbol{l}} \sim \mathcal{N}\Big(\frac{n-1}{n}(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}') + n\hat{\tau}^2(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}'), \frac{n+n^2\hat{\tau}^2-1}{n^2}\mathbf{I}\Big) \tag{67}$$

$$H_1 : \tilde{\boldsymbol{l}} \sim \mathcal{N}\Big((\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}') + n\hat{\tau}^2(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}'), \frac{n+n^2\hat{\tau}^2}{n^2}\mathbf{I}\Big). \tag{68}$$

Next, note that if $\tilde{\boldsymbol{l}} \sim \mathcal{N}(\boldsymbol{l}, \kappa \boldsymbol{I})$, and $\frac{\tilde{\boldsymbol{l}}}{\sqrt{\kappa}} \sim \mathcal{N}(\frac{\boldsymbol{l}}{\sqrt{\kappa}}, \boldsymbol{I})$ then we have that

$$\|\tilde{\boldsymbol{l}}\|_2^2 = \kappa \sum_{j=1}^{d} (U_j + b_j)^2, \tag{69}$$

where U_j are standard normal variables. Hence, $\|\tilde{\boldsymbol{l}}\|_2^2$ follows the law of a (scaled) non-central chi-squared distribution with d degrees of freedom and $\boldsymbol{b} = \frac{1}{\sqrt{\kappa}}\boldsymbol{l}$, i.e.,

$$\frac{\|\tilde{\boldsymbol{l}}\|_2^2}{\kappa} \sim \chi_d'^2(\gamma), \tag{70}$$

where $\gamma = \sum_{j=1}^{d} b_j^2 = \sum_{j=1}^{d} \left(\frac{1}{\sqrt{\kappa}} l_j\right)^2 = \frac{1}{\kappa} \sum_{j=1}^{d} l_j^2$.

Under the null hypothesis, we obtain $\kappa_0 = \frac{n+n^2\hat{\tau}^2-1}{n^2}$ and $\boldsymbol{l}_0 = \frac{n-1+n^2\hat{\tau}^2}{n}(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}')$. Therefore, the distribution of the test statistic under the null hypothesis is given by the scaled, non-central chi-squared random variable:

$$H_0 : S = \frac{n^2\kappa_0}{(n + n^2\hat{\tau}^2 - 1)(n + n^2\hat{\tau}^2)} \frac{\|\tilde{\boldsymbol{l}}\|_2^2}{\kappa_0} \sim \frac{n^2\kappa_0}{(n + n^2\hat{\tau}^2 - 1)(n + n^2\hat{\tau}^2)} \chi_d'^2(\gamma_0) \tag{71}$$

$$\sim \frac{1}{n + n^2\hat{\tau}^2} \chi_d'^2(\gamma_0). \tag{72}$$

In summary,

$$(n + n^2\hat{\tau}^2) S \sim \chi_d'^2(\gamma_0), \tag{73}$$

where $\gamma_0 = \|\boldsymbol{l}_0\|_2^2 / \kappa_0$.

Step 4: Deriving the rejection region for any α. Therefore, the rejection region of the null hypothesis at a significance level of α can be formulated as:

$$\left\{(n + \hat{\tau}^2 n^2) S \geq \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha)\right\} = \left\{S \geq \frac{1}{n + \hat{\tau}^2 n^2} \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha)\right\}. \tag{74}$$

Step 5: Deriving the false negative rate $\beta(\alpha)$ at any α. To compute the type two error rate β (the null hypothesis is accepted, but the alternative H_1 is true), we compute the probability of mistakenly accepting it. To this end, note that $\kappa_1 = \frac{n+n^2\hat{\tau}^2}{n^2}$ and that $\boldsymbol{l}_1 = (1 + +n\hat{\tau}^2)(\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}')$. Now, we can derive the distribution of the test statistic under the alternative hypothesis:

$$H_1 : S = \frac{n^2\kappa_1}{(n + n^2\hat{\tau}^2 - 1)(n + n^2\hat{\tau}^2)} \frac{\|\tilde{\boldsymbol{l}}\|_2^2}{\kappa_1} \sim \frac{1}{n + n^2\hat{\tau}^2 - 1} \chi_d'^2(\gamma_1), \tag{75}$$

where $\gamma_1 = \|\boldsymbol{l}_1\|_2^2 / \kappa_1$. Note that the form of this distribution,

$$(n + n^2\hat{\tau}^2 - 1) S \sim \chi_d'^2(\gamma_1), \tag{76}$$

where $\gamma_1 = n\|\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}'\|_2^2 \coloneqq nK$ when no noise is added ($\hat{\tau} = 0$) is the statistic used in the GLiR attack (Algorithm 1). The type two error is given by:

$$\begin{aligned}
\beta =& P_1\left(S \leq \frac{1}{n + n^2\hat{\tau}^2} \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha)\right) && (77)\\
=& P_1\left((n + n^2\hat{\tau}^2 - 1) S \leq \frac{n + n^2\hat{\tau}^2 - 1}{n + n^2\hat{\tau}^2} \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha))\right)\\
=& P\left(X \leq \frac{n + n^2\hat{\tau}^2 - 1}{n + n^2\hat{\tau}^2} \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha))\right)\\
=& \text{CDF}_{\chi_d'^2(\gamma_1)}\left(\frac{n + n^2\hat{\tau}^2 - 1}{n + n^2\hat{\tau}^2} \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha))\right)\\
=& \text{CDF}_{\chi_d'^2(\gamma_1)}\left(\frac{n + n^2\hat{\tau}^2 - 1}{n + n^2\hat{\tau}^2} \text{CDF}^{-1}_{\chi_d'^2(\gamma_0)}(1-\alpha))\right), && (78)
\end{aligned}$$

where $X \sim \chi_d'^2(\gamma_1)$.

Remark E.1. *Computing the inverse trade-off, i.e., solving for α would result in*

$$\alpha = 1 - CDF_{\chi'^2_d(\gamma_0)}\left(\frac{n+n^2\hat{\tau}^2}{n+n^2\hat{\tau}^2-1} CDF^{-1}_{\chi'^2_d(\gamma_1)}(\beta)\right), \tag{79}$$

giving the trade-off curve for the hypotheses as stated in main paper and resulting in Theorem 5.1. For the subsequent analysis this interchange of H_0, H_1 does not play a role as the trade-off function is symmetric for large d, n (it is its own inverse).

Step 6: Large d, n approximations. The following fact will be useful. Let $V \sim \chi'^2_d(\gamma)$, then $\frac{V-(d+\gamma)}{\sqrt{2(d+2\gamma)}} \to \mathcal{N}(0,1)$ when $d \to \infty$. The trade-off function for our hypothesis test can thus be expressed through the normal CDF Φ as:

$$\beta \approx \Phi\left(\frac{\left(\frac{n+\hat{\tau}^2n^2-1}{n+\hat{\tau}^2n^2}\text{CDF}^{-1}_{\chi'^2_d(\gamma_0)}(1-\alpha)\right) - (d+\gamma_1)}{\sqrt{2(d+2\gamma_1)}}\right) \tag{80}$$

$$\approx \Phi\left(\frac{\left(\frac{n+\hat{\tau}^2n^2-1}{n+\hat{\tau}^2n^2}\left(\sqrt{2(d+2\gamma_0)}\Phi^{-1}(1-\alpha)+(d+\gamma_0)\right)\right) - (d+\gamma_1)}{\sqrt{2(d+2\gamma_1)}}\right) \tag{81}$$

$$= \Phi\Bigg(\Phi^{-1}(1-\alpha) - \Phi^{-1}(1-\alpha)\frac{(n+n^2\hat{\tau}^2)s_1 - (n+n^2\hat{\tau}^2-1)s_0}{(n+n^2\hat{\tau}^2)s_1}$$
$$- \frac{(n+n^2\hat{\tau}^2)m_1 - (n+n^2\hat{\tau}^2-1)m_0}{(n+n^2\hat{\tau}^2)s_1}\Bigg), \tag{82}$$

where $s_1 = \sqrt{2(d+2\gamma_1)}$, $s_0 = \sqrt{2(d+2\gamma_0)}$, $m_0 = d+\gamma_0$ and $m_1 = d+\gamma_1$. Thus we have:

$$\beta \approx \Phi\Bigg(\Phi^{-1}(1-\alpha) - \Phi^{-1}(1-\alpha)\left(1 - \frac{n+\hat{\tau}^2n^2-1}{n+\hat{\tau}^2n^2}\sqrt{\frac{d+2\gamma_0}{d+2\gamma_1}}\right)$$
$$- \frac{(n+\hat{\tau}^2n^2)(d+\gamma_1) - (n+\hat{\tau}^2n^2-1)(d+\gamma_0)}{(n+\hat{\tau}^2n^2)\sqrt{2d+4\gamma_1)}}\Bigg), \tag{83}$$

For large n, $\frac{n+\hat{\tau}^2n^2-1}{n+\hat{\tau}^2n^2}\sqrt{\frac{d+2\gamma_0}{d+2\gamma_1}} \approx 1$ and the second term can be dropped in the approximation. Next, recall that $\gamma_0 = \frac{n^2\|\boldsymbol{l}_0\|^2}{n+n^2\hat{\tau}^2-1}$ and $\gamma_1 = \frac{n^2\|\boldsymbol{l}_1\|^2}{n+n^2\hat{\tau}^2}$:

$$\beta \approx \Phi\left(\Phi^{-1}(1-\alpha) - \frac{(n+\hat{\tau}^2n^2)(d+\frac{n^2\|\boldsymbol{l}_1\|^2}{n+n^2\hat{\tau}^2}) - (n+\hat{\tau}^2n^2-1)(d+\frac{n^2\|\boldsymbol{l}_0\|^2}{n+n^2\hat{\tau}^2-1})}{(n+\hat{\tau}^2n^2)\sqrt{2d+4\frac{n^2\|\boldsymbol{l}_1\|^2}{n+n^2\hat{\tau}^2})}}\right) \tag{84}$$

$$= \Phi\left(\Phi^{-1}(1-\alpha) - \frac{(n+\hat{\tau}^2n^2)d + n^2\|\boldsymbol{l}_1\|^2 - (n+\hat{\tau}^2n^2-1)d - n^2\|\boldsymbol{l}_0\|^2}{(n+\hat{\tau}^2n^2)\sqrt{2d+4\frac{n^2}{n+\hat{\tau}^2n^2}\|\boldsymbol{l}_1\|^2)}}\right) \tag{85}$$

$$= \Phi\left(\Phi^{-1}(1-\alpha) - \frac{d+n^2(\|\boldsymbol{l}_1\|^2 - \|\boldsymbol{l}_0\|^2)}{(n+\hat{\tau}^2n^2)\sqrt{2d+4\frac{n^2}{n+\hat{\tau}^2n^2}\|\boldsymbol{l}_1\|^2)}}\right) \tag{86}$$

$$= \Phi\left(\Phi^{-1}(1-\alpha) - \frac{d+n^2(\|\boldsymbol{l}_1\|^2 - \|\boldsymbol{l}_0\|^2)}{(n+\hat{\tau}^2n^2)\sqrt{2d+4\frac{n^2}{n+\hat{\tau}^2n^2}\|\boldsymbol{l}_1\|^2)}}\right). \tag{87}$$

We further obtain $\|\boldsymbol{l}_0\|^2 = \frac{(n-1+\hat{\tau}^2n^2)^2}{n^2}K$ and $\|\boldsymbol{l}_1\|^2 = \frac{(n+\hat{\tau}^2n^2)}{n^2}K$ where $K = \|\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{x}}'\|^2$. Therefore, we have

$$\beta \approx \Phi\left(\Phi^{-1}(1-\alpha) - \frac{d+(2(n+\hat{\tau}^2n^2)-1)K}{(n+\hat{\tau}^2n^2)\sqrt{2d+4(n+\hat{\tau}^2n^2)K}}\right). \tag{88}$$

Thus, for large n, d the trade-off curve can be well approximated by the μ-GMIP trade-off with:

$$\mu = \frac{d + \big(2(n + \hat{\tau}^2 n^2) - 1\big)K}{(n + \hat{\tau}^2 n^2)\sqrt{2d + 4(n + \hat{\tau}^2 n^2)K}}. \tag{89}$$

We observe that the hardness of the test decreases with K. Therefore, the stochastic composition of tests is uniformly at least as hard as the test with the largest K possible (Theorem D.2). We obtain the result shown in the paper by replacing the data-dependent noise $\hat{\tau}$ by data-independent noise τ at a ratio of $\hat{\tau}^2 = \tau^2/C^2$ as detailed in the next section. □

E.2 Proof of Theorem 5.1 and Corollary 5.1 with data independent noise

In the previous section, we have assumed that $Y \sim \hat{\tau}^2\boldsymbol{\Sigma}$. This assumption does not quite match common practice; in practice, the Gaussian mechanism adds noise of the form: $Y \sim \tau^2\boldsymbol{I}$. Hence, the following subsection investigates the effect of adding unit noise $Y \sim \tau^2\boldsymbol{I}$.

Proof. We first study the case of data-dependent noise using an eigenspace transform. Denoting an eigenvalue composition $\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ and performing the corresponding transform (multiplication by $\boldsymbol{Q}^\top$) in Equation (47), we obtain:

$$\iff \text{Test}\left[\mathcal{N}\left(\frac{n-1}{n}(\boldsymbol{\mu} - \boldsymbol{x}'), \frac{(n-1)}{n^2}\boldsymbol{\Sigma} + \hat{\tau}^2\boldsymbol{\Sigma}\right), \mathcal{N}\left(\boldsymbol{\mu} - \boldsymbol{x}', \frac{1}{n}\boldsymbol{\Sigma} + \hat{\tau}^2\boldsymbol{\Sigma}\right)\right] \tag{90}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n}\boldsymbol{Q}^\top(\boldsymbol{\mu} - \boldsymbol{x}'), \left(\frac{(n-1)}{n^2} + \hat{\tau}^2\right)\boldsymbol{Q}^\top\boldsymbol{\Sigma}\boldsymbol{Q}\right), \mathcal{N}\left(\mathbf{0}, \left(\frac{1}{n} + \hat{\tau}^2\right)\boldsymbol{Q}^\top\boldsymbol{\Sigma}\boldsymbol{Q}\right)\right] \tag{91}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n}\boldsymbol{Q}^\top(\boldsymbol{\mu} - \boldsymbol{x}'), \left(\frac{(n-1)}{n^2} + \hat{\tau}^2\right)\boldsymbol{\Lambda}\right), \mathcal{N}\left(\mathbf{0}, \left(\frac{1}{n} + \hat{\tau}^2\right)\boldsymbol{\Lambda}\right)\right]. \tag{92}$$

We see that the test decomposes in a series of d dimension-wise 1D hypotheses tests of the form:

$$\text{Test}\left[\mathcal{N}\left(-\frac{1}{n}\boldsymbol{Q}^\top(\boldsymbol{\mu} - \boldsymbol{x}')_i, \frac{n-1}{n^2}\lambda_i + \hat{\tau}^2\lambda_i\right), \mathcal{N}\left(0, \frac{1}{n}\lambda_i + \hat{\tau}^2\lambda_i\right)\right]. \tag{93}$$

Here, λ_i are the eigenvalues of $\boldsymbol{\Sigma}$ that are on the diagonal of the matrix $\boldsymbol{\Lambda}$. On the other hand, when adding unit noise of magnitude τ, i.e., setting $Y = \mathcal{N}(\mathbf{0}, \tau^2\boldsymbol{I})$ in Equation (47):

$$\text{Test}\left[\mathcal{N}\left(\frac{n-1}{n}(\boldsymbol{\mu} - \boldsymbol{x}'), \frac{(n-1)}{n^2}\boldsymbol{\Sigma} + \tau^2\boldsymbol{I}\right), \mathcal{N}\left(\boldsymbol{\mu} - \boldsymbol{x}', \frac{1}{n}\boldsymbol{\Sigma} + \tau^2\boldsymbol{I}\right)\right] \tag{94}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n}\boldsymbol{Q}^\top(\boldsymbol{\mu} - \boldsymbol{x}'), \left(\frac{(n-1)}{n^2}\boldsymbol{Q}^\top\boldsymbol{\Sigma}\boldsymbol{Q} + \tau^2\boldsymbol{Q}^\top\boldsymbol{I}\boldsymbol{Q}\right)\right), \tag{95}$$

$$\mathcal{N}\left(\mathbf{0}, \left(\frac{1}{n}\boldsymbol{Q}^\top\boldsymbol{\Sigma}\boldsymbol{Q} + \tau^2\boldsymbol{Q}^\top\boldsymbol{I}\boldsymbol{Q}\right)\right)\Bigg] \tag{96}$$

$$\iff \text{Test}\left[\mathcal{N}\left(-\frac{1}{n}\boldsymbol{Q}^\top(\boldsymbol{\mu} - \boldsymbol{x}'), \frac{(n-1)}{n^2}\boldsymbol{\Lambda} + \tau^2\boldsymbol{I}\right), \mathcal{N}\left(\mathbf{0}, \frac{1}{n}\boldsymbol{\Lambda} + \tau^2\boldsymbol{I}\right)\right]. \tag{97}$$

This test also decomposes in d 1D tests of the form:

$$\text{Test}\left[\mathcal{N}\left(\underbrace{-\frac{1}{n}\boldsymbol{Q}^\top(\boldsymbol{\mu} - \boldsymbol{x}')_i}_{\mu_i}, \underbrace{\frac{n-1}{n^2}\lambda_i + \tau^2}_{\sigma_1^2}\right), \mathcal{N}\left(0, \underbrace{\frac{1}{n}\lambda_i + \tau^2}_{\sigma_2^2}\right)\right]. \tag{98}$$

The test is therefore equally hard as testing d independent normal random variables because the covariance matrices shown are diagonal. We will show that by setting

$$\tau^2 = \hat{\tau}^2 \cdot \max_i\{\lambda_i\} \tag{99}$$

each individual, dimension-wise test is made strictly harder than the corresponding dimension-wise test in the data-dependent noise case and therefore the composed test is also harder (this is shown in Lemma D.1).

We will do so by computing the power function of the test and showing that is is montonically decreasing in τ, which means that for each individual test, that higher effective noise in that dimension makes the test harder. To show this, note that the likelihood ratio for this test results in

$$S = \frac{(m - \hat{\mu})^2}{\hat{\sigma}^2} \tag{100}$$

where

$$\hat{\sigma}^2 = \left(\frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right)^{-1} = \frac{\left(\frac{n-1}{n^2} \lambda_i + \tau^2 \right) \left(\frac{1}{n} \lambda_i + \tau^2 \right) n^2}{\lambda_i}, \tag{101}$$

$$\begin{aligned}
\hat{\mu} &= \hat{\sigma}^2 \left(\frac{1}{\sigma_1^2} \boldsymbol{\mu}_1 - \frac{1}{\sigma_2^2} \boldsymbol{\mu}_2 \right) && (102)\\
&= \hat{\sigma}^2 \frac{1}{\sigma_2^2 \sigma_1^2} \left(\sigma_2^2 \boldsymbol{\mu}_1 - \sigma_1^2 \boldsymbol{\mu}_2 \right) && (103)\\
&= \frac{\sigma_2^2 \sigma_1^2}{\sigma_2^2 - \sigma_1^2} \frac{1}{\sigma_2^2 \sigma_1^2} \left(\sigma_2^2 \boldsymbol{\mu}_1 - \sigma_1^2 \boldsymbol{\mu}_2 \right) && (104)\\
&= \frac{1}{\sigma_2^2 - \sigma_1^2} \left(\sigma_2^2 \boldsymbol{\mu}_1 - \sigma_1^2 \boldsymbol{\mu}_2 \right) && (105)\\
&= \frac{n^2}{\lambda_i} \left(-\frac{1}{n} \boldsymbol{Q}^\top \left(\boldsymbol{\mu} - \boldsymbol{x}' \right) \left(\frac{1}{n} \lambda_i + \tau^2 \right) \right) && (106)\\
&= -\boldsymbol{Q}^\top \left(\boldsymbol{\mu} - \boldsymbol{x}' \right)_i \left(\frac{n^2}{\lambda_i} \left(\frac{\lambda_i}{n^2} + \frac{\tau^2}{n} \right) \right) && (107)\\
&= -\boldsymbol{Q}^\top \left(\boldsymbol{\mu} - \boldsymbol{x}' \right)_i \left(1 + \frac{n\tau^2}{\lambda_i} \right) = n\mu_i \left(1 + \frac{n\tau^2}{\lambda_i} \right). && (108)
\end{aligned}$$

Distribution under the null hypothesis: Again, as before we derive the distribution under the null hypothesis:

$$\begin{aligned}
m - \hat{\mu} &\sim \mathcal{N} \left(\mu_i \left(1 - n \left(1 + \frac{n\tau^2}{\lambda_i} \right) \right), \frac{(n-1)}{n^2} \lambda_i + \tau^2 \right) && (109)\\
\frac{m - \hat{\mu}}{\sigma_1} &\sim \mathcal{N} \left(\mu_i \frac{1 - n - \frac{n^2 \tau^2}{\lambda_i}}{\sqrt{\frac{(n-1)}{n^2} \lambda_i + \tau^2}}, 1 \right) && (110)\\
\frac{m - \hat{\mu}}{\sigma_1} &\sim \mathcal{N} \left(-\mu_i \frac{n - 1 + \frac{n^2 \tau^2}{\lambda_i}}{\sqrt{\frac{\lambda_i}{n^2} \left(n - 1 + \frac{n^2 \tau^2}{\lambda_i} \right)}}, 1 \right) && (111)\\
\frac{m - \hat{\mu}}{\sigma_1} &\sim \mathcal{N} \left(-\mu_i \sqrt{n - 1 + \frac{n^2 \tau^2}{\lambda_i}} \sqrt{\frac{n^2}{\lambda_i}}, 1 \right) && (112)\\
\frac{m - \hat{\mu}}{\sigma_1} &\sim \mathcal{N} \left(-\mu_i \left(n \sqrt{\frac{n-1}{\lambda_i} + \frac{n^2 \tau^2}{\lambda_i^2}} \right), 1 \right) && (113)\\
\left(\frac{m - \hat{\mu}}{\sigma_1} \right)^2 &\sim \chi_1'^2 \left(\mu_i^2 n^2 \left(\frac{n-1}{\lambda_i} + \frac{n^2 \tau^2}{\lambda_i^2} \right) \right) = \chi_1'^2 \left(\gamma_0 \right), && (114)
\end{aligned}$$

where $\chi_d(\gamma)$ again denotes the non-central χ-distribution with d degrees of freedom.

Rejection region: Therefore, the rejection region of the null hypothesis at a significance level of α can be formulated as:

$$\left\{ \frac{\hat{\sigma}^2}{\sigma_1^2} S \geq \text{CDF}^{-1}_{\chi_1'^2(\gamma_0)}(1 - \alpha) \right\} = \left\{ S \geq \frac{\sigma_1^2}{\hat{\sigma}^2} \text{CDF}^{-1}_{\chi_1'^2(\gamma_0)}(1 - \alpha) \right\}, \tag{115}$$

where $\frac{\sigma_1^2}{\hat{\sigma}^2} = \frac{(\frac{n-1}{n}\lambda_i+\tau^2)\lambda_i}{n^2(\frac{n-1}{n}\lambda_i+\tau^2)(\frac{1}{n}\lambda_i+\tau^2)} = \frac{\lambda_i}{n^2(\frac{1}{n}\lambda_i+\tau^2)} = \frac{\lambda_i}{n\lambda_i+n^2\tau^2}$.

Distribution under the alternative hypothesis: As before, we also need to derive the distribution of the test statistic under the alternative hypothesis:

$$m - \hat{\mu} \sim \mathcal{N}\left(-n\mu_i\left(1+\frac{n\tau^2}{\lambda_i}\right), \frac{1}{n}\lambda_i + \tau^2\right) \tag{116}$$

$$\frac{m-\hat{\mu}}{\sigma_2} \sim \mathcal{N}\left(-n\mu_i\frac{1+\frac{n\tau^2}{\lambda_i}}{\sqrt{\frac{1}{n}\lambda_i+\tau^2}}, 1\right) \tag{117}$$

$$\frac{m-\hat{\mu}}{\sigma_2} \sim \mathcal{N}\left(-n\mu_i\frac{1+\frac{n\tau^2}{\lambda_i}}{\sqrt{\frac{\lambda_i}{n}\left(1+\frac{n\tau^2}{\lambda_i}\right)}}, 1\right) \tag{118}$$

$$\frac{m-\hat{\mu}}{\sigma_2} \sim \mathcal{N}\left(-\mu_i n\sqrt{\frac{n}{\lambda_i}\left(1+\frac{n\tau^2}{\lambda_i}\right)}, 1\right) \tag{119}$$

$$\left(\frac{m-\hat{\mu}}{\sigma_2}\right)^2 \sim \mathcal{X}_1'^2\left(\mu_i^2 n^2\left(\frac{n}{\lambda_i}+\frac{n^2\tau^2}{\lambda_i^2}\right)\right) \tag{120}$$

Therefore,

$$\frac{\hat{\sigma}^2}{\sigma_2^2}S \sim \mathcal{X}_1'^2\left(\mu_i^2 n^2\left(\frac{n}{\lambda_i}+\frac{n^2\tau^2}{\lambda_i^2}\right)\right) = \mathcal{X}_1'^2\left(\gamma_1\right). \tag{121}$$

False negative rate: Finally, 1 - power of the test is given by:

$$\beta(\alpha) = P\left\{S \leq \frac{\sigma_1^2}{\hat{\sigma}^2}\text{CDF}^{-1}_{\mathcal{X}_1'^2(\gamma_0)}(1-\alpha)\right\} \tag{122}$$

$$= P\left\{\frac{\hat{\sigma}^2}{\sigma_2^2}S \leq \frac{\hat{\sigma}^2}{\sigma_2^2}\frac{\sigma_1^2}{\hat{\sigma}^2}\text{CDF}^{-1}_{\mathcal{X}_1'^2(\gamma_0)}(1-\alpha)\right\} \tag{123}$$

$$= \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}\left(\frac{\sigma_1^2}{\sigma_2^2}\text{CDF}^{-1}_{\mathcal{X}_1'^2(\gamma_0)}(1-\alpha)\right). \tag{124}$$

We will now introduce two quantities on which the power of this test depends. In particular

$$q \coloneqq \frac{n+\frac{n^2\tau^2}{\lambda_i}-1}{n+\frac{n^2\tau^2}{\lambda_i}}, \tag{125}$$

which has the intriguing property that

$$\frac{\gamma_0}{\gamma_1} = \frac{\sigma_1^2}{\sigma_2^2} = q. \tag{126}$$

Using this insight, the trade-off function can be expressed as

$$\beta(\alpha, \gamma_1(\lambda_i,\mu_i), q(\lambda_i)) = \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}\left(q\text{CDF}^{-1}_{\mathcal{X}_1'^2(q\gamma_1)}(1-\alpha)\right). \tag{127}$$

We determine the hardest test by showing that when considering a fixed μ_i, α, we obtain

$$\frac{\partial\beta}{\partial(\tau^2)} > 0, \tag{128}$$

which indicates that at a higher level of noise, the type 2 error rate will increase, making the test harder (more privacy). The derivative computation can be found in Appendix E.2.1. When we set $\tau^2 = \hat{\tau}^2 \cdot \max_i\{\lambda_i\}$, the individual tests in the case of data-independent noise are thus harder (μ_i, n, d stays the same) than the respective dimension-wise test for the data-dependent noise. Therefore the corresponding bound derived for $\hat{\tau}$ will hold for data independent noise of strength τ as well.

The largest possible eigenvalue λ_i of the covariance matrix $\mathbf{\Sigma}$ when cropping each vector at norm C is given by C^2. Therefore, we can set $\tau^2 = \hat{\tau}^2 C^2$ to obtain a harder test with data independent noise. On the converse, applying data-independent noise of level τ^2 results in a harder test than applying data-dependent noise with $\hat{\tau}^2 = \frac{\tau^2}{C^2}$. Plugging this result in Equation (89) requires to only replace $\hat{\tau}^2 = \frac{\tau^2}{C^2}$ and introducing the quantity $n_{\text{effective}}$ as

$$n_{\text{effective}} = n + \frac{n^2\tau^2}{C^2}. \tag{129}$$

□

E.2.1 Derivatives of the type 2 error

In this section we calculate the derivatives of the type 2 error function

$$\beta(\alpha, \gamma_1(\lambda_i, \mu_i), q(\lambda_i) = \text{CDF}_{\chi'^2_1(\gamma_1)}\left(q\text{CDF}^{-1}_{\chi'^2_1(q\gamma_1)}(1-\alpha)\right). \tag{130}$$

Considering a fixed μ_i we show that

$$\frac{\partial \beta}{\partial(\tau^2)} > 0, \tag{131}$$

which indicates that with larger added noise, the type 2 error rate will decrease, making the test uniformly harder (more privacy). Thus the hardest test is the one with maximum τ^2. To emphasize that we are deriving by (τ^2) (the squared quantity), we write $\tau_2 \coloneqq \tau^2$.

We perform the following calculations:

Derivatives of CDFs and inverse CDFs. We will start by deriving some useful derivatives of the CDFs and inverse CDFs. We start with the inverse CDF. Initially, Let $u = \sqrt{\text{CDF}^{-1}_{\chi'^2_1(\gamma_0)}(1-\alpha)}$ or $u^2 = \text{CDF}^{-1}_{\chi'^2_1(\gamma_0)}(1-\alpha)$ (note that the quantile function here is always > 0):

$$\Phi(u - \sqrt{\gamma_0}) - \Phi(-(u + \sqrt{\gamma_0})) = \alpha. \tag{132}$$

Keeping α constant, we can use the implicit function derivative theorem with $F(\gamma_0, u(\gamma_0)) = \Phi(u - \sqrt{\gamma_0}) - \Phi(-(u + \sqrt{\gamma_0})) - \alpha = 0$, we obtain

$$\frac{\partial u}{\partial \gamma_0} = -\frac{-\phi(u - \sqrt{\gamma_0}))\frac{1}{2\sqrt{\gamma}} + \phi(-u - \sqrt{\gamma_0}))\frac{1}{2\sqrt{\gamma}}}{\phi(u - \sqrt{\gamma_0})) + \phi(-u - \sqrt{\gamma_0})} \tag{133}$$

$$= \frac{1}{2\sqrt{\gamma_0}}\frac{\phi(u - \sqrt{\gamma_0})) - \phi(-u - \sqrt{\gamma_0}))}{\phi(u - \sqrt{\gamma_0})) + \phi(-u - \sqrt{\gamma_0}))} \tag{134}$$

We furthermore have

$$\frac{\text{CDF}^{-1}_{\chi'^2_1(\gamma_0)}(1-\alpha)}{\partial \gamma_0} = 2\sqrt{\text{CDF}^{-1}_{\chi'^2_1(\gamma_0)}(1-\alpha)}\frac{\partial u}{\partial \gamma_0}. \tag{135}$$

We note that

$$\text{CDF}_{\chi'^2_1(\gamma_1)}(s) = \Phi(\sqrt{s} - \sqrt{\gamma_1}) - \Phi(-\sqrt{s} - \sqrt{\gamma_1})), \tag{136}$$

where s is the argument of the CDF. The derivative

$$\frac{\partial \text{CDF}_{\chi'^2_1(\gamma_1)}}{\partial \gamma_1} = -\frac{1}{2\sqrt{\gamma_1}}\left(\phi(\sqrt{s} - \sqrt{\gamma_1}) - \phi(-\sqrt{s} - \sqrt{\gamma_1})\right) \leq 0 \tag{137}$$

$$\frac{\partial \text{CDF}_{\chi'^2_1(\gamma_1)}(s)}{\partial s} = \frac{1}{2\sqrt{s}}\left(\phi(\sqrt{s} - \sqrt{\gamma_1}) + \phi(-\sqrt{s} - \sqrt{\gamma_1})\right) > 0. \tag{138}$$

Derivative with respect to μ_i. We compute the derivative with respect to μ_i first, which will be handy in the subsequent derivation of the derivative by τ_2. To denote that we consider a fixed α, we write β_α. We have

$$\beta_\alpha(\gamma_1(\mu_i, \lambda_i)), q(\lambda_i)) = \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}\left(q\text{CDF}^{-1}_{\mathcal{X}_1'^2(q\gamma_1)}(1-\alpha)\right). \tag{139}$$

Thus, we can calculate

$$\frac{\partial \beta_\alpha}{\partial \mu_i} = \frac{\partial \beta_\alpha}{\partial \gamma_1}\frac{\partial \gamma_1}{\partial \mu_i} = \underbrace{\left(\frac{\partial \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial \gamma_1} + \frac{\partial \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial s}\frac{\partial s}{\partial \gamma_1}\right)}_{<0?}\underbrace{\frac{\partial \gamma_1}{\partial \mu_i}}_{>0}, \tag{140}$$

where $s = q\text{CDF}^{-1}_{\mathcal{X}_1'^2(q\gamma_1)}(1-\alpha)$. We can confirm that $\frac{\partial \gamma_1}{\partial \mu_i} > 0$ because

$$\gamma_1 = \mu_i^2 n^2 \left(\frac{n}{\lambda_i} + \frac{n^2\tau^2}{\lambda_i^2}\right) \tag{141}$$

is monotonically increasing in μ_i, because all eigenvalues $\lambda_i > 0$ and the factor after μ_i^2 is positive overall. To establish that the overall derivative is negative, we therefore have to confirm that the first factor is always negative:

$$\frac{\partial \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial \gamma_1} = -\frac{1}{2\sqrt{\gamma_1}}\left(\phi(\sqrt{s}-\sqrt{\gamma_1}) - \phi(-\sqrt{s}-\sqrt{\gamma_1})\right) \tag{142}$$

$$\frac{\partial s}{\partial \gamma_1} = q\frac{\text{CDF}^{-1}_{\mathcal{X}_1'^2(q\gamma_1)}(1-\alpha)}{\partial \gamma_1} = 2q^2\sqrt{\text{CDF}^{-1}_{\mathcal{X}_1'^2(q\gamma_1)}(1-\alpha)}\frac{\partial u}{\partial \gamma_0}\Big|_{\gamma_0 = q\gamma_1}. \tag{143}$$

Let $s = q\text{CDF}^{-1}_{\mathcal{X}_1'^2(qL)}(1-\alpha)$

$$\frac{\partial \text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial s}\frac{\partial s}{\partial \gamma_1} = \frac{\text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial s}2q^2\sqrt{\text{CDF}^{-1}_{\mathcal{X}_1'^2(q\gamma_1)}(1-\alpha)}\frac{\partial u}{\partial \gamma_0}|_{\gamma_0 = q\gamma_1} \tag{144}$$

$$= \frac{\text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial s}2q^2\sqrt{\frac{s}{q}}\frac{\partial u}{\partial \gamma_0}|_{\gamma_0 = q\gamma_1} \tag{145}$$

$$= \frac{\text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial s}2q^2\sqrt{\frac{s}{q}}\frac{1}{2\sqrt{\gamma_0}}\frac{\phi(u-\sqrt{\gamma_0})) - \phi(-u-\sqrt{\gamma_0}))}{\phi(u-\sqrt{\gamma_0})) + \phi(-u-\sqrt{\gamma_0}))}. \tag{146}$$

We can plug in $\gamma_0 = \gamma_1 q$, and $u = \frac{\sqrt{s}}{\sqrt{q}}$

$$= \frac{\text{CDF}_{\mathcal{X}_1'^2(\gamma_1)}(s)}{\partial s}2q^2\sqrt{\frac{s}{q}}\frac{1}{2\sqrt{\gamma_1 q}}\frac{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})) - \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}))}{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})) + \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}))} \tag{147}$$

$$= \frac{1}{2\sqrt{s}}\left(\phi(\sqrt{s}-\sqrt{\gamma_1}) + \phi(-\sqrt{s}-\sqrt{\gamma_1})\right)\frac{q\sqrt{s}}{\sqrt{\gamma_1}}\frac{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})) - \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}))}{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})) + \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}))} \tag{148}$$

$$= \frac{1}{2}\frac{q}{\sqrt{\gamma_1}}\left(\phi(\sqrt{s}-\sqrt{\gamma_1}) + \phi(-\sqrt{s}-\sqrt{\gamma_1})\right)\frac{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})) - \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}))}{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}) + \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})} \tag{149}$$

In total we obtain

$$\frac{\partial \beta_\alpha(\gamma_1)}{\partial \gamma_1} = -\frac{1}{2\sqrt{\gamma_1}}\left(\phi(\sqrt{s}-\sqrt{\gamma_1}) - \phi(-\sqrt{s}-\sqrt{\gamma_1})\right) \tag{150}$$

$$+ \frac{1}{2}\frac{q}{\sqrt{\gamma_1}}\left(\phi(\sqrt{s}-\sqrt{\gamma_1}) + \phi(-\sqrt{s}-\sqrt{\gamma_1})\right)\frac{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})) - \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}))}{\phi(\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q}) + \phi(-\frac{\sqrt{s}}{\sqrt{q}}-\sqrt{\gamma_1 q})} \tag{151}$$

$$= \frac{1}{2\sqrt{\gamma_1}}\left(-S_1 + qS_2\frac{T_1}{T_2}\right). \tag{152}$$

By inspection of the terms, we see that

$$S_1 = \phi(\sqrt{s} - \sqrt{\gamma_1}) - \phi(-\sqrt{s} - \sqrt{\gamma_1}) \geq 0 \tag{153}$$

because $|\sqrt{s} - \sqrt{\gamma_1}| \leq |-\sqrt{s} - \sqrt{\gamma_1}|$ and the normal density ϕ decreases with the absolute value of its argument. We can therefore rewrite the expression as

$$\frac{\partial \beta_\alpha(\gamma_1)}{\partial \gamma_1} = \frac{S_1}{2\sqrt{\gamma_1}} \left(-1 + q \frac{S_2 T_1}{S_1 T_2} \right) \tag{154}$$

and the sign is determined by the second factor (the first fraction $\frac{S_1}{2\sqrt{\gamma_1}}$ will be non-negative).

$$\frac{S_2 T_1}{S_1 T_2} = \frac{\left(\phi(\sqrt{s} - \sqrt{\gamma_1}) + \phi(-\sqrt{s} - \sqrt{\gamma_1})\right)}{\left(\phi(\sqrt{s} - \sqrt{\gamma_1}) - \phi(-\sqrt{s} - \sqrt{\gamma_1})\right)} \frac{\left(\phi(\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q})) - \phi(-\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q}))\right)}{\left(\phi(\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q}) + \phi(-\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q}))\right)} \tag{155}$$

$$= \frac{\Gamma - \left(\phi(\sqrt{s} - \sqrt{\gamma_1})\phi(-\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q}) - \phi(-\sqrt{s} - \sqrt{\gamma_1})\phi(\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q})\right)}{\Gamma + \left(\phi(\sqrt{s} - \sqrt{\gamma_1})\phi(-\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q}) - \phi(-\sqrt{s} - \sqrt{\gamma_1})\phi(\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q})\right)} \tag{156}$$

$$= \frac{\Gamma - \Delta}{\Gamma + \Delta}, \tag{157}$$

where we have:

$$\Delta = \frac{1}{2\pi} \left(\exp\left(-\frac{(\sqrt{s} - \sqrt{\gamma_1})^2 + (-\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q})^2}{2} \right) - \exp\left(-\frac{(-\sqrt{s} - \sqrt{\gamma_1})^2 + (\frac{\sqrt{s}}{\sqrt{q}} - \sqrt{\gamma_1 q})^2}{2} \right) \right) \tag{158}$$

$$= \frac{1}{2\pi} \left(\exp\left(-\frac{s - 2\sqrt{s}\sqrt{\gamma_1} + \gamma_1 + \frac{s}{q} + 2\frac{\sqrt{s}\sqrt{\gamma_1 q}}{\sqrt{q}} + \gamma_1 q}{2} \right) \right. \tag{159}$$

$$\left. - \exp\left(-\frac{s + 2\sqrt{s}\sqrt{\gamma_1} + \gamma_1 + \frac{s}{q} - 2\frac{\sqrt{s}\sqrt{\gamma_1 q}}{\sqrt{q}} + \gamma_1 q}{2} \right) \right) \tag{160}$$

$$= \frac{1}{2\pi} \left(\exp\left(-\frac{s + \gamma_1 + \frac{s}{q} + \gamma_1 q}{2} \right) - \exp\left(-\frac{s + \gamma_1 + \frac{s}{q} + \gamma_1 q}{2} \right) \right) \tag{161}$$

$$= 0. \tag{162}$$

Bringing this result ($\frac{S_2 T_1}{S_1 T_2} = 1$) back, we obtain

$$\frac{\partial \beta_\alpha(\gamma_1)}{\partial \gamma_1} = -\frac{S_1}{2\sqrt{\gamma_1}} (1 - q) \leq 0. \tag{163}$$

Derivative w.r.t. τ^2. We denote $\tau^2 = \tau_2$ to emphasize that we are deriving with respect to the square of τ:

$$\frac{\partial \beta_\alpha}{\partial \tau_2} = \underbrace{\frac{\partial \beta_\alpha}{\partial \gamma_1}}_{<0} \underbrace{\frac{\partial \gamma_1}{\partial \tau_2}}_{>0} + \underbrace{\frac{\partial \beta_\alpha}{\partial q}}_{>0} \underbrace{\frac{\partial q}{\partial \tau_2}}_{>0}. \tag{164}$$

We perform the following calculations:

$$\frac{\partial \beta_\alpha}{\partial \gamma_1} = -\frac{S_1}{2\sqrt{\gamma_1}} (1 - q) < 0 \tag{165}$$

$$\frac{\partial \gamma_1}{\partial \tau_2} = \frac{n^4 \mu^2}{\lambda_i^2} > 0 \tag{166}$$

$$\frac{\partial \beta_\alpha}{\partial q} = \frac{S_2 \sqrt{s}}{2q} + \frac{S_1 \sqrt{\gamma_1}}{2} > 0 \tag{167}$$

$$\frac{\partial q}{\partial \tau_2} = \frac{n^2}{\lambda_i (n + r)^2} > 0, \tag{168}$$

where $r = \frac{n^2\tau_2}{\lambda_i}$. Plugging the terms together, we see that

$$\frac{\partial\beta_\alpha}{\partial\tau_2} = -\frac{S_1}{2\sqrt{\gamma_1}}(1-q)\frac{n^4\mu^2}{\lambda_i^2} + \left(\frac{S_2\sqrt{s}}{2q} + \frac{S_1\sqrt{\gamma_1}}{2}\right)\frac{n^2}{\lambda_i(n+r)^2} \tag{169}$$

$$= \frac{n^2}{2\lambda_i}\left(-\frac{S_1}{\sqrt{\gamma_1}}(1-q)\frac{n^2\mu^2}{\lambda_i} + \left(\frac{S_2\sqrt{s}}{q} + S_1\sqrt{\gamma_1}\right)\frac{1}{(n+r)^2}\right), \tag{170}$$

using $1-q = \frac{1}{n+\frac{n^2\tau^2}{\lambda_i}} = \frac{1}{n+r}$ we obtain

$$\frac{\partial\beta_\alpha}{\partial\tau_2} = \frac{n^2}{2\lambda_i(n+r)}\left(-\frac{S_1}{\sqrt{\gamma_1}}\frac{n^2\mu^2}{\lambda_i} + \left(\frac{S_2\sqrt{s}}{q} + S_1\sqrt{\gamma_1}\right)\frac{1}{(n+r)}\right). \tag{171}$$

Further using $\frac{n^2\mu^2}{\lambda_i^2} = \frac{\gamma_1}{n+r}$ we obtain

$$\frac{\partial\beta_\alpha}{\partial\tau_2} = \frac{n^2}{2\lambda_i(n+r)^2}\left(-S_1\sqrt{\gamma_1} + \frac{S_2\sqrt{s}}{q} + S_1\sqrt{\gamma_1}\right) \tag{172}$$

$$= \frac{S_2 n^2\sqrt{s}}{2q\lambda_i(n+r)^2} > 0. \tag{173}$$

F On the Relation between f-MIP and f-DP

F.1 Proof of Theorem 4.2

We first restate the theorem:

Theorem 4.2 (f-DP implies f-MIP). *Let an algorithm $\mathcal{A} : D^n \to \mathbb{R}^d$ be f-differentially private [11]. Then, the algorithm $\mathcal{A}$ will also be f-membership inference private.*

Proof. If the (probabilistic) algorithm $\mathcal{A}$ is f-DP, this requires that for all neighboring datasets S and S' (where only a single instance is changed), we have [11]

$$\text{Test}\left(A(S), A(S')\right) \geq f. \tag{174}$$

This includes any randomly sampled dataset $\mathcal{X} \in D^{n-1}$ plus an instance $\boldsymbol{x}$ as well as the dataset $\mathcal{X}$ plus the additional instance $\boldsymbol{x}'$ that is known to the attacker. Therefore setting

$$S = \mathcal{X} \cup \{\boldsymbol{x}\} \tag{175}$$

$$S' = \mathcal{X} \cup \{\boldsymbol{x}'\} \tag{176}$$

we immediately see that for every $\mathcal{X}$, $\boldsymbol{x}$, $\boldsymbol{x}'$, under f-DP, we have

$$\text{Test}\left(\mathcal{A}(\mathcal{X} \cup \{\boldsymbol{x}\}), \mathcal{A}(\mathcal{X} \cup \{\boldsymbol{x}'\})\right) \geq f. \tag{177}$$

Consulting our hypothesis test formulation of MI attacks, we note that the sets S, S' correspond to the inputs to A which result in output distributions A_0 and $A_1(\boldsymbol{x}')$ in the formulation of Equation (3), but for a specific $\boldsymbol{x}$, $\mathcal{X}$.

Additionally, as opposed to DP, in the membership inference attack scenario, the attacker has no knowledge of $\boldsymbol{x}$, $\mathcal{X}$. As the privacy constraint in Equation (177) holds for each individual choice of $\boldsymbol{x}$, $\mathcal{X}$, the test with randomly sampled instances cannot be easier than the test in Equation (177) (in practice, the test will be much harder when $\boldsymbol{x}, \mathcal{X}$ are stochastic as well), i.e.,

$$\forall \boldsymbol{x}' : \ \text{Test}\left(A_0; A_1(\boldsymbol{x}')\right) \geq f. \tag{178}$$

We can now leverage Theorem D.2 which highlights that when stochastically drawing from tests that are bounded by a certain trade-off function, the composed test can also be bounded by this function, which results in

$$\bigotimes_{\boldsymbol{x}' \sim \mathcal{D}} \text{Test}\left(A_0; A_1(\boldsymbol{x}')\right) \geq f, \tag{179}$$

the definition of f-MIP. □

F.2 On the Correspondence between μ-DP and μ-MIP through noisy SGD

Our results on noisy SGD in Theorem 5.1 allow to translate between the parameters μ_{DP} of μ-GDP and μ_{MIP} of μ-GMIP, *when our algorithm is used to guarantee these privacy notions*. This relation can be made explicit through the following Corollary:

Corollary F.1 (Translating μ values between DP and MIP)**.** *For one step of DP-SGD and the usual setting of $K = d$, we can convert the privacy parameters μ_{DP} and μ_{MIP} as follows:*

1. *If the step is μ_{DP}-GDP it will be μ_{MIP}-GMIP with*

$$\mu_{MIP} = \min\left\{\sqrt{\frac{d}{n + \frac{4C^2}{\mu_{DP}^2} + \frac{1}{2}}},\ \mu_{DP}\right\}.$$

2. *If the step is μ_{MIP}-GMIP it will be μ_{DP}-GDP with*

$$\mu_{DP} = \begin{cases} \frac{2}{\sqrt{\frac{d}{\mu_{MIP}^2} - n - \frac{1}{2}}}, & \text{if } \mu_{MIP} < \sqrt{\frac{2d}{2n+1}} \\ \infty, & \text{else.} \end{cases}$$

Proof. The result can be seen by solving for the required noise level τ. When guaranteeing μ_{DP}-GDP, we have

$$\mu_{\text{DP}} = \frac{1}{\sigma} = \frac{2C}{n\tau} \tag{180}$$

by [11]. Additionally, for MIP ($K = d$) we have

$$\mu_{\text{MIP}} = \frac{2n_{\text{effective}}d}{n_{\text{effective}}\sqrt{2d + 4n_{\text{effective}}d}} = \sqrt{\frac{2d}{2n_{\text{effective}} + 1}}. \tag{181}$$

(1) We can solve Equation (180) for τ arriving at

$$\tau = \frac{2C}{n\mu_{\text{DP}}} \tag{182}$$

and plug it into Equation (181) to arrive at the first term of the $\min$ in statement 1. We note that due to Theorem 4.2, the level of μ_{MIP} cannot be higher than the level of μ_{DP} and thus take the min of both terms.

(2) To arrive at statement 2, we perform the opposite conversion and solve Equation (181) for τ. Solving for $n_{\text{effective}}$, we obtain

$$n_{\text{effective}} = n + \frac{n^2\tau^2}{C^2} = \frac{d}{\mu_{\text{MIP}}^2} - \frac{1}{2}. \tag{183}$$

As we have $\tau^2 > 0$ and therefore require $n_{\text{effective}} > n$, no solution exists for $\mu_{\text{MIP}}^2 < \frac{2d}{2n+1}$. For the remaining values, we obtain

$$\tau^2 = C^2\frac{\frac{d}{\mu_{\text{MIP}}^2} - \frac{1}{2} - n}{n^2} \tag{184}$$

which, plugged into Equation (180), gives rise to the result in statement 2. □

4

Discussion and Conclusion

Having presented and discussed our contributions individually, we elaborate on their connections in this final section. Additionally, we discuss the relation to the remaining components of TSRML that were not considered in the main part of this thesis and identify pressing questions for future work.

4.1 Connecting Privacy and Interpretability

We have studied privacy and interpretability mostly independently in our contributions. In this section, we will explore the connections between the two disciplines. We highlight possible challenges and opportunities when both privacy and interpretability requirements have to be met at the same time.

To begin, we think that privacy and interpretability must not be at odds with each other but can often be implemented as orthogonal requirements. The privacy methods proposed in this work (PUC through PUC-IDA, GMIP through DP-SGD) are applied at data curation or training time. The considered explanation techniques, conceptual explanations, counterfactuals and feature attribution methods are applied post-hoc. They can further be applied to any model, including private models trained with GMIP, PUC, or both. In the standard setting of private training, the model parameters are learned with DP-SGD resulting in tangible privacy guarantees. The information-theoretic post-processing property also holds for f-MIP and ensures that once the output of an algorithm is privatized, its privacy cannot be reduced through post-processing (Dong et al., 2022, cf. Proposition 4). Applying post-hoc explanations to a trained model can be seen as a form of post-processing of these parameters and therefore does not present additional attack surface in terms of DP. In conclusion, we can usu-

ally apply the post-hoc explanations on private models without further ramifications. Nevertheless, there are cases where further consideration is necessary, particularly when the model is not privatized using a DP mechanism, or when the explanations leverage additional, non-private information.

Privacy attacks leveraging XAI. If models are not trained using privacy-preserving training, attacks can leverage explanations to gain additional advantage. Notably, Shokri et al. (2021) study membership inference attacks based on post-hoc explanations such as SHAP (Lundberg and Lee, 2017). The privacy guarantee only holds when the model is used as an input to compute the explanation. Some counterfactual explanations require access to training data as an additional input to make sure the counterfactual examples are in-distribution (Poyiadzi et al., 2020; Redelmeier et al., 2024). Privacy-preserving generative approaches like private generative models (Jordon et al., 2018) or private clustering (Su et al., 2017) can potentially be used to enhance privacy characteristics of in-distribution recourse methods. These findings highlight that there are setups with a non-trivial trade-off between privacy and interpretability such that caution is advised. We think a careful evaluation of this trade-off may lead to interesting results in future work. Besides attacks on the data, Aïvodji et al. (2020) show that counterfactual explanations can be leveraged for model stealing attacks. These observations highlight that in some use cases that do not follow the paradigm of DP training with pure post-hoc explanations, the potential impact of interpretability on privacy needs to be carefully gauged.

Identifying privacy risks with XAI. We are also interested in studying whether interpretability techniques can augment privacy in a positive way. In Section 3.5 introducing GMIP, we observe that the membership inference risk is dependent on the characteristics of an individual instance (intuitively, outliers with respect to the data distribution incur a higher MI risk). In a workshop contribution (Leemann et al., 2024), we consider whether explanations can be used to better identify such individual instances at risk of privacy leakage and potentially inform protection mechanisms. For loss-based membership inference attacks, e.g., Shokri et al. (2017), the prediction loss is most predictive of attack risk, which is expected. For other attacks that are not loss-based, e.g., CFD (Pawelczyk et al., 2023b) and GLiR (Leemann et al., 2023a), we find that the variance of SHAP attributions between the features of an instance is highly predictive of MIA risk on the IMDB sentiment classification dataset (Maas et al., 2011). These initial results raise hope that explanations can be beneficial to identify points at risk and ultimately inform defences. Unfortunately, the results are not supportive of this conclusion for image datasets, highlighting that further work on this topic is required for a deeper understanding of this novel connection. We encourage further research studying the trade-offs but also potential merits of the combining

interpretability and privacy techniques.

4.2 Additional building blocks of TSRML

In this section, we briefly discuss other important components of TSRML outside the scope of this dissertation. Specifically, TSRML also includes the topic of fairness and other AI safety topics besides privacy, for instance robustness against distribution shifts or adversarial attacks (Goodfellow et al., 2014). We discuss the compatibility of our techniques with methods for implementing the remaining TSRML desiderata to build truly trustworthy systems that satisfy all relevant requirements at the same time.

Fairness in ML. Fairness constraints can be added to machine learning models to prevent discrimination between protected groups. These are usually differentiated by one attribute value, e.g., gender, ethnicity, etc. (Verma and Rubin, 2018). This attribute is often referred to as a protected attribute. A most basic example for such a constraint is *statistical parity* (Dwork et al., 2012; Kusner et al., 2017), demanding that all protected protected groups have an equal probability of getting a positive classification outcome. Popular alternatives are *equalized odds* and *equal opportunity* (Hardt et al., 2016), *predictive parity* (Chouldechova, 2017) or *individual fairness* (Dwork et al., 2012). However, choosing the right notion of fairness for a specific application is very challenging, especially since it has been shown that some of the notions are incompatible and cannot be satisfied at the same time (Chouldechova, 2017; Kleinberg et al., 2016). We do not consider strategies to choose a suitable notion of fairness in this thesis and refer the reader to frameworks such as the "Fairness Compass" by Ruf and Detyniecki (2021) for this purpose.

We are interested in methods to implement these fairness constraints alongside the interpretability and privacy notions discussed in this work. Fairness notions are commonly implemented by adding constraints to the optimization problem for classifiers such as Support Vector Machines (SVMs) or logistic regression models (Zafar et al., 2017). For more complex classifiers like deep learning models, some of the fairness constraints can be transformed to regularizers that are added to the loss (Bendekgey and Sudderth, 2021).

Regarding post-hoc interpretability, this has few implications. The interpretability techniques can be applied to fair models trained both ways as the model design does not change. For post-hoc explanations, we thus see no major ramifications when combining fairness constraints with the interpretability methods proposed in this thesis.

	Interpretability	Privacy	Fairness	Robustness
Interpretability	*			
Privacy	✓	*		
Fairness	✓	◯	*	
Robustness	✓	✓✓	◯	*

Table 4.1: Simplified illustration of compatibility challenges between components of TSRML. We indicate combinations where the respective methods can be applied at the same time without major ramifications by ✓, combinations where implementing one component may even support the other by ✓✓, and combinations where more caution is advised because non-trivial trade-offs are common by ◯.

However, combining fairness and privacy is substantially more challenging as both are usually enforced at training time. Bagdasaryan et al. (2019) show that when DP is applied, it might have disparate effects on different marginalized groups and thus adversarially impact fairness. There are variants of DP-SGD that can be applied using a regularized loss or are specifically adapted to fairness notions like disparate impact (Xu et al., 2021). However, most fairness regularizers cannot be decomposed to an instance level, and calculating the fairness loss incurs additional privacy costs in DP-SGD.

Besides this concern, our work on privacy by handing control back to the user (through PUC) also shows a potential discrimination and fairness risk. These findings highlight that there are non-trivial trade-offs between fairness and privacy that need to be considered. Unfortunately, there are only few works like Yaghini et al. (2023), which explicitly consider this trade-off between fairness and privacy. Their results show that the characterization of this trade-off is non-trivial, and the research area between fairness and privacy remains to be more thoroughly mapped in future research.

Safety and robustness. Safety, for instance as mentioned in the AI Act, also contains other risks beside privacy attacks. Most notably, Art. 15 of the AIA (European Parliament, 2024) requires *"an appropriate level of accuracy, robustness, and cybersecurity"* for high-risk AI systems. The legislation explicitly mentions data poisoning and model poisoning attacks, where attackers attempt to supply

malicious data points or model components for the training process. We believe that those attacks can be most effectively addressed before the training run, by either relying only on data from trusted and secure sources, or by carefully examining data from unreliable sources (e.g., crawled data from the web). Data quality should be monitored during and after training, for instance, using data attribution methods (Nguyen et al., 2024; Wang et al., 2024). These checks are generally independent of the methods proposed in this thesis.

Robustness can be enhanced during training and the data curation stage. One of the most popular training time techniques is regularization, for instance through adversarial training (Madry et al., 2017; Shaham et al., 2018) or dropout layers (Srivastava et al., 2014). Interestingly, privacy already can be interpreted as a notion of regularization (Kulynych et al., 2022). It is well known that adding noise to the data is a form of regularization (Bishop, 1995), but it can also be seen as a way to anonymize training data to fulfill DP requirements. For instance, the US Census Bureau uses this technique to ensure DP in their data releases (Majeed and Lee, 2020). Intuitively, robustness and privacy pursue technically similar goals by attempting to bound a (potentially probabilistic) function's sensitivity to small changes in the inputs and can go hand in hand. Some works have already shown how DP-methods such as noising layers can be used to ensure adversarial robustness (Lecuyer et al., 2019). On top of regularization, monitoring techniques like uncertainty estimation (Kendall and Gal, 2017; Van Amersfoort et al., 2020) can be helpful for detecting out-of-distribution examples. Distribution shift detection can be applied to detect natural or adversarial shifts in data streams and may even leverage interpretability methods like importance estimates for improved performance (Haug et al., 2020). We conclude that robustness and privacy have the potential to support or even reinforce each other with suitable techniques, but leave a thorough investigation of benefits and trade-offs to future work.

Concerning interpretability, we note that common post-hoc techniques can readily be applied to regularized or robust models as well. We also note that simpler models, which are more explainable, are also often more robust (e.g., linear models, shallow trees, cf. Molnar, 2019), indicating good compatibility between the two.

We summarize our findings on the compatibility of the main TSRML components in Table 4.1. We would like to point out that the above picture is simplified and only considers the basic techniques. Due to the post-hoc and model-agnostic paradigm of many XAI methods, it is simple to apply them to models that fulfill other desiderata. We think that combining privacy and fairness in a suitable way remains one of the biggest challenges. Due to the inherent similarities between robustness and privacy, we also indicated fairness and robustness as challeng-

ing as well. Problems such as disparate impact on groups through regularization also require additional consideration. To advance towards holistic TSRML, studying these fundamental trade-offs will be critical in future work. As every application has its own TSRML requirements and methods to implement them, their individual trade-offs may be different as well and still require careful consideration.

4.3 Conclusion

In this thesis, we studied interpretability and privacy as key components for Trustworthy and Socially Responsible Machine Learning (TSRML). We condensed these requirements from recent legal frameworks in the EU and distilled them into practical, technical solutions. We observed that research often either takes a merely technical or a user-centric perspective which has led to several misconceptions in the past. In this thesis, we therefore proposed realistic ways to model the user of TSRML systems with their respective goals and needs. Bridging the two perspectives, we provide practitioners with useful tools and insights for building accurate, explainable, and privacy-preserving ML systems. Our results indicate that technically sound and user-friendly solutions often exist, but may require additional considerations.

We identified several open research questions beyond the scope of this dissertation. It remains unclear whether and how explainable AI techniques can potentially inform users and model owners of privacy threats. In addition, we would like to emphasize that the interplay between privacy and various other TSRML requirements, such as fairness, needs more explicit characterization. While we have taken a first step towards the user by explicitly considering and modeling the human users in our work, we would also like to suggest further verification through controlled subject studies.

As TSRML requirements increasingly manifest in legislation, it will be interesting to see how the newly founded AI office will interpret these requirements (Pavlidis, 2024). While there is still significant ambiguity and uncertainty, we hope this thesis will provide valuable guidance to practitioners and lawmakers on what is feasible and lay the ground for law-compliant TSRML implementations. We believe that a careful collaboration and exchange between policymakers and the scientific community is needed to advance toward the greater goal of making machine learning more transparent, safer, and accessible to all.

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Sushant Agarwal, Shahin Jabbari, Chirag Agarwal, Sohini Upadhyay, Steven Wu, and Himabindu Lakkaraju. Towards the unification and robustness of perturbation and gradient based explanations. In *International conference on machine learning*, pages 110–119. PMLR, 2021.

Ulrich Aïvodji, Alexandre Bolot, and Sébastien Gambs. Model extraction from counterfactual explanations. *arXiv preprint arXiv:2009.01884*, 2020.

Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *AAAI conference on artificial intelligence*, volume 34, pages 2594–2601, 2020.

Noor Saleh Alfaiz and Suliman Mohamed Fati. Enhanced credit card fraud detection model using machine learning. *Electronics*, 11(4):662, 2022.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks, 2016.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58:82–115, 2020.

Asymina Aza. Scores as decisions? Article 22 GDPR and the judgment of the

CJEU in SCHUFA holding (scoring) in the labour context. *Industrial law journal*, page dwae035, 2024.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of machine learning research*, 11:1803–1831, 2010.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

Michael Barbaro, Tom Zeller, and Saul Hansell. A face is exposed for AOL searcher no. 4417749. *New York Times*, 9(2008):8, 2006.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

Henry C Bendekgey and Erik Sudderth. Scalable and stable surrogates for flexible classifiers with fairness constraints. *Advances in neural information processing systems*, 34:30023–30036, 2021.

Asia J Biega and Michèle Finck. Reviving purpose limitation and data minimisation in data-driven systems. *Columbia science and technology law review*, 315: 317, 2021.

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Miranda Bogen and Aaron Rieke. Help wanted: An examination of hiring algorithms, equity, and bias. *Upturn, December*, 7, 2018.

Sebastian Bordt, Michèle Finck, Eric Raidl, and Ulrike von Luxburg. Post-hoc explanations fail to achieve their purpose in adversarial contexts. In *ACM conference on fairness, accountability, and transparency*, pages 891–905, 2022.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium*, pages 2633–2650, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*, pages 1897–1914. IEEE, 2022.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*, 2024.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *USENIX Security Symposium*, pages 5253–5270, 2023.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Michael Chromik, Malin Eiband, Felicitas Buchner, Adrian Krüger, and Andreas Butz. I think i get your point, AI! the illusion of explanatory depth in explainable AI. In *International conference on intelligent user interfaces*, pages 307–317, 2021.

Xolani Dastile, Turgay Celik, and Moshe Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied soft computing*, 91:106263, 2020.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the royal statistical society: Series B (statistical methodology)*, 84(1):3–37, 2022.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Lena Enqvist. 'Human oversight' in the EU artificial intelligence act: what, when and by whom? *Law, innovation and technology*, 15(2):508–535, 2023.

European Parliament. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such

data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official journal of the European Union*, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

European Parliament. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act). *Official journal of the European Union*, 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

Timo Freiesleben. The intriguing relation between counterfactual explanations and adversarial examples. *Minds and machines*, 32(1):77–109, 2022.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Nils Gruschka, Vasileios Mavroeidis, Kamer Vishi, and Meiko Jensen. Privacy issues and data protection in big data: a case study analysis under GDPR. In *IEEE international conference on big data*, pages 5027–5033. IEEE, 2018.

David Gunning and David Aha. DARPA's explainable artificial intelligence (XAI) program. *AI magazine*, 40(2):44–58, 2019.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in neural information processing systems*, 36, 2023.

Johannes Haug, Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Leveraging model inherent variable importance for stable online feature selection. In *26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1478–1502, 2020.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European conference on computer vision*, pages 3–19. Springer, 2016.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *European conference on computer vision*, pages 264–279, 2018.

Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in neural information processing systems*, 32, 2019.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM computing surveys*, 54(11s):1–37, 2022.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

Georgios Kaissis, Alexander Ziller, Stefan Kolek, Anneliese Riess, and Daniel Rueckert. Optimal privacy guarantees for a relaxed threat model: Addressing sub-optimal adversaries in differentially private machine learning. *Advances in neural information processing systems*, 36, 2023.

Gautham Kamath. Lecture 1: Some attempts at data privacy, 2020. URL `http://www.gautamkamath.com/CS860notes/lec1.pdf`.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6199*, 2013.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

Patrick Kolpaczki, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier. Approximating the shapley value without marginal contributions. In *AAAI conference on artificial intelligence*, volume 38, pages 13246–13255, 2024.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Bogdan Kulynych, Yao-Yuan Yang, Yaodong Yu, Jarosław Błasiok, and Preetum Nakkiran. What you see is what you get: Principled deep learning via distributional generalization. *Advances in neural information processing systems*, 35: 2168–2183, 2022.

Bogdan Kulynych, Hsiang Hsu, Carmela Troncoso, and Flavio P Calmon. Arbitrary decisions are a hidden cost of differentially private training. In *ACM Conference on fairness, accountability, and transparency*, pages 1609–1623, 2023.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE symposium on security and privacy*, pages 656–672. IEEE, 2019.

Tobias Leemann, Yao Rong, Stefan Kraft, Enkelejda Kasneci, and Gjergji Kasneci. Coherence Evaluation of Visual Concepts With Objects and Language. In *ICLR2022 workshop on the elements of reasoning: objects, structure and causality*, 2022.

Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Gaussian membership inference privacy. *Advances in neural information processing systems*, 36, 2023a.

Tobias Leemann, Yao Rong, Thai-Trang Nguyen, Enkelejda Kasneci, and Gjergji Kasneci. Caution to the exemplars: On the intriguing effects of example choice on human trust in XAI. In *XAI in action: Past, present, and future applications (NeurIPS 2023 workshop)*, 2023b. URL https://openreview.net/forum?id=SCcOu4hJ97.

Tobias Leemann, Bardh Prenkaj, and Gjergji Kasneci. Is my data safe? predicting instance-level membership inference success for white-box and black-box attacks. In *ICML 2024 next generation of AI safety workshop*, 2024.

Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational*

linguistics, pages 150–157, 2003.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, page 4768–4777, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.

Abdul Majeed and Sungchang Lee. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. *IEEE access*, 9:8512–8545, 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

Christoph Molnar. *Interpretable Machine Learning*. 2019.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in neural information processing systems*, 33:17153–17163, 2020.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE symposium on security and privacy*, pages 111–125. IEEE, 2008.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023a.

Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In *32nd USENIX security symposium)*, pages 1631–1648, 2023b.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review

on evaluating explainable ai. *ACM computing surveys*, 55(13s):1–42, 2023.

Elisa Nguyen, Minjoon Seo, and Seong Joon Oh. A bayesian approach to analysing training data attribution in deep learning. *Advances in neural information processing systems*, 36, 2024.

Raymond S Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175–220, 1998.

Henrik Nolte, Miriam Rateike, and Michele Finck. Robustness and cybersecurity in the EU artificial intelligence act. In *Generative AI and law Workshop at 41st international conference on machine learning*, 2024.

Cecilia Panigutti, Ronan Hamon, Isabelle Hupont, David Fernandez Llorca, Delia Fano Yela, Henrik Junklewitz, Salvatore Scalzo, Gabriele Mazzini, Ignacio Sanchez, Josep Soler Garrido, et al. The role of explainable AI in the context of the AI act. In *ACM conference on fairness, accountability, and transparency*, pages 1139–1150, 2023.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. SoK: Security and privacy in machine learning. In *IEEE European symposium on security and privacy*, pages 399–414. IEEE, 2018.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *IEEE conference on computer vision and pattern recognition*, pages 8779–8788, 2018.

Georgios Pavlidis. Unlocking the black box: analysing the EU artificial intelligence act's framework for explainability in AI. *Law, innovation and technology*, 16(1):293–308, 2024.

Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *The web conference*, pages 3126–3132, 2020.

Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In *International conference on artificial intelligence and statistics*, pages 4574–4594. PMLR, 2022.

Martin Pawelczyk, Teresa Datta, Johannes Van-den Heuvel, Gjergji Kasneci, and Himabindu Lakkaraju. Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In *Interna-*

tional conference on learning representations, 2023a.

Martin Pawelczyk, Himabindu Lakkaraju, and Seth Neel. On the privacy risks of algorithmic recourse. In *International conference on artificial intelligence and statistics*, pages 9680–9696. PMLR, 2023b.

Martin Pawelczyk, Tobias Leemann, Asia Biega, and Gjergji Kasneci. On the trade-off between actionable explanations and the right to be forgotten. In *International conference on learning representations*, 2023c.

PayPal Editorial Staff. Harnessing machine learning fraud detection technologies, 2024. URL https://www.paypal.com/us/brc/article/payment-fraud-detection-machine-learning.

Christian Peukert, Stefan Bechtold, Michail Batikas, and Tobias Kretschmer. Regulatory spillovers and data governance: Evidence from the GDPR. *Marketing science*, 41(4):746–768, 2022.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: feasible and actionable counterfactual explanations. In *AAAI/ACM conference on AI, ethics, and society*, pages 344–350, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Conference on fairness, accountability, and transparency*, pages 469–481, 2020.

Annabelle Redelmeier, Martin Jullum, Kjersti Aas, and Anders Løland. MCCE: Monte carlo sampling of valid and realistic counterfactual explanations for tabular data. *Data mining and knowledge discovery*, pages 1–32, 2024.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM computing surveys*, 56(4):1–34, 2023.

Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. Evaluating feature attribution: An information-theoretic perspective. In *International conference on machine learning*, pages 18770 – 18795, 2022.

Yao Rong, Tobias Leemann, Thai-Trang Nguyen, Lisa Fiedler, Peizhu Qian, Vaib-

hav Unhelkar, Tina Seidel, Gjergji Kasneci, and Enkelejda Kasneci. Towards human-centered explainable AI: A survey of user studies for model explanations. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

Leonid Rozenblit and Frank Keil. The misunderstood limits of folk science: An illusion of explanatory depth. *Cognitive science*, 26(5):521–562, 2002.

Boris Ruf and Marcin Detyniecki. Towards the right kind of fairness in AI. *arXiv preprint arXiv:2102.08453*, 2021.

Jérôme Rutinowski, Simon Klüttermann, Jan Endendyk, Christopher Reining, and Emmanuel Müller. Benchmarking trust: A metric for trustworthy machine learning. In *World conference on explainable artificial intelligence*, pages 287–307. Springer, 2024.

Naomi Saphra and Sarah Wiegreffe. Mechanistic? In *The 7th BlackboxNLP workshop*, 2024.

A. Saranya and R. Subhashini. A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends. *Decision analytics journal*, 7:100230, 2023.

Miriam Schirmer, Tobias Leemann, Gjergji Kasneci, Jürgen Pfeffer, and David Jurgens. The language of trauma: Modeling traumatic event descriptions across domains with explainable AI. In *Findings of the association for computational linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE international conference on computer vision*, pages 618–626, 2017.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? *Advances in neural information processing systems*, 34:2046–2059, 2021.

Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018.

Lloyd S Shapley. A value for n-person games. *Contribution to the theory of games*, 2, 1953.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE symposium on security and privacy*, pages 3–18. IEEE, 2017.

Reza Shokri, Martin Strobel, and Yair Zick. On the privacy risks of model explanations. In *AAAI/ACM conference on AI, ethics, and society*, pages 231–241, 2021.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. In *International conference on learning representations*, 2014.

Leon Sixt, Martin Schuessler, Oana-Iuliana Popescu, Philipp Weiß, and Tim Landgraf. Do users benefit from interpretable vision? a user study, baseline, and dataset. In *International conference on learning representations*, 2022.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM conference on AI, ethics, and society*, pages 180–186, 2020.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data–anonymisation groundhog day. In *31st USENIX security symposium*, pages 1451–1468, 2022.

Statista. Number of purchase transactions on global general purpose card brands american express, diners/discover, jcb, mastercard, unionpay and visa from 2014 to 2023, 2023. URL https://www.statista.com/statistics/261327/number-of-per-card-credit-card-transactions-worldwide-by-brand-as-of-2011/.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, Min Lyu, and Hongxia Jin. Differentially private k-means clustering and a hybrid approach to private optimization. *ACM transactions on privacy and security*, 20(4):1–33, 2017.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–

570, 2002.

C Szegedy. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer circuits thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Anvith Thudi, Ilia Shumailov, Franziska Boenisch, and Nicolas Papernot. Bounding membership inference. *arXiv preprint arXiv:2202.12232*, 2022.

Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics. In *AAAI conference on artificial intelligence*, volume 34, pages 6021–6029, 2020.

Ehsan Toreini, Mhairi Aitken, Kovila Coopamootoo, Karen Elliott, Carlos Gonzalez Zelaya, and Aad Van Moorsel. The relationship between trust in AI and trustworthy machine learning technologies. In *Conference on fairness, accountability, and transparency*, pages 272–283, 2020.

Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *International conference on machine learning*, pages 9583–9592. PMLR, 2020.

Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. Towards robust and reliable algorithmic recourse. *Advances in neural information processing systems*, 34:16926–16937, 2021.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Conference on fairness, accountability, and transparency*, pages 10–19, 2019.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Sahil Verma and Julia Rubin. Fairness definitions explained. In *IEEE/ACM International workshop on software fairness*, pages 1–7. IEEE, 2018.

Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International data privacy law*, 7(2):76–99, 2017.

Jiachen T Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. *arXiv preprint arXiv:2406.11011*, 2024.

Chris Whong. FOILing NYC's taxi trip data. https://chriswhong.com/open-data/foil_nyc_taxi, 2014.

Yufei Xia, Chuanzhe Liu, YuYing Li, and Nana Liu. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert systems with applications*, 78:225–241, 2017.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

Depeng Xu, Wei Du, and Xintao Wu. Removing disparate impact on model accuracy in differentially private stochastic gradient descent. In *27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1924–1932, 2021.

Mohammad Yaghini, Patty Liu, Franziska Boenisch, and Nicolas Papernot. Learning with impartiality to walk on the pareto frontier of fairness, privacy, and utility. *arXiv preprint arXiv:2302.09183*, 2023.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *ACM SIGSAC conference on computer and communications security*, pages 3093–3106, 2022.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in neural information processing systems*, 32, 2019.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33: 20554–20565, 2020.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE 31st*

computer security foundations symposium, pages 268–282. IEEE, 2018.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR, 2017.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.