Towards Robust Machine Learning: Benchmarking and Adaptation in Challenging Settings

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Dipl.-Inform. Ori Press

aus Petah Tikva, Israel

Tübingen

2025

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

Tag der mündlichen Qualifikation:	25.07.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Matthias Bethge
2. Berichterstatter/-in:	Prof. Dr. Seong Joon Oh

For Avia.

Abstract

Neural networks often excel when their inputs closely match the data on which they were trained, yet they frequently fail when inputs differ even slightly from their training data. This issue, known as distribution shift, remains a significant challenge when deploying machine learning models in practical applications such as medical imaging and autonomous driving. Traditional methods to address distribution shift typically involve additional training or data collection, which may not always be feasible for models already deployed. This thesis explores alternative strategies aimed at enhancing the robustness of already trained models to distribution shifts.

The first part of this work introduces a benchmark specifically designed to evaluate test-time adaptation (TTA) methods under prolonged and varied distribution shifts. Using this benchmark, we demonstrate that while existing TTA techniques initially improve performance, they often lead to performance degradation with extended adaptation. We also propose a simple baseline method capable of consistently outperforming other tested methods, maintaining high performance even throughout prolonged adaptation.

Building on these insights, the second part analyzes the underlying mechanisms of entropy-based loss functions commonly employed in TTA. We show that entropy minimization initially clusters embeddings of similar images together, thus increasing accuracy. However, continued entropy minimization eventually drives input image embeddings further away from training embeddings, thereby reducing accuracy. Leveraging this insight, we propose Weighted Flips (WF), a novel method capable of predicting model accuracy on arbitrary image sets without the need for labeled data.

The final part of this work extends the principles of TTA to language models (LMs), focusing on the task of literature recommendation. We propose a benchmark that evaluates LMs in their ability to infer academic papers when given a short description that references them. Our

benchmark demonstrates that LMs are unable to effectively perform this task. Therefore, we propose a simple agent that allows LMs to search for and read relevant papers, significantly improving their performance.

Kurzfassung

Neuronale Netze erzielen oft hervorragende Ergebnisse, wenn ihre Eingaben den Daten ähneln, auf denen sie trainiert wurden. Sie versagen jedoch häufig, sobald sich die Eingaben auch nur geringfügig von ihren Trainingsdaten unterscheiden. Dieses Problem, bekannt als Distribution Shift (Verteilungsshift), stellt weiterhin eine große Herausforderung dar, wenn maschinelle Lernmodelle in praktischen Anwendungen wie der medizinischen Bildgebung oder dem autonomen Fahren eingesetzt werden. Traditionelle Ansätze zur Bewältigung des Distribution Shifts umfassen typischerweise zusätzliches Training oder die Sammlung neuer Daten, was jedoch nicht immer für bereits eingesetzte Modelle praktikabel ist. Diese Arbeit untersucht daher alternative Strategien, um die Robustheit bereits trainierter Modelle gegenüber Distribution Shifts zu verbessern.

Im ersten Teil dieser Arbeit wird ein Benchmark vorgestellt, der speziell zur Bewertung von Testzeit-Adaptionsmethoden (TTA) bei langanhaltenden und vielfältigen Distribution Shifts entwickelt wurde. Mithilfe dieses Benchmarks zeigen wir, dass bestehende TTA-Techniken zwar zunächst die Leistung verbessern, jedoch bei länger anhaltender Adaptation oft zu einem Leistungsabfall führen. Zusätzlich schlagen wir eine einfache Basismethode vor, die durchgängig bessere Ergebnisse als die anderen getesteten Methoden erzielt und ihre Leistung selbst bei längerer Adaptation aufrechterhält.

Aufbauend auf diesen Erkenntnissen analysiert der zweite Teil die zugrundeliegenden Mechanismen entropiebasierter Verlustfunktionen, die häufig in TTA verwendet werden. Wir zeigen, dass die Minimierung der Entropie zunächst dazu führt, dass Einbettungen ähnlicher Bilder näher zusammenrücken, was die Genauigkeit erhöht. Allerdings führt eine fortgesetzte Entropieminimierung dazu, dass sich die Einbettungen der Eingabebilder von den ursprünglichen Trainingseinbettungen entfernen, was letztlich die Genauigkeit verringert. Basierend auf dieser

Erkenntnis schlagen wir Weighted Flips (WF) vor, eine neuartige Methode, die in der Lage ist, die Modellgenauigkeit auf beliebigen Bildsätzen vorherzusagen, ohne dafür gelabelte Daten zu benötigen.

Im abschließenden Teil dieser Arbeit werden die Prinzipien der TTA auf Sprachmodelle (LMs) ausgeweitet, mit einem Fokus auf die Aufgabe der Literatur-Empfehlung. Wir schlagen einen Benchmark vor, der bewertet, wie gut LMs in der Lage sind, akademische Arbeiten basierend auf kurzen beschreibenden Zitaten zu erkennen. Unser Benchmark zeigt, dass LMs diese Aufgabe nicht effektiv bewältigen können. Daher präsentieren wir einen einfachen Agenten, der es LMs ermöglicht, relevante Arbeiten zu suchen und zu lesen, was ihre Leistung erheblich verbessert.

Contents

CHAPTER 1

Introduction

1.1 The Rise of Deep Learning

"The ability to adapt is critical to success."

GARRY KASPAROV

How Life Imitates Chess

The early 2010s saw a major change in Artificial Intelligence (AI), mainly driven by the widespread use and fast development of deep learning techniques LeCun *et al.* (2015); Schmidhuber (2015); Goodfellow *et al.* (2016). Notably, in specific cases, these models achieved or even surpassed human-level abilities on tough image recognition benchmarks, with ImageNet being a key example Krizhevsky *et al.* (2012a); Russakovsky *et al.* (2015); He *et al.* (2016).

This wave of success was not limited to image-related problems; significant progress was also reported in areas like natural language processing Devlin *et al.* (2019), speech recognition Hinton *et al.* (2012); Graves *et al.* (2013), and mastering complex strategy games Silver *et al.* (2016, 2017). Driven by these advances and the growing availability of computing power and large datasets Deng *et al.* (2009), deep learning systems rapidly began to move from controlled research labs into practical, real-world uses. Ambitious deployments were extensively explored in high-stakes areas, such as medical image analysis for diagnostic help Esteva *et al.* (2017); Litjens *et al.* (2017) and the creation of autonomous driving technology Bojarski *et al.* (2016); Grigorescu *et al.* (2020).

However, along with this impressive technological progress, a critical and widespread problem emerged: a fundamental lack of robustness when these complex models were used outside their training data distributions or in changing and unpredictable real-world situations Szegedy *et al.* (2013); Koh *et al.* (2021). This "brittleness" showed that even if models could get high accuracy on test data similar to their training data, their performance could drop significantly, and often without warning, when they encountered new inputs or small changes in the environment Hendrycks and Dietterich (2019b).

Early evidence for this weakness was seen in studies looking at the effect of dataset bias, which showed that models trained on specific data distributions often did not generalize well to other, even closely related, distributions Torralba and Efros (2011); Recht *et al.* (2019). The discovery and later detailed study of adversarial examples gave an even clearer example of this weakness: carefully made changes to inputs, often too small for humans to see, were shown to consistently cause top classifiers to make wrong predictions with high confidence Goodfellow *et al.* (2014); Madry *et al.* (2017); Carlini and Wagner (2017).

Figure 1.1: Image classifiers trained on ImageNet (52) **(left)** often struggle to correctly classify noisy images, like those sourced from ImageNet-C (95) **(right)**.

The underlying reasons for this brittleness are complex, but a leading idea focuses on "shortcut learning" Geirhos *et al.* (2020). This theory suggests that deep learning models, because of how they are trained on limited datasets, often learn to use misleading connections or shallow features that predict the right answer in the training data, but which do not represent a true or reliable understanding of the task Lapuschkin *et al.* (2019); Ilyas *et al.* (2019). Instead of learning the intended concepts, models may rely on unwanted biases or flaws present in the data Beery *et al.* (2018); Stock and Cisse (2018).

This challenge makes the concept of robustness a central focus of AI research. Robustness, in this context, refers to the ability of an AI system to keep working correctly and performing well when faced with various out-of-distribution inputs, common corruptions, perturbations, or other differences from the ideal conditions of its training Biggio *et al.* (2013); Hendrycks and Dietterich (2019b). Achieving such robustness is not just an academic goal but an essential need for the safe and reliable use of AI.

The consequences of this lack of robustness are especially serious in areas where system failures can have severe results. For example, in autonomous driving, not being able to reliably see the environment under different weather conditions, lighting changes, or with sensor noise can lead to disastrous outcomes Dai and Van Gool (2018); Michaelis *et al.* (2019); Volk *et al.* (2019). Similarly, in medical diagnosis, models that do not generalize well across different patient groups, imaging machines, or hospitals might give wrong assessments, potentially negatively affecting patient care Zech *et al.* (2018); AlBadawy *et al.* (2018); Wiens *et al.* (2019); Castro *et al.* (2020). Furthermore, a lack of robustness can make existing societal biases larger if models perform differently between demographic groups, highlighting the ethical need to develop more robust AI systems Buolamwini and Gebru (2018); Mehrabi *et al.* (2021); Barocas *et al.* (2023).

1.2 Why AI Fails Unexpectedly

The impressive performance of deep learning models, as noted in the previous section, is typically observed when training and testing data are drawn independently and identically distributed (IID) from the same underlying data generating process Vapnik (1991). However, this IID assumption frequently breaks down in real-world deployment scenarios, leading to a phenomenon broadly termed "distribution shift" Sugiyama *et al.* (2007). This shift occurs when the statistical properties of the data encountered by the model at test time differ significantly from the data it was trained on, and it stands as a primary reason for the observed fragility of many contemporary AI systems Torralba and Efros (2011); Recht *et al.* (2019).

One common and well-studied form of distribution shift is covariate shift, where the distribution of input features $P(x)$ changes between training and deployment, while the conditional distribution of labels given inputs $P(y|x)$ remains the same Shimodaira (2000). In computer vision, this can manifest as changes in lighting conditions Dai and Van Gool (2018), weather patterns Michaelis *et al.* (2019), camera sensor characteristics Hendrycks and Dietterich (2019b), or object pose or scale variations Alcorn *et al.* (2019). Models trained without explicit consideration for these variations often experience a substantial drop in performance when encountering such shifted inputs Hendrycks *et al.* (2020b).

Beyond simple input feature changes, models can also suffer from concept shift, where the relationship between inputs and outputs $P(y|x)$ itself changes over time or across contexts Gama *et al.* (2014); Webb *et al.* (2016). While less commonly addressed in standard image classification benchmarks, this is a critical concern in dynamic environments. More directly relevant to many image-based tasks is domain shift or source shift, where training data comes from one or more "source" domains (e.g., images from one hospital or one set of cameras) and the model is expected to perform on a "target" domain with different underlying data characteristics (e.g., images from a new hospital or a different geographical location) Zech *et al.* (2018); AlBadawy *et al.* (2018).

The vulnerability of deep neural networks to these shifts is increasingly attributed to their tendency to engage in "shortcut learning" Geirhos *et al.* (2020). Instead of learning the robust, causally relevant features that define a concept, models often exploit simpler, superficial, or spurious correlations present in the training data that are predictive of the labels within that specific dataset but do not generalize to new distributions Arjovsky *et al.* (2019); Ilyas *et al.* (2019). These shortcuts are often easier for the model to learn during the optimization process, given the high dimensionality of the input space and the nature of typical loss functions Neyshabur *et al.* (2017).

Numerous examples of such shortcut learning have been documented in the literature. For instance, models have been shown to heavily rely on texture statistics rather than object shape, leading to misclassifications when presented with objects having atypical textures Geirhos *et al.*

(2019a). Other studies have found models learning to classify objects based on their typical backgrounds or co-occurring objects rather than the primary object itself Beery *et al.* (2018); Xiao *et al.* (2020). The "tench and fingers" example, where models identify a fish by recognizing human fingers holding it (a common occurrence in ImageNet tench images), is another example of a spurious correlation being exploited Brendel and Bethge (2019). These shortcuts, while effective on IID test data, fail catastrophically under distribution shifts where these spurious cues are absent or misleading.

The very process of training deep neural networks, typically via Empirical Risk Minimization (ERM) using stochastic gradient descent, can inadvertently encourage the discovery of these shortcuts Vapnik (1991); Bottou and Bousquet (2007). ERM aims to minimize the average loss on the training data, and if a simple, non-robust feature offers a strong signal for this minimization, the model is likely to latch onto it, irrespective of its generalizability Sagawa *et al.* (2019). The high capacity of deep networks allows them to fit even complex, noisy patterns present in the training data, potentially leading to memorization of dataset-specific idiosyncrasies rather than abstraction of generalizable knowledge Zhang *et al.* (2017).

Consequently, simply collecting more training data, while often beneficial, cannot ensure robustness if the new data still carries the same biases or fails to capture the full range of real-world variations Torralba and Efros (2011). Similarly, while training-time data augmentation techniques (e.g., random crops, color jittering, or more advanced methods like AutoAugment Cubuk *et al.* (2019)) can improve robustness to certain anticipated variations Shorten and Khoshgoftaar (2019); Hendrycks *et al.* (2021a), they are limited by the designer's ability to foresee all relevant shifts and may not prepare the model for entirely unexpected changes. Furthermore, augmentations need to be carefully chosen as some can even harm generalization if they introduce unrealistic artifacts Geirhos *et al.* (2019a). This highlights the need for strategies that can address robustness beyond the standard training paradigm.

1.3 Towards Reliable AI: Adaptation and Evaluation

Given the pervasive challenges of distribution shift and the inherent limitations of models trained solely via ERM, it has become increasingly clear that static, pre-trained models are often insufficient for robust real-world deployment Hendrycks *et al.* (2020b); Koh *et al.* (2021). The dynamic nature of many operational environments necessitates AI systems that possess some capability to adjust or acclimate to new, unforeseen conditions encountered after their initial training phase Gama *et al.* (2014). This recognition has spurred research into various paradigms for developing more adaptive and resilient machine learning systems.

Broadly, the long-term vision for addressing these challenges includes concepts like online learning, where models continuously update their parameters as new data arrives Hoi *et al.* (2021), and continual or lifelong learning, where models aim to learn new tasks or adapt to new data distributions sequentially without catastrophically forgetting previously acquired knowledge Parisi *et al.* (2019); De Lange *et al.* (2021). While these represent ambitious and crucial research directions, they often require significant architectural changes, complex memory management strategies, or access to labeled data streams, which may not always be feasible, particularly for widely deployed, large-scale pre-trained models.

A more pragmatic and increasingly explored approach, especially for leveraging the power of existing state-of-the-art pre-trained classifiers, is Test-Time Adaptation (TTA) Sun *et al.* (2020b); Wang *et al.* (2020b). TTA methods aim to adapt a given pre-trained model during deployment (at test time) using only the incoming unlabeled test data itself. This is particularly appealing because it allows for performance improvements on out-of-distribution data without the need for retraining the entire model from scratch, accessing the original training data, or requiring additional labeled samples for the new domain Wang *et al.* (2020b).

The core idea behind many TTA techniques involves updating specific parts of the model, often just the normalization layers like Batch Normalization Ioffe and Szegedy (2015), based on an unsupervised loss function computed on the current batch of test inputs Wang *et al.* (2020b). For instance, some methods re-estimate batch statistics (mean and variance) Schneider *et al.* (2020); Nado *et al.* (2020), while others optimize unsupervised objectives like entropy on the test data Grandvalet and Bengio (2004); Wang *et al.* (2020b). The promise is that these small, targeted adjustments can help align the model's internal representations with the characteristics of the new test distribution.

While TTA offers a compelling route to improved robustness without retraining, the adaptation process itself is not without its challenges. Since adaptation typically relies on unsupervised objectives (e.g., minimizing prediction entropy or aligning batch statistics) and unlabeled test data, there is a risk that the model might 'adapt' in ways that are not beneficial or even detrimental in the long run. For instance, the model might overfit to transient patterns in the current batch of test data, leading to instability or error accumulation over time Niu *et al.* (2023). The very act of modifying model parameters based on incoming data necessitates careful consideration of how these changes compound and whether the adapted model remains reliable across diverse and potentially non-stationary data streams.

This underscores the paramount importance of rigorous and comprehensive evaluation methodologies for any adaptive AI system. If models are changing their parameters at test time, standard evaluation protocols that rely on static test sets may not fully capture their behavior or long-term viability. It becomes crucial to assess not only the immediate performance gain on a specific type of shift but also the stability of the adaptation process, its susceptibility to undesirable drift, and its behavior when faced with sequences of different shifts or even a return to the original training distribution.

Unfortunately, many existing benchmarks for evaluating robustness to distribution shift, while valuable for highlighting model weaknesses Hendrycks and Dietterich (2019b); Michaelis *et al.* (2019), were not primarily designed to assess the long-term dynamics of adaptive systems. They often consist of a fixed set of corruptions or domains, evaluated independently, and may not contain enough unique images to simulate prolonged deployment where a model continuously updates itself over many iterations without re-encountering the same data points. This limitation makes it difficult to confidently ascertain whether a TTA method that shows initial promise will maintain its benefits or eventually degrade.

Therefore, a critical step towards building truly reliable adaptive AI is the development of new benchmarking paradigms specifically designed to probe the long-term stability and potential failure modes of test-time adaptation methods. Such benchmarks need to provide a continuous stream of diverse, out-of-distribution data, allowing for repeated adaptation steps and the observation of performance trajectories over time. Building upon these critical insights and addressing the identified gaps, this thesis presents three works aimed at enhancing the reliability of AI systems when faced with real-world distribution shifts, with a particular focus on test-time adaptation and rigorous evaluation.

This thesis directly addresses these critical gaps by focusing on improving the robustness of machine learning models at test time, without resorting to costly retraining or requiring additional labeled data. Our approach is twofold: first, by designing more realistic and challenging benchmarks that better reflect the continuous and evolving nature of real-world data streams, we aim to provide a more robust platform for evaluating adaptive methods. Second, by analyzing the underlying mechanisms and failure modes of existing TTA techniques on these enhanced benchmarks, we seek to develop novel test-time strategies that offer more stable and reliable performance improvements, not just in computer vision but also extending these principles to other modalities like language understanding.

1.4 Thesis Contributions

This thesis presents three works aimed at enhancing the reliability of AI systems when faced with real-world distribution shifts, with a particular focus on test-time adaptation and rigorous evaluation. Each contribution identifies a critical challenge in current methodologies and proposes novel solutions, spanning both computer vision and natural language processing domains.

The first contribution, detailed in Chapter 2, addresses the critical need for more comprehensive benchmarks to evaluate the long-term stability of (TTA) methods. We argue that existing benchmarks often lack the scale and diversity required to simulate prolonged deployment scenarios. To this end, we introduce “Continuously Changing Corruptions” (CCC), a benchmark designed with significantly more unique images to facilitate the study of TTA methods over many adaptation steps. Using CCC, we demonstrate that while many contemporary TTA methods show initial performance gains, their efficacy can degrade significantly over time, sometimes falling below non-adapting baselines. Notably, this work reveals that a simple strategy involving periodic resets of the model to its original pre-trained weights can outperform more complex adaptive techniques in maintaining long-term robustness.

Building upon the insights into TTA behavior, the second contribution (Chapter 3) delves into the underlying mechanisms of common TTA strategies, specifically those employing entropy minimization. We investigate why such methods initially improve performance but may subsequently falter. Our analysis reveals that entropy minimization influences the clustering of image embeddings in the model's feature space. Initially, it tends to pull test image embeddings closer to those of same-class training images, improving accuracy. However, continued adaptation can cause these embeddings to drift away, causing performance degradation. Leveraging this understanding, we propose "Weighted Flips" (WF), a novel, label-free method to estimate a classifier's accuracy on unlabeled test data during adaptation, providing a practical tool for monitoring and potentially mitigating TTA instability in real-world deployments.

The final contribution, presented in Chapter 4, extends the principles of robust evaluation and test-time enhancement to the domain of language models (LMs). We identify a significant challenge in the ability of state-of-the-art LMs to accurately infer referenced academic papers from descriptive, unambiguous citations. This is a crucial task in the domain of literature recommendation and scholarly search. We introduce a new, challenging benchmark for this task and propose CiteAgent, an LM-based agent augmented with the ability to perform targeted search and retrieval of academic papers at test time. This approach demonstrates how test-time capabilities, without requiring model retraining, can substantially improve performance on complex reasoning and retrieval tasks in natural language processing, mirroring the test-time adaptation philosophy applied in computer vision.

In essence, this thesis advocates for a dual approach: first, the development of more realistic and demanding benchmarks to truly understand model limitations, and second, the creation of intelligent, low-overhead test-time strategies that enhance model reliability without the need for extensive retraining or additional labeled data. Through these contributions, we aim to advance the deployment of more dependable AI systems in complex, ever-changing environments.

CHAPTER 2

RDumb: A simple approach that questions our progress in continual test-time adaptation

Traditionally, neural network-based image classifiers are trained on a training set and tested on a test set, with both sets coming from the same group of images. What happens when a network is tested on images of animals taken on a clear day, but then tested on a set of images of animals taken during a snowstorm? In such instances, performance has been shown to decrease Hendrycks and Dietterich (2019b), when compared to performance on "clean", unperturbed images. A practical way to deal with these differences between images seen in training and those seen at test time is by continuously updating the model at test time, to help it adjust to the input images it receives. This is usually referred to as "Test-Time Adaptation" (TTA).

TTA can help pre-trained classification models adapt to their inputs at test-time, thereby greatly improving performance while costing very little and requiring no extra training. Though the variations of these techniques were widely explored in previous works, there were no benchmarks that adequately and thoroughly emulate realistic settings in which a model may see many images in a row.

In the setting explored in this chapter, a pre-trained image classification model updates its weights based on its inputs, without extra data or labels. Previous work, notably Wang *et al.* (2020b) showed how effective this can be in practice, but how reliable is it? If TTA, though changing a given model many times, can improve the performance of a model, could it also worsen the performance?

In this chapter, we build a comprehensive benchmark, CCC, which we use to evaluate how well continual learning methods perform in a realistic setting, when given many input images in a row. Our benchmarks shows a failure mode of all but one TTA methods tested. Additionally, we also propose a simple baseline method that achieves state-of-the-art performance on CCC as well as previous benchmarks. The material in this chapter was adapted from Press *et al.* (2023).

2.1 Introduction

Biological vision is remarkably robust at adapting to continually changing environments. Imagine cycling through the forest on a cloudy day and observing the world around you: You will encounter a wide variety of animals and objects, and be able to recognize them without effort. Even as the weather changes, rain sets in, or you start cycling faster, the human visual system effortlessly adapts and robustly estimates the surroundings Van de Ven and Tolias (2019). Equipping machine vision with similar capabilities is a long-standing and unsolved challenge, with numerous applications in autonomous driving, medical imaging, and quality control, to name a few.

Techniques for improving the robustness to domain shifts of ImageNet-scale Russakovsky *et al.* (2015) classification models include pre-training of large models on diverse and/or large-scale datasets Xie *et al.* (2020); Mahajan *et al.* (2018); Radford *et al.* (2021) and robustification of smaller models by specifically designed data augmentation Hendrycks *et al.* (2020b,a); Rusak *et al.* (2020a). While these techniques are applied during training time, recent work Schneider *et al.* (2020); Nado *et al.* (2020); Wang *et al.* (2020b); Rusak *et al.* (2021); Mummadi *et al.* (2021); Goyal *et al.* (2022); Wang *et al.* (2022); Niu *et al.* (2022) explored possibilities of further adapting models by Test-Time Adaptation (TTA). Such methods continuously update a given pretrained model exclusively using their input data, without having access to its labels. Test-time entropy minimization (Tent; Wang *et al.*, 2020b) has become a foundation for state-of-the-art TTA methods. Given an input stream of images, Tent updates a pretrained classification model

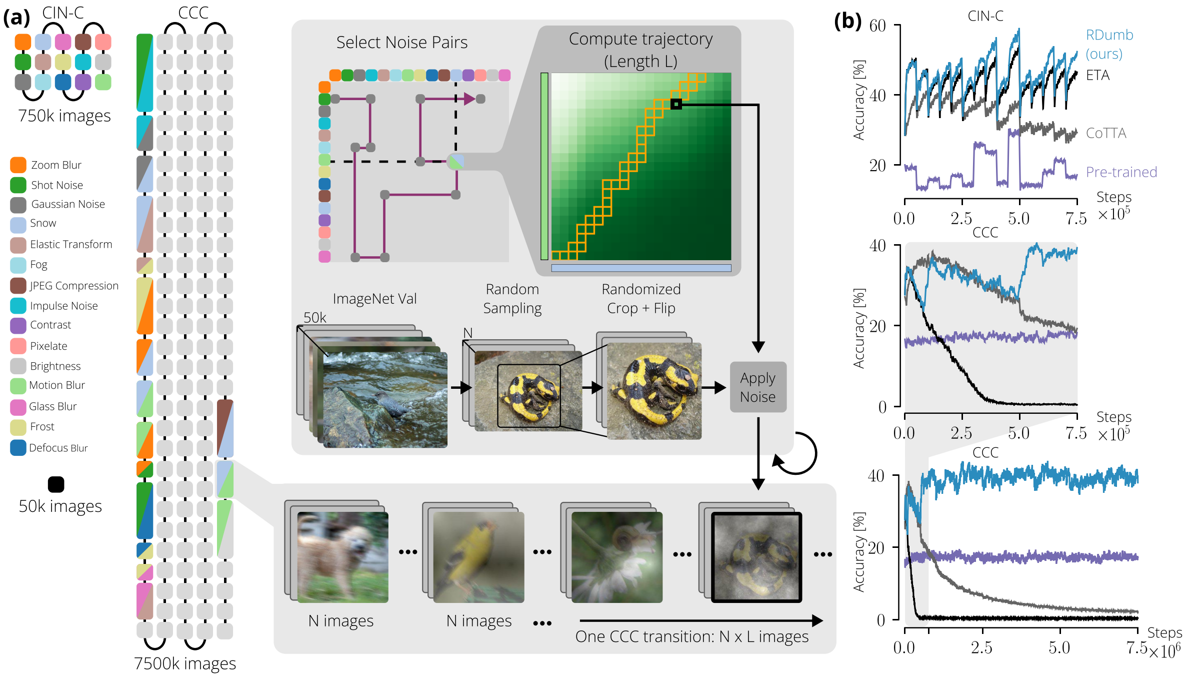


Figure 2.1: Continuously Changing Corruptions show limitations of existing TTA methods. (a) Comparison between ImageNet-Val, CIN-C and CCC. The proposed version of CCC is 10× longer than CIN-C and could naturally be extended even further without repeating images. CCC consists of sequences of smooth transitions from one ImageNet-C noise to another one. For each such pair, we construct a trajectory continuously interpolating from one pure noise to the other pure noise such that baseline accuracy is kept constant. For each point along the trajectory, we sample a batch of 1k, 2k, or 5k images from ImageNet-Val, randomly crop and flip it and apply the noise combination. (b) Due to its short length and high variability in difficulty, CIN-C (top) is unable to reveal the collapse of methods such as ETA and CoTTA, while CCC (middle and bottom) can.

by minimizing the entropy of its outputs, thereby continuously increasing the model's confidence in its predictions for every input image.

Previous TTA work Schneider *et al.* (2020); Nado *et al.* (2020); Wang *et al.* (2020b); Rusak *et al.* (2021); Mummadi *et al.* (2021); Goyal *et al.* (2022); Zhang *et al.* (2022) evaluate their models on ImageNet-C Hendrycks and Dietterich (2019b) or smaller scale image classification benchmarks LeCun *et al.* (2010); Krizhevsky *et al.* (2009). ImageNet-C consists of 75 copies of the ImageNet validation set, wherein each copy is corrupted according to 15 different noises at 5 different severity levels. When TTA models are evaluated on ImageNet-C, they are adapted on each noise and severity combination individually starting from their pretrained weights. Such a one-time adaptation approach is of little relevance when it comes to deploying TTA models in realistic scenarios. Instead, stable performance over a long run time after deployment is the desirable goal.

TTA methods are by design readily applicable to this setting and recently the field has started to move towards testing TTA models in continual adaptation settings Wang *et al.* (2022); Niu *et al.* (2022); Gong *et al.* (2022). Strikingly, this revealed that the dominant TTA approach Tent Wang *et al.* (2020b) decreases in accuracy over time, eventually being less accurate than a non-adapting, pretrained model Wang *et al.* (2022); Niu *et al.* (2022). In this work, we refer to any model whose classification accuracy falls below that of a non-adapting, pretrained model, as having "collapsed".

This collapsing behaviour of Tent shows that it cannot be used in continual adaptation over long time scales without modifications. While previous benchmarking of TTA methods already managed to reveal the collapse of Tent, our work shows that in fact all current TTA methods collapse sooner or later, *including methods with explicit built-in anti-collapse strategies*.

Since current benchmarks have not been sufficient to detect collapse in several models, we introduce an image classification benchmark designed to thoroughly evaluate TTA models for their long-term behavior. Our benchmark, *Continuously Changing Corruptions* (CCC), tests models on their ability to adapt to image corruptions that are constantly changing, much like when fog turns to rain or day turns to night. CCC allows us to easily control different factors that could affect the ability of a given method to continuously adapt: the corruptions and their order, the difficulty of the images themselves, and the speed at which corruptions transition. Most importantly, the length of our benchmark is ten times longer than that of previous benchmarks, and more diverse by including all kinds of combinations of corruptions (see Figure 2.1a). Using CCC, we discover that seven recently published state-of-the-art TTA methods are less accurate than a non-adapting, pretrained model. While Tent was already shown to collapse Wang *et al.* (2022); Niu *et al.* (2022); Gong *et al.* (2022), we show that this problem is not specific to Tent, and that many other methods – including specifically designed continual adaptation methods – collapse as well.

Finally, we propose "*RDumb*" [1] as a minimalist baseline mechanism that simply *Resets* the model to its pretrained weights at regular intervals. Previous work employs more sophisticated methods combining entropy minimization with various regularization approaches, yet we show that RDumb is superior on both existing benchmarks and ours (CCC). Our results call the progress made in continual TTA so far into question, and provide a richer set of benchmarks for realistic evaluation of future methods.

Our contributions are:

- We introduce the continual adaptation benchmark CCC. We show that previous benchmarks are too short to meaningfully assess long-term continual adaptation behaviour, and are too uncontrolled to assess the short-term learning dynamics.
- Using CCC, we show that the performances of all but one current TTA methods drop below a non-adapting, pre-trained baseline when trained over long timescales.
- We propose "*RDumb*" as a baseline and show that it outperforms all previous methods with a minimalist resetting strategy.

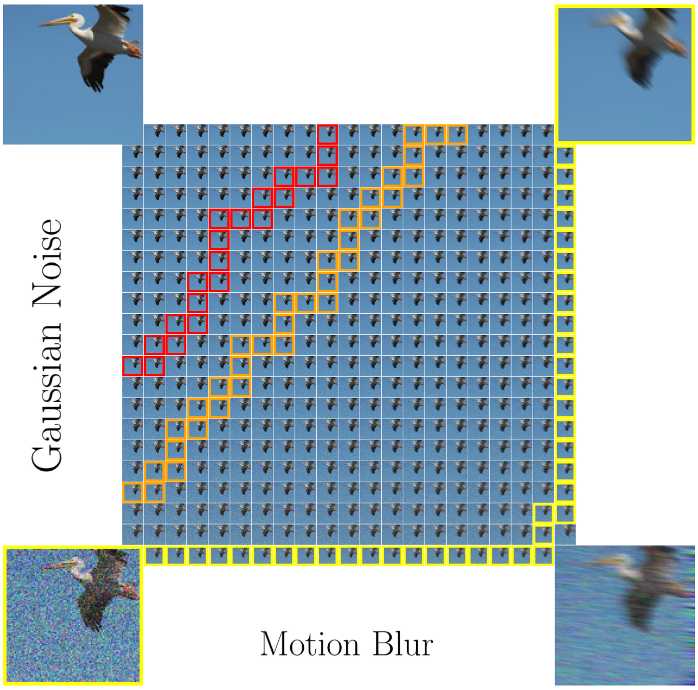
[1] The name was inspired by GDumb Prabhu *et al.* (2020).

2.2 CCC: Towards Infinite Testing with Continuously Changing Corruptions

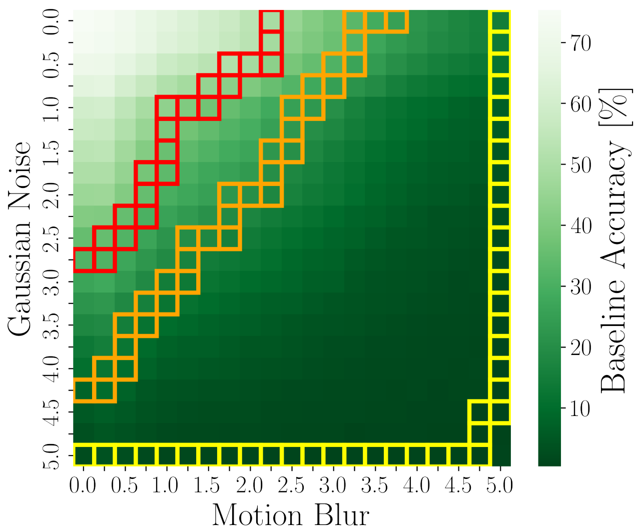
Until recently, it was common to evaluate TTA methods only on datasets on individual domain shifts such as the corruptions of ImageNet-C (Hendrycks and Dietterich, 2019b). However, the world is steadily changing and recently the community started moving towards continual adaptation, i.e., evaluating methods with respect to their ability to adapt to ongoing domain shifts (Wang *et al.*, 2022; Niu *et al.*, 2022; Sun *et al.*, 2019a).

The dominant method of evaluating continual adaptation on ImageNet scale is to concatenate the top severity datasets of the 15 ImageNet-C corruptions into one big dataset. We refer to the variant of this dataset introduced by Wang *et al.* (2022) as *Concatenated ImageNet-C* (CIN-C). CIN-C was used to demonstrate the collapse of Tent and the stability of recent TTA methods by Wang *et al.* (2022); Niu *et al.* (2022); Gong *et al.* (2022).

In Figure 2.1b, we evaluate a range of TTA methods on CIN-C and notice three potential problems: Firstly, ETA Niu *et al.* (2022) appears to be stable and better than a non-adapting, pretrained baseline, but is revealed to collapse when tested on CCC. Additionally, while CoTTAWang *et al.* (2022) clearly goes down in performance, it is not yet clear whether it collapses or stabilizes above or below baseline performance. Fundamentally, CIN-C turns out to be too short to yield reliable, conclusive results. Secondly, assessing adaptation dynamics is further obscured by the considerable variations of the baseline performance among the different corruptions in CIN-C. This is not only a factor that affects adaptation itself (shown in Wang *et al.* (2020b); Niu *et al.* (2022)), it also leads to substantial fluctuations in performance across multiple runs, making it difficult to obtain a clear and reliable assessment. Finally, CIN-C features exclusively abrupt transitions between different corruption types. In contrast, in the real world, domain changes may often be smooth and subtle with varying speeds: day to night, rain to sunshine, or the accumulation of dust on a camera. Therefore, it is important to also probe TTA methods on continual domain changes that are not tied to a specific point in time and thus constitute a relevant test for stable continual adaptation.



(a)



(b)

Figure 2.2: (a) Each corruption of CCC consists of applying two ImageNet-C corruptions at different severities. We extend the individual severities to be more fine-grained than in ImageNet-C, allowing for smoother noise changes, and exponentially more (noise, severity) combinations. The corners are enlarged for easier viewing, zoom in for greater detail. (b) Sample dataset sequences with a constant baseline accuracy. The sequences start from the left where Motion Blur is zeroed out, and end at the top with Gaussian noise zeroed out. The colors red, orange, and yellow correspond to trajectories in CCC-Easy, CCC-Medium and CCC-Hard, respectively.

Here we propose a new benchmark, *Continuously Changing Corruptions* (CCC), to address these issues. CCC solves the issues of benchmark length, uncontrolled baseline difficulty, and transition smoothness in a simple and effective manner. Firstly, the length issue is remedied because the individual runs of CCC are constructed by a generation process which can generate very long datasets without reusing images. In this work we use runs of 7.5M images, which is 10 times as long as CIN-C. If required to compare methods in future work (where collapse is even slower), it is straightforward to generate even longer benchmarks within the CCC framework. Secondly, since both Wang *et al.* (2020b); Niu *et al.* (2022) have shown that dataset difficulty is a confounder when studying adaptation, the difficulty of individual benchmark runs is kept stable. Additionally, we examine three different difficulty levels to ensure a comprehensive yet controlled evaluation. Finally, CCC exhibits smooth domain shifts: it applies two corruptions to each image. Over time, the severity of one corruption is smoothly increased while the severity of the other is decreased, maintaining the desired difficulty. We also study three different speeds for applying this process. We will now outline the generation procedure of the dataset.

Continuously changing image corruptions To allow smooth transitions between corruptions, we introduce a more fine-grained severity level system to the ImageNet-C dataset. We interpolate the parameters of the original corruptions (integer-valued severities from 1 to 5) to finer grained severity levels from 0 to 5 in steps of 0.25. We apply two different ImageNet-C corruptions to each image, such that we can decrease the severity of one corruption while increasing the severity of another one. Hence, the corruptions of CCC are given by quadruples (c_1, s_1, c_2, s_2), where c_1 and c_2 are ImageNet-C corruption types and s_1 and s_2 are severity levels. When applying such a corruption, we first apply c_1 and then c_2 at their respective severities (see Figure 2.2a).

Calibration to desired baseline accuracy In order to control baseline accuracy, we need to know how difficult each combination of 2 noises and their respective severities is. To that end, we first select a subset of 5,000 images from the ImageNet validation set. For each corruption (c_1, s_1, c_2, s_2), we corrupt all 5,000 images accordingly and evaluate the resulting images with a pre-trained ResNet-50 He *et al.* (2016). The resulting accuracy is what we refer to as *baseline* *accuracy* and what we use for controlling difficulty. In total, we evaluate more than 463 million corrupted images. Previous work, Hendrycks and Dietterich (2019a), measures normalized accuracy using AlexNet Krizhevsky *et al.* (2012b), which is less pertinent in present-day contexts. In addition, the accuracy of non-adapting Vision Transformers are stable on CCC as well (Figure 2.4).

Generating Benchmark Runs Having calibrated the corruptions pairs, we prepare benchmark runs with different baseline accuracies, transition speeds, and noise orderings. We pick 3 different baseline accuracies: 34%, 17%, and 2% (CCC-Easy, CCC-Medium, CCC-Hard respectively). For each one of the difficulties, we select a further 3 transition speeds: 1k, 2k, 5k. Lastly, for each difficulty and transition speed combination we use 3 different noise orderings, determined by 3 random seeds. To generate each run, we first select the initial corruption at the severity which according to our calibration is closest to the desired baseline accuracy. We then transition to the second corruption of the noise ordering by repeatedly either decreasing the severity of the first noise by 0.25 or increasing the severity of the second noise by 0.25 such that the baseline accuracy is as close to the target as possible (see Figure 2.2). In each step along each path, we sample 1k, 2k, or 5k images from the ImageNet validation set depending on the desired transition speed. Each image is randomly cropped and flipped for increasing the diversity of the dataset, and then corrupted.

Once the path from the initial to the second corruption is finished, the process is repeated for transitioning to the third corruption and so on (for more details see Appendix A.2). In the end, we have 3 difficulties consisting of 9 benchmark runs each. CCC-Medium at a speed of 2k corresponds roughly to CIN-C's difficulty and transition speed.

2.3 RDumb: Turning your model off and on again

Continual test-time adaptation needs to successfully adapt models over arbitrarily long timescales during deployment. Resetting a model to its initial weights at fixed intervals fulfills this criterion by design, yet allows to benefit from adaptation over short time scales. Surprisingly, such an approach has never been tried before (see Appendix A.5 for more discussion).

Regarding the choice of the adaptation loss, we build on the weighted entropy used in ETA Niu *et al.* (2022). For a stream of input images $\mathbf{x}_1, \mathbf{x}_2, \ldots$, we compute class probabilities $\mathbf{y}_t = f_{\Theta_t}(\mathbf{x}_t)$ and optimize the loss function

$$L(\mathbf{y}_t; \overline{\mathbf{y}}_{t-1}) = \left(\frac{1[(|\cos(\mathbf{y}_t, \overline{\mathbf{y}}_{t-1})| < \varepsilon) \wedge (H(\mathbf{y}_t) < H_0)]}{\exp(H(\mathbf{y}_t) - H_0)} \right) H(\mathbf{y}_t) \tag{2.1}$$

which weights the entropy $H(\mathbf{y}_t) = -\mathbf{y}_t^\top(\log \mathbf{y}_t)$ of each prediction using the similarity to averaged previously predicted class probabilities, $\overline{\mathbf{y}}_t = (\mathbf{y}_1 + \cdots + \mathbf{y}_t)/t$, and a comparison to a fixed entropy threshold H_0. $\cos(\mathbf{u}, \mathbf{v})$ refers to the cosine similarity between vectors $\mathbf{u}$ and $\mathbf{v}$. At each step, (part of) the weights Θ_t are updated using the Adam optimizer Kingma and Ba (2014). At every T-th step, Θ_t is reset to the baseline weights Θ_0. We use $\varepsilon = 0.05$ and $H_0 = 0.4 \times \ln 10^3$ following Niu *et al.* (2022), and select $T = 1000$ based on the holdout noises in IN-C (see Section 2.6).

2.4 Experiment Setup

We benchmark RDumb alongside a range of recently published TTA models. For all models, we use a batch size of 64. In all models, the BatchNorm statistics are estimated on the fly, and the affine shift and scale parameters are optimized according to a model-specific strategy outlined below.

- **BatchNorm (BN) Adaptation** Schneider *et al.* (2020); Nado *et al.* (2020) estimates the BatchNorm statistics (mean and variance) separately for each batch at test time. The affine transformation parameters are not adapted.

- **Tent** Wang *et al.* (2020a) optimizes the entropy objective on the test set in order to update the scale and shift parameters of BatchNorm (in addition to learning the statistics).

- **Robust Pseudo-Labeling (RPL)** Rusak *et al.* (2021) uses a teacher-student approach in combination with a label noise resistant loss.

- **Conjugate Pseudo Labels (CPL)** Goyal *et al.* (2022) use meta learning to learn the optimal adaptation objective function across a class of possible functions.

- **Soft Likelihood Ratio (SLR)** Mummadi *et al.* (2021) uses a loss function that is similar to entropy, but without vanishing gradients. *Anti-Collapse Mechanism:* An additional loss is used to encourage the model to have uniform predictions over the classes, and the last layer of the network is kept frozen.

- **Continual Test Time Adaptation (CoTTA)** Wang *et al.* (2022) uses a teacher student approach in combination with augmentations. *Anti-Collapse Mechanism:* Every iteration, 0.1% of the weights are reset back to their pretrained values.

- **Efficient Test Time Adaptation (EATA)** Niu *et al.* (2022) uses 2 weighing functions to weigh its outputs: the first based on their entropy (lower entropy outputs get a higher weight), the second based on diversity (outputs that are similar to seen before outputs are excluded). *Anti-Collapse Mechanism:* An L_2 regularizer loss is used to encourage the model's weights to stay close to their initial values.

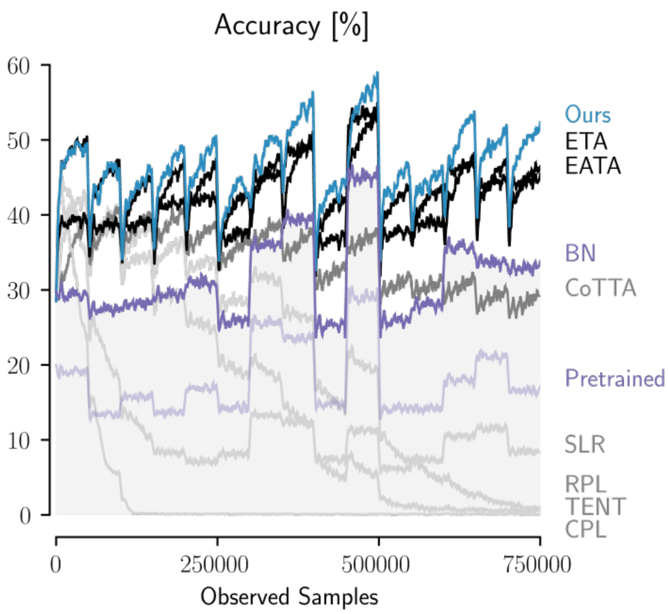
- **EATA Without Weight Regularization (ETA)** For completeness, we also test ETA, which is EATA but without the regularizer loss, proposed in Niu *et al.* (2022).

- **RDumb** is our proposed baseline to mitigate collapse via resetting. We reset every $T = 1,000$ steps, as determined by a hyperparameter search on the holdout set (Section 2.6).

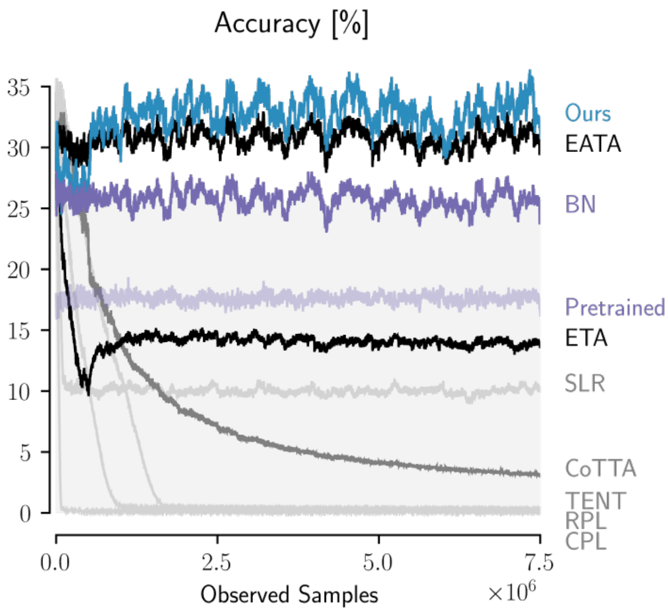
Following the original implementations, Tent, ETA, EATA, and RDumb use SGD with a learning rate of $2.5 \cdot 10^{-5}$. RPL uses SGD with a learning rate of $5 \cdot 10^{-3}$. SLR uses the Adam optimizer with a learning rate of $6 \cdot 10^{-4}$. CoTTA uses SGD with a learning rate of 0.01, and CPL uses SGD with a learning rate of 0.001.

2.5 Results

CCC reveals collapse during continual adaptation, unlike CIN-C. For three models that were evaluated, evaluation on CIN-C yielded inconclusive or inaccurate results in detecting collapse: CoTTA collapses on CCC, while CIN-C shows it to be on a downward trend, but end performance still outperformed the baseline (Figure 2.3a). Additionally, ETA shows no signs



(a)



(b)

Figure 2.3: Adaptation performance of all evaluated models depending on the number of observed samples so far. (a) CIN-C. Model performances are averaged over the 10 runs of the benchmark. (b) CCC. Model performances are averaged over the 27 runs of the three difficulty levels. See Appendix A.3, Figure A.4 for separate plots for CCC Easy, Medium, and Hard.

Table 2.1: Mean accuracy of ResNet-50 models on CIN-C, CIN-3DCC and CCC. For each CCC split (Easy, Medium, and Hard), a mean of 9 runs is taken. For the CIN-C and CIN-3DCC experiments, the accuracy reported is the mean of 10 different noise permutations. Grey indicates collapse.

Adaptation method	CIN-C	CIN-3DCC	CCC-Easy	CCC-Medium	CCC-Hard	Average
Pretrained 92	18.0 $\pm$ 0.0	31.5 $\pm$ 0.0	34.1 $\pm$ 0.22	17.3 $\pm$ 0.21	1.5 $\pm$ 0.02	20.5
BN 200; 162	31.5 $\pm$ 0.02	35.7 $\pm$ 0.02	42.6 $\pm$ 0.39	27.9 $\pm$ 0.74	6.8 $\pm$ 0.31	28.9
Tent 238	15.6 $\pm$ 3.5	24.4 $\pm$ 3.5	3.9 $\pm$ 0.58	1.4 $\pm$0.17	0.51 $\pm$ 0.07	9.2
RPL 192	21.8 $\pm$ 3.6	30.0 $\pm$ 3.6	7.5 $\pm$ 0.83	2.7 $\pm$ 0.36	0.67 $\pm$ 0.14	12.5
SLR 159	12.4 $\pm$ 7.7	12.2 $\pm$ 7.7	22.2 $\pm$ 18.4	7.7 $\pm$ 9.0	0.66 $\pm$ 0.57	11.0
CPL 80	3.0 $\pm$ 3.3	5.7 $\pm$ 3.3	0.41 $\pm$ 0.06	0.22 $\pm$ 0.03	0.14 $\pm$ 0.01	1.9
CoTTA 240	34.0 $\pm$ 0.68	37.6 $\pm$ 0.68	14.9 $\pm$ 0.88	7.7 $\pm$ 0.43	1.1 $\pm$ 0.16	19.1
EATA 165	41.8 $\pm$ 0.98	43.6 $\pm$ 0.98	48.2 $\pm$ 0.6	35.4 $\pm$1.0	8.7 $\pm$0.8	35.5
ETA 165	43.8 $\pm$ 0.33	42.7 $\pm$ 0.33	41.4 $\pm$ 0.95	1.1 $\pm$ 0.43	0.23 $\pm$ 0.05	25.8
RDumb (ours)	**46.5 $\pm$ 0.15**	**45.2 $\pm$ 0.15**	**49.3 $\pm$ 0.88**	**38.9 $\pm$ 1.4**	**9.6 $\pm$ 1.6**	**37.9**

of collapse on CIN-C, while collapsing very clearly on CCC (Figure 2.3b, more precisely on CCC-Medium and CCC-Hard, see Appendix Figure A.4). When tested using ViT backbones, EATA is better than the pretrained model on CIN-C (Figure 2.4a), but worse than the pretrained model on CCC (Figure 2.4b, Table 2.2,2.3). Lastly, SLR on CIN-C appears to be somewhat stable, but only at around 10% accuracy. CCC reveals this to be only partly true: on CCC-Hard, SLR is not stable and collapses to nearly chance accuracy. In summary, models evaluated on CCC show clear limits, which are impossible to see on CIN-C because of the high difficulty variance between runs, and its short length.

RDumb is a strong baseline for continual adaptation. RDumb outperforms all previous methods on both established benchmarks (CIN-C, CIN-3DCC) as well as our continual adaptation benchmark, CCC (Table 2.1 and Figure 2.3). Concretely, we outperform EATA and increase accuracy by more than 11% on CIN-C (improving from 41.8% points to 46.5% points), and by almost 7% when averaged all evaluation datasets. While not able to outperform RDumb, we note that EATA is also a strong method for preventing collapse except for the counterexample in Table 2.2.

The results transfer to Imagenet-3D Common Corruptions. To further demonstrate the effectiveness of our method, we show results on Imagenet-3DCC Kar *et al.* (2022), which features 12 types of corruptions, which take the geometry and distances between objects into account when applied to an image. Similarly to CIN-C, we test our models on 10 different

Table 2.2: Mean accuracy of different backbone architectures on CCC-Medium. Accuracy reported is an average across 9 runs. Backbones used: He *et al.* (2016); Xie *et al.* (2017); Dosovitskiy *et al.* (2020); Tu *et al.* (2022); Liu *et al.* (2022), †: AugMix Hendrycks *et al.* (2020a), ‡: DeepAugment Hendrycks *et al.* (2020b). Grey indicates collapse.

Method	RN18	RN50	RN50†	RN50 †‡	RNXt101 †‡	ViT-B16	MaxViT-T	SwinViT-T
Pretrained	12.2	17.3	27.9	38.9	47.8	42.0	45.1	33.2
EATA	26.8	35.4	46.5	**52.3**	**58.5**	38.5	47.1	35.6
RDumb	**32.5**	**38.9**	**47.0**	51.9	58.4	**50.2**	**49.9**	**36.5**

permutations of concatenations of all the noises of IN-3DCC, which we call CIN-3DCC. As in the case of CIN-C and CCC, RDumb outperforms all previous methods (Table 2.1).

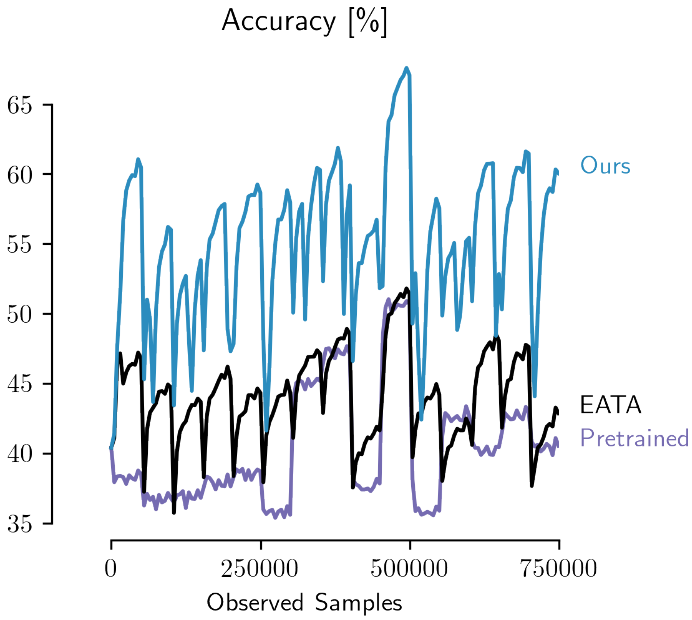
The results transfer to Vision Transformers. To further validate our claims, we test both EATA and our method with a Vision Transformer (ViT, Dosovitskiy *et al.*, 2020) backbone (Figure 2.4). The difference in average accuracy between our method and EATA is larger when using a ViT, as compared to a ResNet-50: on CIN-C and CCC the gap is 10.9% points and 11.7% points respectively. Additionally, EATA's accuracy on CCC is below that of a pretrained, non-adapting model[2]. This collapse can only be seen by using CCC, and not when evaluating on CIN-C.

Table 2.3: Mean accuracy of different backbone architectures on CCC-Hard. Accuracy reported is an average across 9 runs. Backbones used: He *et al.* (2016); Xie *et al.* (2017); Dosovitskiy *et al.* (2020); Tu *et al.* (2022); Liu *et al.* (2022), †: AugMix Hendrycks *et al.* (2020a), ‡: DeepAugment Hendrycks *et al.* (2020b). Grey indicates collapse.

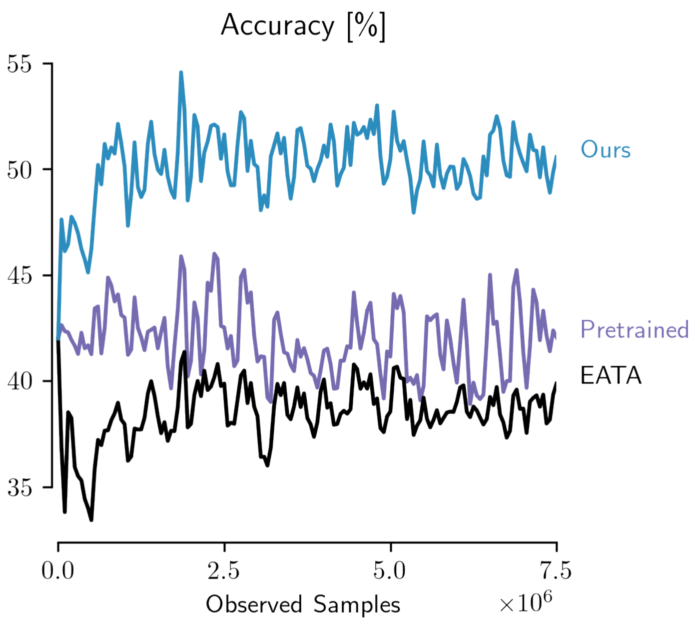
Method	RN18	RN34	RN50	RN50†	RN50†‡	RNXt101†‡	ViT-B16	MaxViT-T	SwinViT-T
Pretrained	0.82	1.3	1.5	5.6	24.3	15.6	**22.0**	**22.0**	**9.3**
EATA	6.4	7.6	8.7	**17.7**	**30.3**	**36.3**	8.6	15.4	8.6
RDumb	**8.3**	**10.7**	**9.6**	14.7	29.9	35.6	**23.8**	**22.0**	8.0

RDumb allows adaptation of a variety of architectures without tuning. We evaluate RDumb and EATA across a range of popular backbone architectures. Out of the nine architectures evaluated (see Table 2.2,2.3), RDumb outperformed EATA by an average margin of 4.5%

[2]Increasing the regularizer parameter value does not help stabilize the model, see Appendix A.4.



(a)



(b)

Figure 2.4: TTA using a ViT backbone: (a) On CIN-C, EATA is better than the pretrained baseline (44.4% points vs 40.1% points). (b) On CCC-Medium, EATA is worse than the pretrained baseline (38.5% points vs 42.0% points). RDumb (ours) is consistently better than both EATA and the baseline.

points on seven of them, and worse by an average margin of only 0.25% points on the remaining two.

2.6 Analysis and Ablations

Optimal reset intervals. To determine the optimal reset interval, we run ETA with reset intervals $T \in [125, 250, 500, 1000, 1500, 2000]$ on CIN-C using the IN-C holdout noises. We concatenate the 4 holdout noises at severity 5 as our base test set. This base test set is repeated until the model sees 750k images, which is equal to the length of CIN-C. We do this for every permutation of the 4 holdout corruptions. On this holdout set, we find that the optimal T is equal to 1,000.

Table 2.4: Accuracy of our method for different resetting times on CIN-C-Holdout

T (steps)	125	250	500	1000	1500	2000
Acc. [%]	42.1	44.4	46.0	**46.7**	46.5	46.4

RDumb is less sensitive to hyperparameters. An added benefit to our method is that it is less sensitive to hyperparameters than EATA. We conduct a simple hyperparameter search of the E_0 parameter—the hyperparameter that controls how many outputs get filtered out because of their high entropy. Our method consistently outperforms EATA across every hyperparameter tested (Table 2.5), and for the highest value, 0.7, EATA collapses to almost chance accuracy on all splits, while our method does not. In addition, RDumb's performance benefits from finetuning ($H_0 = \{0.2, 0.3\}$), while EATA is not able to improve.

Table 2.5: Average accuracy on all of CCC splits on a variety of H_0 values. For all other experiments in this paper we use $H_0 = 0.4 \times \ln 10^3$, as in Niu *et al.* (2022).

$H_0 \times \ln 10^3$	0.1	0.2	0.3	0.4	0.5	0.6	0.7
EATA	27.8	27.9	29.9	**30.8**	28.7	28.0	0.33
RDumb (ours)	31.6	32.9	**33.1**	32.6	30.7	25.7	16.8

RDumb is effective because ETA reaches maximum adaptation performance fast. ETA is quick to adapt to new noises from scratch. On each of the holdout set noises and severities, ETA reaches its maximum accuracy after seeing only about 12,500 samples, which is about 200 adaptation steps (Figure 2.5a). After that, accuracy decays at a pace slower than its initial increase. Therefore, when resetting and readapting from scratch, only a few steps with substantially suboptimal predictions are encountered before performance is again close to optimal.

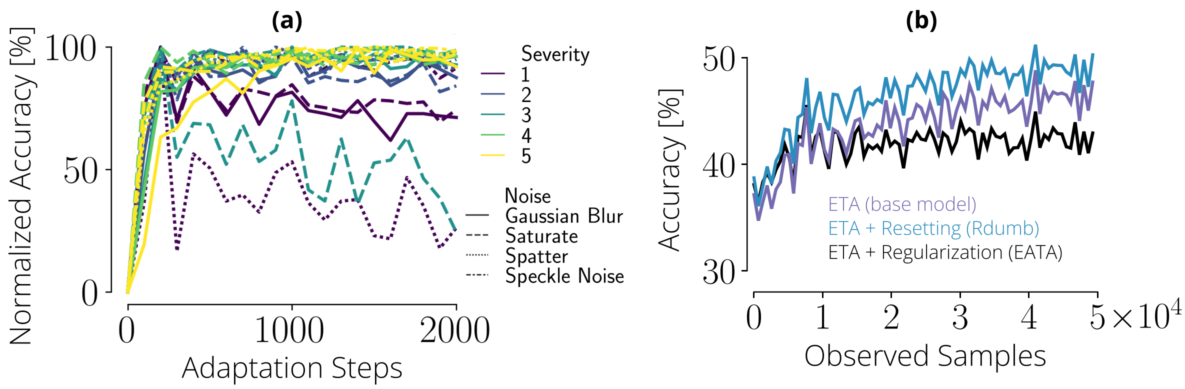


Figure 2.5: (a) ETA's normalized accuracy over time, on the ImageNet-C holdout noises and each of their severities. For every noise in the holdout set, ETA reaches its maximum accuracy very quickly. (b) Rdumb shares ETA's property of fast adaptation, while regularization in EATA slows adaptation.

Comparing resetting to regularization. Previous works typically optimize two loss terms: one loss encourages adaptation on unseen data, another loss regularizes the model to prevent collapse. Having to optimize two losses should be harder than optimizing just one – we see evidence for this in short term adaptation on CIN-C (Figure 2.5b): ETA and EATA optimize the same loss, but EATA additionally optimizes an anti-collapse loss. Consequently, ETA beats EATA by 2% points on CIN-C.

Collapse Analysis. We now investigate potential causes and effects of the observed collapse behavior. We propose a theoretical model, fully specified in Appendix A.1, which can explain both collapsing and non-collapsing runs. The model consists of a batch norm layer followed by a linear layer trained with the Tent objective. Within this model, we can present two scenarios. In the first, the model successfully adapts and plateaus at high accuracy (Figure A.1a). In the second, we see early adaptation which is then followed by collapse (Figure 2.6a,A.1a). The properties of noise in the data influence whether we observe the case of successful or unsuccessful adaptation.

Interestingly, the model predicts that the magnitude of weights increases over the course of optimization; this signature of entropy minimization can be found in both the theoretical model and a real experiment using RDumb without resetting on CCC-Medium (Figure 2.6). Unfortunately, weight explosion happens only *after* model performance is already collapsed (Figure A.2b). The effect is observable across all layers (Figure 2.6c,A.2).

2.7 Discussion and Related Work

Domain Adaptation. In practice, the data distribution at deployment is different from training, and hence the task of *domain adaptation*, i.e., the task of adapting models to different target distributions has received a lot of attention Geirhos *et al.* (2018b); Rusak *et al.* (2020a); Hendrycks *et al.* (2020a); Lee (2013); Sun *et al.* (2019a); Wang *et al.* (2020b); Liang *et al.* (2021). The methods on domain adaptation split into different categories based on what information is assumed to be available during adaptation. While some methods assume access to labeled data for the target distribution Motiian *et al.* (2017); Yue *et al.* (2021), *unsupervised domain adaptation* methods assume that the model has access to labeled source data and unlabeled target data at adaptation time Lee (2013); Liang *et al.* (2021); Ganin *et al.* (2016); Sun *et al.* (2019b). Most useful for practical applications is the case of *test-time adaptation*, where the task is to adapt to the target data on the fly, without having access to the full target distribution, or the original training distribution Schneider *et al.* (2020); Nado *et al.* (2020); Wang *et al.* (2020b); Rusak *et al.* (2021); Niu *et al.* (2022); Sun *et al.* (2019a); Zhang *et al.* (2022).

In addition to the division made above, one can further distinguish what assumptions are made about how the target domain is changing. Many academic benchmarks focus on one-time distribution shifts. However, in practical applications, the target distribution can easily change perpetually over time, e.g., due to changing weather and lightness conditions, or due to sensor corruptions. Therefore, the latter setting of *continual adaptation* has been receiving increasing attention recently. The earliest example of adapting a classifier to an evolving target domain that we are aware of is Hoffman *et al.* (2014), which learn a series of transformations to keep the data representation similar over time. Wulfmeier *et al.* (2018); Ganin *et al.* (2016); Tzeng *et al.*

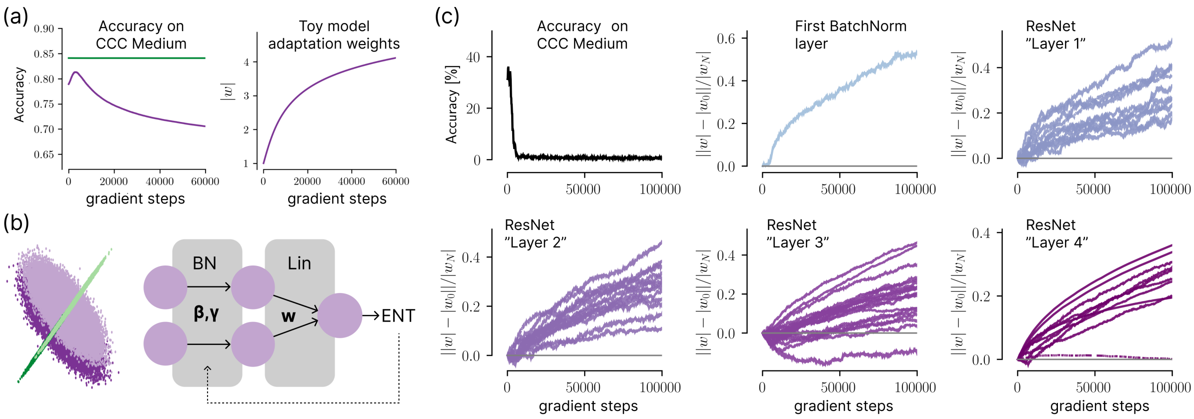


Figure 2.6: Analysis of entropy minimization collapse on real data. Learning dynamics in terms of accuracy and weight magnitude are shown in (a) for a two layer toy model consisting of batch norm and linear layer (b). Consistent with the theoretical analysis, we find that the adaptation weights in all layers increase over time continually (c), even long after the collapse as indicated by Accuracy on CCC-Medium has happened. Refer to Figure A.1–A.2 and Appendix A.1 for additional properties of the toy model (a–b) and a zoomed-in view on (c).

(2017) use an adversarial domain adaptation approach for this. Bobu *et al.* (2018) pointed out that two of these approaches can be prone to catastrophic forgetting Ratcliff (1990). To deal with this, different solutions have been proposed Bobu *et al.* (2018); Liu *et al.* (2020); Abnar *et al.* (2021); Niu *et al.* (2022); Wang *et al.* (2022); Mummadi *et al.* (2021).

Test-time adaptation methods have classically been applied in the setting of one-time domain change, but can be readily applied in the setting of continual adaptation, and some recent methods have been explicitly designed and tested with continual adaptation in mind Wang *et al.* (2020b, 2022); Niu *et al.* (2022). Because TTA methods use only test data and don't alter the training procedure, they are particularly easy to apply and have been shown to be superior to other domain adaptation approaches Schneider *et al.* (2020); Nado *et al.* (2020); Geirhos *et al.* (2018b); Rusak *et al.* (2020a), Therefore, we focus only on TTA methods, which we discussed in more detail in Section 2.4.

Continual Adaptation Benchmarks. While continually changing datasets are used in the continual learning literature, e.g. Lomonaco and Maltoni (2017); Shi *et al.* (2020); Feng *et al.* (2019); Han *et al.* (2021a); Sun *et al.* (2020a); Yu *et al.* (2020), they have been used in TTA benchmarks only very recently. In contrast to all previous benchmarks, we want to evaluate how continual adaptation methods change over long periods of time, when the noise changes in a continuous manner. The longest datasets for TTA were made up of hundreds of thousands of labeled images in total, while we adapt to 7.5M images per run. Other datasets are comprised of short video clips Sun *et al.* (2020a); Shi *et al.* (2020); Lomonaco and Maltoni (2017) 10-20 seconds in length. Besides maximizing its length, we set out to create a dataset that is well calibrated and closely related to the well-known ImageNet-C dataset. Additionally, with our noise synthesis, we can guarantee a wide variety of noises in each one of our evaluation runs, we can control the speed at which the noise changes, and we can control the difficulty of the generated noise. Lastly, CCC accounts for different adaptation speeds, as demonstrated by Rusak *et al.* (2021) and Mummadi *et al.* (2021). They showed that training their methods on ImageNet-C for more than one epoch leads to better performance.

2.8 Conclusion

TTA techniques are increasingly applied to continual adaptation settings. Yet, we show that all current TTA techniques collapse in some continual adaptation settings, eventually performing worse than even non-adapting models. And while some methods are stable in some situations, they are still outperformed by our simplistic baseline "RDumb", which avoids collapse by resetting the model to its pretrained state periodically. These observations were made possible by our new benchmark for continual adaptation (CCC), which was carefully designed for the precise assessment of long and short term adaptation behaviour of TTA methods and we envision it to be a helpful tool for the development of new, more stable adaptation methods.

CHAPTER 3

The Entropy Enigma: Success and Failure of Entropy Minimization

In the previous chapter, we looked at methods that take pre-trained classifiers and adapt them to their inputs at test-time, without having access to labels. For the majority of the methods analyzed, adaptation is carried out by minimizing the classifier's entropy on its inputs. We showed that entropy minimization improves performance at first, but then fails (which is why simply resetting the classifier to its pre-trained state can serve as a solid baseline). Still, the question of why this process eventually fails remained unanswered.

This chapter began with a key observation: There seems to be a link between how quickly a classifier "collapses" when minimizing entropy and how accurate it is on the images it adapts to. In short, if the classifier's accuracy on a dataset starts out low, entropy minimization might boost its accuracy by a bit, after which it drives it down near zero. If the classifier begins with higher accuracy, the boost is larger and more adaptation steps have to be performed for the accuracy to drop to near zero.

Wanting to find out why this happens, I reached out to Ravid. We tried to see if the loss function or specific inputs were causing adaptation for some images and accuracy loss for others. After we got stuck, Ravid spoke with Yann, who suggested that it had something to do with clustering. That insight set us on the right track.

In this chapter, we dive into this mechanism to explain why it works the way it does and how it leads to the failure modes discussed in Chapter 2. Our analysis shows that entropy minimization in essence pulls the embeddings of input images closer together in the embedding space. Early on, that helps bring the embeddings of input images closer to similar images seen in training. After enough images, the process pushes the new input images too far from where the training images lie, causing accuracy to drop.

This analysis not only helps us understand many state-of-the-art methods that adapt classifiers to their inputs, but it also has a practical benefit: by watching how these embedding clusters change during adaptation, we can estimate the accuracy of a classifier on any dataset, without using labels. We propose the Weighted Flips (WF) method to do just that, and show it to be effective on a wide array of image datasets.

The material in this chapter is adapted from Press *et al.* (2024b).

3.1 Introduction

Practitioners commonly employ model adaptation strategies to enhance classifier performance on real-world data, which often differs significantly from training data. Unsupervised losses play a crucial role in adapting models to images corrupted by noise, such as snow or motion blur, or images from domains not seen in training, such as paintings or computer rendered images. Entropy minimization (EM) is a Test Time Adaptation (TTA) method that can improve the accuracy of a model on new datasets, without the need for additional labeled training data. EM adapts classifiers by iteratively increasing the probabilities assigned to the most likely classes while diminishing those of the others, and is an integral part of many recent TTA methods Wang *et al.* (2020b); Mummadi *et al.* (2021); Rusak *et al.* (2022a); Goyal *et al.* (2022); Niu *et al.* (2022); Cho *et al.* (2023); Niu *et al.* (2023); Press *et al.* (2023); Döbler *et al.* (2024); Marsden *et al.* (2024). In this paper, we analyze EM to understand how it works, when and why it fails, and how to use it to predict model accuracy.

The initial intuition behind using entropy minimization, given by Wang *et al.* (2020b) was based on the observation that models tend to be more accurate on images for which they make predictions with higher confidence. The logical extension of this observation was to encourage models to bolster their confidence on such images. However, our analysis reveals this intuition to be only partly true. Remarkably, even when we construct datasets by excluding samples initially classified correctly — effectively creating datasets with a 100% classification error rate at the start — entropy minimization performance remains largely intact.

Our analysis uncovers that during entropy minimization, embeddings of images from the input dataset tend to form distinct clusters. The distances between samples within each cluster diminish, creating more defined groupings, while the centers of these clusters gradually move apart, a phenomenon akin to neural collapse Papyan *et al.* (2020); Han *et al.* (2021b); Ben-Shaul *et al.* (2023). At first, embeddings of the input images not only cluster, but also stay close to the embeddings of original training images. Only after numerous optimization steps do these embeddings begin to diverge, distancing themselves from the embeddings of the clean training data (Fig. 4.1). We show this divergence to be intricately tied to a reduction in the model's accuracy.

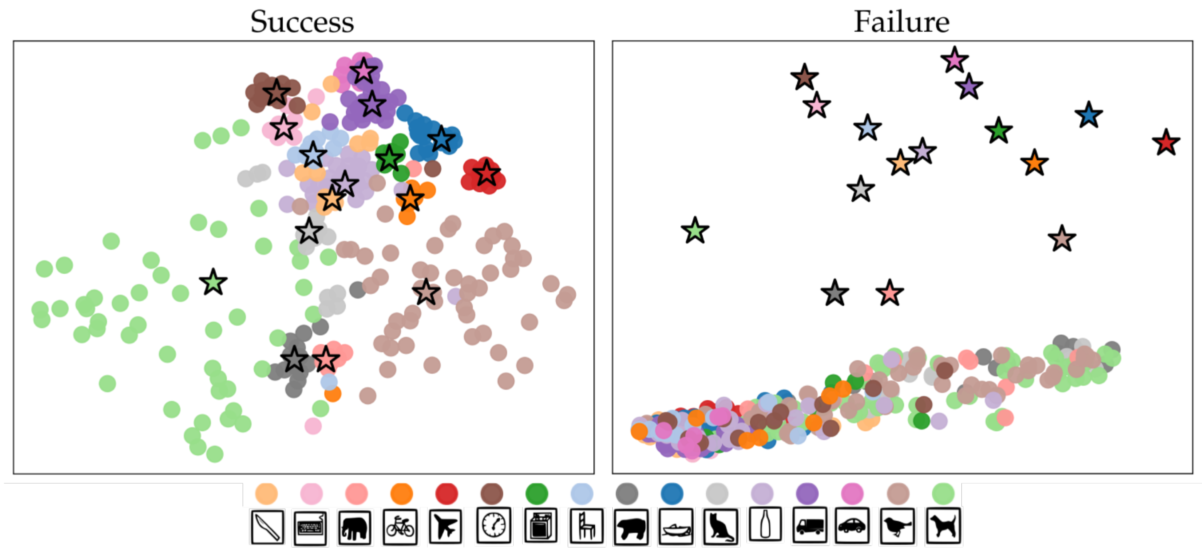


Figure 3.1: **Understanding the successes and failures of EM through clustering embedding dynamics.** After a few iterations of adaptation (left), EM improves the accuracy of pretrained classifiers by embedding the input test data near mean embeddings of classes from the training data, marked by stars. Eventually, after many iterations (right), EM fails, because it embeds input test data far from where training data is embedded. We show the t-SNE embeddings of 16-class-Imagenet Geirhos *et al.* (2018a), throughout adaptation to Gaussian Noise 3 Hendrycks and Dietterich (2019a).

Drawing from our insights, we present a method designed to estimate the accuracy of a given model on any dataset, without labels. This task is notably difficult, because in some cases in-distribution accuracy is tied to out-of-distribution (OOD) accuracy Miller *et al.* (2021), while in other cases it is not Teney *et al.* (2022). Our approach, termed Weighted Flips (WF), works in conjunction with TTA methods as they adapt to input data, with minimal added overhead. Using approximations of cluster consistency, WF estimates the accuracy of the network by measuring how the predictions of a fixed set of images change: the more they change, the lower the consistency of the clusters and the lower the predicted accuracy. We validate the efficacy of our method across an extensive array of 23 ImageNet-scale Deng *et al.* (2009) datasets, encompassing diverse challenges, such as random adversarial noises, hard images, and datasets featuring OOD classes. WF surpasses the prior state-of-the-art methods by a substantial margin of 29.62%, setting a new benchmark in the accuracy estimation domain.

3.2 The Mystery of Entropy Minimization

EM has been validated as effective in semi-supervised settings, with pioneering work by Grandvalet and Bengio (2004) and subsequent advancements, such as Tent Wang *et al.* (2020b), which demonstrated EM's ability to enhance the accuracy of pre-trained classifiers on unlabeled ImageNet-scale Deng *et al.* (2009) datasets. EM operates by iteratively optimizing the model to minimize the entropy of the output classification probabilities, denoted by $H(\hat{y}) = -\sum_c p(\hat{y}_c)\log p(\hat{y}_c)$, where $\hat{y}$ is the logits vector and $p(\hat{y}_c)$ is the probability assigned to class c. This approach inherently boosts the likelihood of the most probable classes while diminishing that of the others. Wang *et al.* (2020b) observed a correlation between lower output entropy and accuracy, indicating that images with low entropy outputs are more likely to be classified correctly. Subsequent studies, including Niu *et al.* (2022); Press *et al.* (2023); Marsden *et al.* (2024), have built on this foundation, assigning more weight to lower-entropy samples, and even ignoring high-entropy samples entirely.

To assess the influence of correctly classified images on EM's effectiveness, we tested the effects of omitting images that were initially correctly classified by the model. If such images are pivotal in EM's ability to enhance classifier performance, we expect a notable decline in the EM efficacy.

For this purpose, we utilized ImageNet-C Hendrycks and Dietterich (2019a) Gaussian Noise level 3, dividing it into training and holdout sets. The training set was replicated seven times, systematically omitting images for which the ground truth label lay somewhere in the pre-trained model's top-k predictions, for ($k \in [1, 2, 3, 5, 10, 20, 50]$). Concretely, for $k = 1$, all accurately classified images were excluded, and for $k = 2$, images whose label ranked within the top two predictions were removed, and so forth. Each altered training set was used to adapt a Tented model. The model's accuracy was then evaluated on the holdout set, with evaluations every ten iterations, spanning a total of 1,000 iterations.

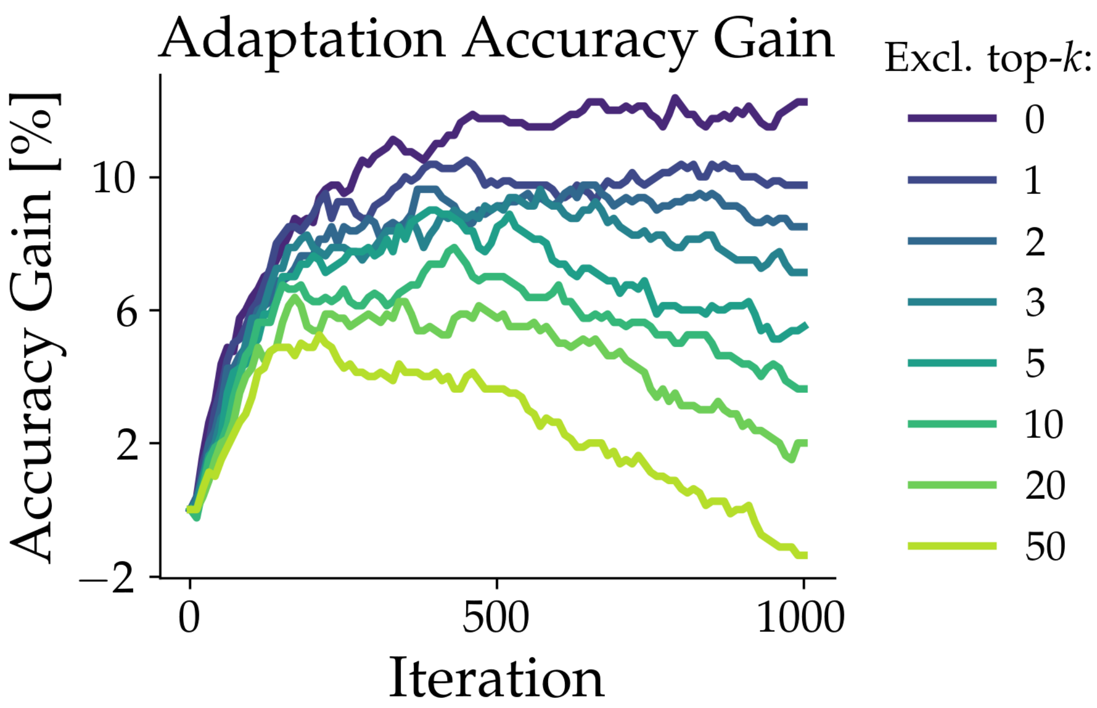


Figure 3.2: **EM remains effective even when initially correctly classified images are excluded**. Accuracy gain per iteration on a holdout set, as Tent adapts to its inputs. Surprisingly, the performance gain on the holdout set is high, even when we exclude top-k samples from the training set. When top-$k = 0$, no images are excluded.

The experiment results (see Figure 3.2) are revealing, underscoring the robustness of EM. Notably, EM's effectiveness endures even when images initially classified correctly are excluded.

For instance, removing all initially correctly classified images before adaptation produces an increase in accuracy comparable to not removing any images, with gains of 10.50% and 12.38%, respectively. Even more remarkable is the persistence of this trend: with $k = 10$, the model still registers a notable accuracy improvement of 7.88%. This observation is particularly striking given the nature of the excluded images – they are not just numerous, but also represent the highest quality, being those the network is most certain about. Specifically, images excluded at $k = 1$, which constitute 45% of the dataset, have an average entropy of 1.85, markedly lower than the original dataset's average entropy of 2.84.

Additionally, we also tested the effects of removing images according on their initial entropy level, and found similar results (see Appendix B.7). These findings intriguingly suggest that the model's accuracy and entropy on individual images may not be as pivotal to EM's success in enhancing classifier performance as previously thought. It reveals a nuanced dimension of EM's functionality and hints at the presence of deeper mechanisms, which we will investigate next.

3.3 Phases of Entropy Minimization: Clustering Dynamics and Embedding Alignment

We analyze the evolution of input data embeddings as EM progresses through its iterations. At first, EM causes the model to increase in accuracy, which we refer to as the first phase, followed by a decrease in accuracy, which we refer to as the second phase. The number of EM iterations needed for the model to reach its maximum accuracy (the end of the first phase, and the beginning of the second) is varied and depends on the input data.

In the first phase, these embeddings align closely with the embeddings of samples from the original training distribution. However, in the second phase, this alignment starts to deteriorate; the embeddings drift progressively further from the training distribution, disrupting the initial alignment, as conceptually depicted in Figure 3.3.

To examine the clustering process across the two phases of the EM, we focus on two measures: (1) the quality of the clusters and (2) their alignment with the original training data distribution. For evaluating cluster quality, we ran k-means on the embeddings and computed the Silhouette score Rousseeuw (1987), a widely recognized metric for measuring cluster quality. The Silhouette score gauges how closely an embedding corresponds to its own cluster in contrast to neighboring clusters, with a high score indicating distinct and well-separated clusters.

To quantify the alignment between clusters and embeddings of the original training distribution, we looked at mean embeddings for the classes in the ImageNet validation set, alongside

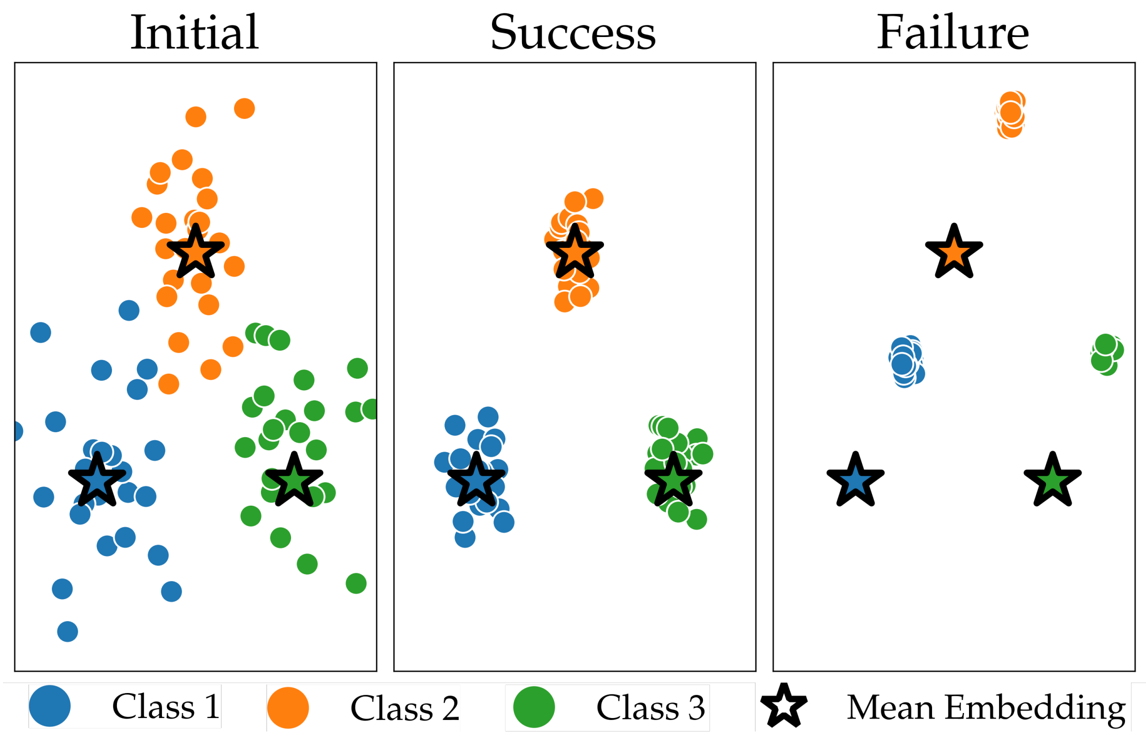


Figure 3.3: **The two-phase clustring paradigm explains EM behavior**. Intuitive visualization of EM's phases. In the first phase (success), input test data becomes more clustered, aligning closely with the mean embeddings of corresponding classes from the training data (the colored stars). In the second phase (failure), these clusters diverge from the mean embeddings.

the centroids of clusters found by k-means. We use the Hungarian method Kuhn (1955) to find a matching between mean class embeddings and centroids, which minimizes the average distance between each assigned pair of (class embedding, centroid). Henceforth, we refer to this average of distances as "Shift distance".

As ImageNet contains many similar fine-grained classes, we restrict our analyses to the 16 classes outlined in Geirhos *et al.* (2018a), which represent approximately 20% of the total images. Consequently, we use $k = 16$ when we cluster the embeddings using k-means. This focused approach allowed for a detailed and controlled examination of clustering behaviors within the framework of EM.

We now examine changes in the Silhouette score and Shift distance as Tent adapts to the input data, over 50,000 iterations using a ResNet-50 He *et al.* (2016). Figure 3.4 showcases the comparative Silhouette scores and Shift distances for both phases, incorporating findings from three diverse datasets: IN-C Hendrycks and Dietterich (2019a), IN-$\overline{\text{C}}$ Mintun *et al.* (2021), and IN-3DCC Kar *et al.* (2022).

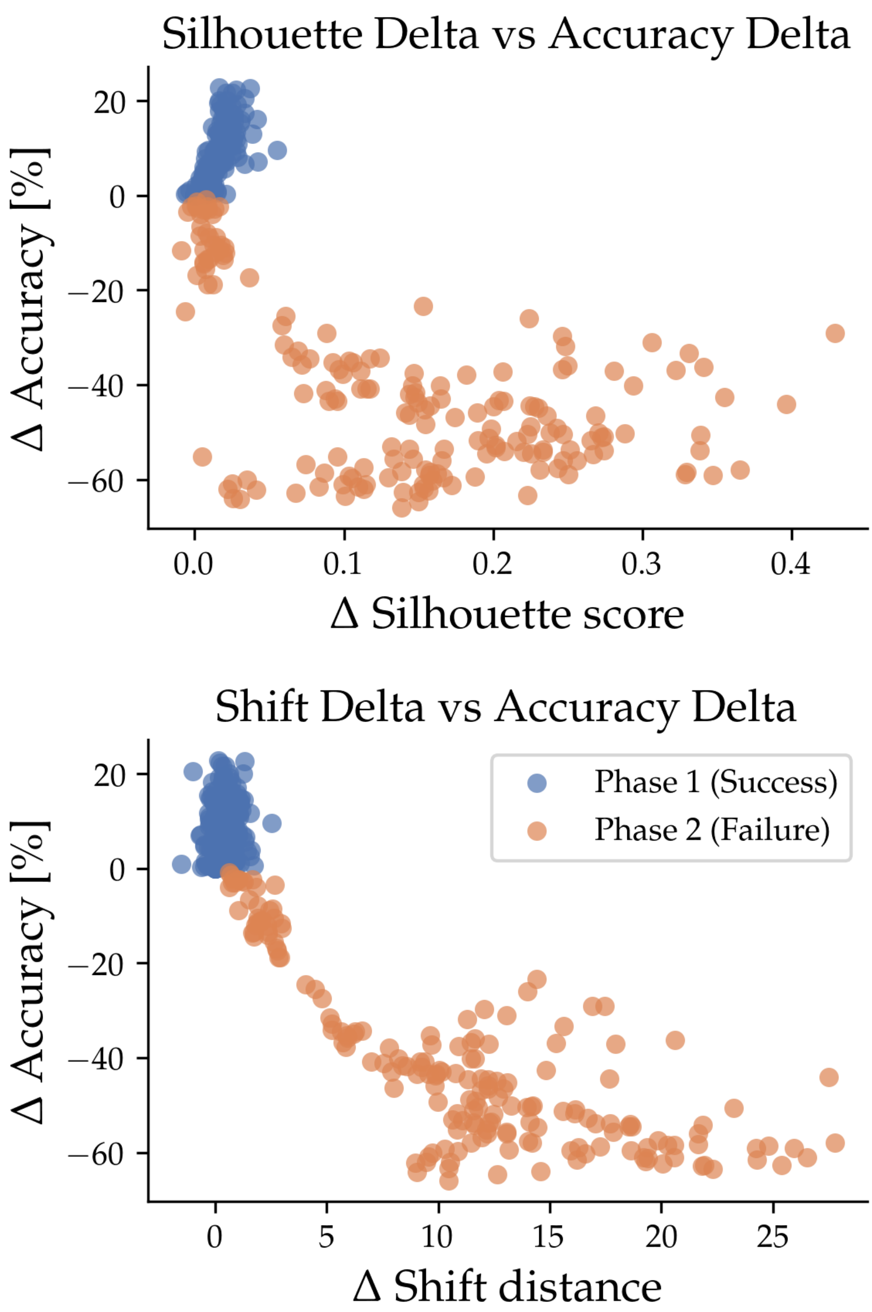


Figure 3.4: **Two-phase behavior during the EM adaption predicts accuracy.** Differences in Silhouette score, Shift distance, and accuracy for Tent adaptation. Each point corresponds to a test dataset; each dataset appears twice: once in blue, corresponding to phase 1 (success, Δ Acc ≥ 0), and once in orange, corresponding to phase 2 (failure, Δ Acc < 0). **Top:** In both phases, and across almost all datasets, the Silhouette score of embeddings increases, corresponding to a better-clustered embedding space. **Bottom:** In the first phase, input data embeddings are kept close to training image embeddings, while in the second phase, they drift away, exhibiting large Shift distance changes. The datasets used are IN-C, IN-$\overline{\text{C}}$, and IN-3DCC.

Our findings distill into two primary insights: **First**, a positive change in Silhouette score, indicative of enhanced clustering, is observed in both phases for more than 98% of cases. Notably, during the initial phase, a positive correlation exists between changes in Silhouette score and accuracy ($\rho = 0.70$, significant at $\alpha = 0.05$). **Second**, Shift distances minimally change (and sometimes diminish, signifying closer proximity to training data embeddings) in the first phase, they notably grow larger in the second phase. During this latter phase, a substantial negative correlation emerges between changes in Shift distance and accuracy ($\rho = -0.79$, significant at $\alpha = 0.05$).

Synthesizing these results reveals a nuanced picture: EM bolsters accuracy by clustering the embedded data into more concentrated clusters. This strategy remains efficacious as long as these embeddings align closely with the embeddings of the training data. However, as input data embeddings diverge from the training distribution, the classifier's accuracy diminishes. This intricate interplay offers a deeper understanding of EM's operation and its dependency on the spatial dynamics of data embeddings. We discuss the connection between EM and clustering in more detail in Appendix B.1.

3.4 Estimating Dataset Accuracy

Leveraging our understanding of EM, we tackle a critical challenge in TTA settings: estimating the accuracy of a classification model on a given dataset. Ideally, one might resort to the metrics used in this paper, namely Silhouette score or Shift distance, for this purpose. However, these metrics encounter practical hurdles: the Silhouette score depends on clustering, which varies across datasets due to differences in class distributions or the total number of classes, and calculating the Shift distance is impossible, as accessing the training data (in order to calculate mean embedding vectors per class) is forbidden in most TTA settings Wang *et al.* (2020b); Niu *et al.* (2022); Yuan *et al.* (2023).

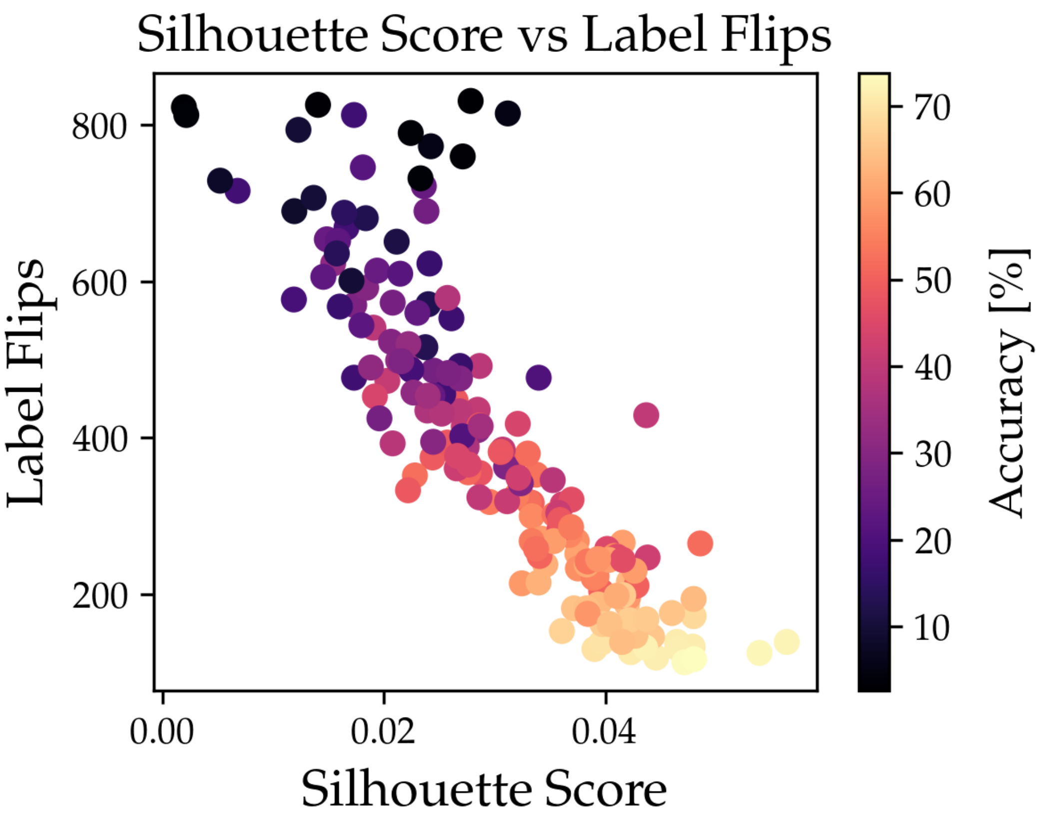


Figure 3.5: **Label flips are strongly correlated with Silhouette score**. Silhouette score at the initial iteration and the total number of label flips at the final iteration are correlated for datasets in IN-C, IN-$\overline{\text{C}}$, and IN-3DCC. Both metrics are correlated with accuracy, but measuring label flips is easier and more practical.

3.4.1 Label Flipping

Due to the difficulties of measuring these scores in practice, we take a different approach. We look at the number of images for which the model's prediction changes somewhere between the initial and the final iteration of the EM ("label flips"). According to our hypothesis, the number of label flips is correlated with the pre-trained model's accuracy on the dataset. Our reasoning is as follows: there exists a tight correlation between accuracy and Silhouette score at iteration 0 — the higher the accuracy, the better clustered the input data, shown in Figure 3.5. Therefore, we do not expect EM, which works by clustering its inputs, to significantly change an already well-clustered set of embeddings. It follows that there will likely be only a few label flips. Conversely, given a dataset with a low accuracy, its image embeddings will likely be badly clustered initially, which leads EM to change them significantly, resulting in many label flips.

We demonstrate the validity of this reasoning by adapting the state-of-the-art TTA method, Rdumb Press *et al.* (2023), to IN-C, IN-$\overline{\text{C}}$, and IN-3DCC. Initially, we used the pre-trained model to classify 1,000 input images and then recorded the total number of label flips after adaptation. The model is adapted for 1,000 iterations because Rdumb resets itself every 1,000 iterations. We find a strong correlation between accuracy and label flips, seen in Figure 3.5.

3.4.2 Weighted Flips

We now describe the Weighted Flips (WF) method of converting the count of label flips into a dataset accuracy estimate. Instead of just counting the number of flips, we additionally consider the classifier's initial confidence in its predictions for each image; images initially classified with high confidence that later flip should contribute more significantly than those with lower initial confidence. We then compute the WF as:

$$WF = \sum_i 1_{\{flip\}}(i) \cdot c_i$$

where $1_{flip}(i)$ is 1 if image i's label flipped and 0 otherwise, and c_i is the confidence percentile of image i. Utilizing pairs of weighted flips and accuracy $((\text{WF}, \text{accuracy})_k)$ from IN-Validation and ImageNet-C holdout noises, we interpolate the weighted-flips-to-accuracy function, f (refer to Figure 3.6). To estimate the accuracy of a model on an unfamiliar dataset, we adapt the model to it using RDumb (for details, see Appendix B.10), measuring flips on the first 1,000 input images. After adaptation, we count and weigh the flips, estimating the model's accuracy as $f(\text{WF})$. Importantly, WF is versatile and can work with a range of TTA methods (see Appendix B.5), and f can be interpolated in a variety of different ways (see Appendix B.2). In Appendix B.4.1, B.4.2, we present ablation studies on the effects of varying end iterations and holdout set sizes on performance.

3.4.3 Experimental Setting

Accuracy estimation methods must yield robust estimates across diverse and challenging datasets to be considered reliable. In our evaluation, we probe the effectiveness of our proposed method

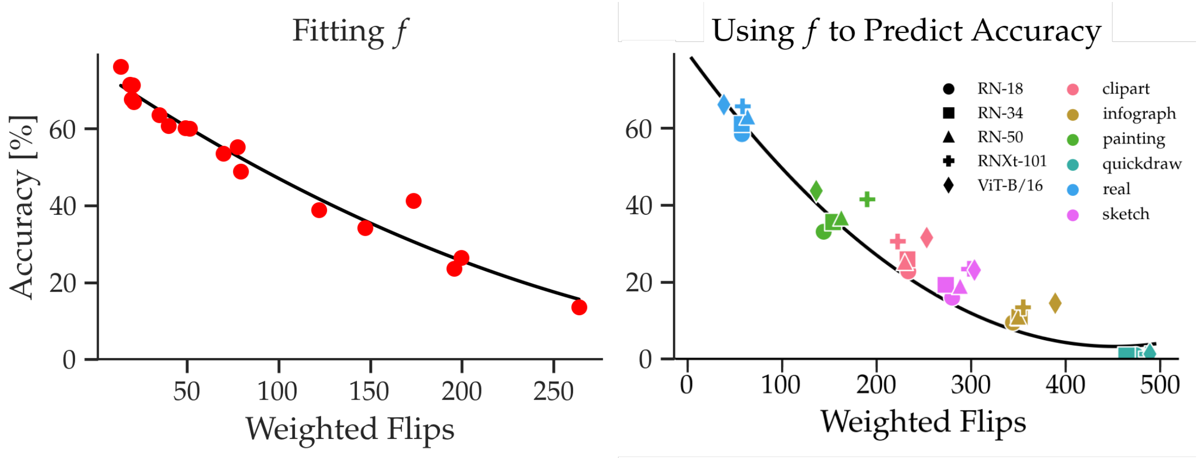


Figure 3.6: **Fitting and accuracy prediction using the WF method. Left:** Fitting f: using the noises in IN-C Holdout and ImageNet-Validation, we fit pairs of (weighted flips, accuracy), shown in red. The black curve shows the function resulting from interpolating the points, $f(x) = 0.00036x^2 - 0.32x + 75.66$. **Right:** With our weighted-flips-to-accuracy function f, we can estimate the accuracy of a model across the six splits from Rusak *et al.* (2022b). We use the same f function and show that it works across different architectures, without refitting.

using an extensive selection of popular ImageNet-scale classification datasets. This includes all classification datasets from the Shift-Happens benchmark[1]. Our chosen datasets encompass a wide spectrum, from various types of noise (IN-C, IN-$\overline{\text{C}}$, IN-3DCC, CCC) and domain shifts (IN-R, IN-V2, IN-D), to adversarial noises (Patch-IN, BG Challenge, IN-Obfuscations), and even images featuring classes not present in ImageNet (NINCO).

Several datasets provide multiple splits of a similar nature, the results of which we average, except for ImageNet-D Rusak *et al.* (2022b), which encompasses a variety of distinct domains. The CCC dataset Press *et al.* (2023) is particularly expansive, containing 27 splits with 7.5M images each; for practicality, we only include the initial 25k images from each split in our analysis. Altogether, our evaluation spans 326 individual dataset splits.

We briefly describe the other methods tested alongside ours:

- AC Hendrycks and Gimpel (2016): Computes the dataset-wide average confidence for the top-predicted class in each image.

- DoC Guillory *et al.* (2021): Builds upon AC by assessing the variance in mean confidence between the validation and OOD sets, demonstrating consistent enhancements in performance.

[1] https://github.com/shift-happens-benchmark/icml-2022

- ATC Garg *et al.* (2022): Estimates accuracy by determining the fraction of unlabeled data samples where the model's confidence exceeds a learned threshold.
- COT Lu *et al.* (2023): Estimates accuracy by applying Optimal Transport to quantify the disparity between OOD and in-distribution model outputs.

3.4.4 Results

Looking at Table 3.1 reveals that our WF method consistently stands out as the best estimator across a broad spectrum of ImageNet-scale datasets. WF sets a new benchmark by achieving an average estimation error of just 5.75%, significantly outperforming the nearest competitor, COT, reducing the relative error by 29.62%. This exemplary performance of WF is not limited to average cases; even in the most challenging scenarios of worst-case performance, WF maintains its superiority, cutting the error by 29.74% compared to COT. Furthermore, WF demonstrates remarkable consistency as an estimator. In 18 of the 23 datasets evaluated, it either leads the pack or comes a close second. This is in stark contrast to the performance of COT, which, despite being second-best, only achieves top-two rankings in 12 datasets. The persistent effectiveness of WF across diverse conditions underscores its reliability and superiority in accuracy estimation.

Practicality of WF: Beyond its top-tier performance, WF stands out for its practicality. It operates concurrently with the EM process, requiring only three parameters that define the weighted-flips-to-accuracy function, f. This process adds minimal computational overhead, requiring only 20 additional forward passes for every 1,000 Rdumb iteration steps. Lastly, WF is effective even when only a small number of samples are available, see Appendix B.3.

Table 3.1: Mean Absolute Error between estimated accuracy, and true accuracy on a ResNet-50 model, for 4 estimation methods (AC, DoC, ATC, COT) Hendrycks and Gimpel (2016); Guillory *et al.* (2021); Garg *et al.* (2022); Lu *et al.* (2023), and ours. Our method (WF) is consistently either best or second best, with the best average and worst-case performance across many different OOD datasets. **Best** results are in bold; <u>second best</u> are underlined, {.} indicates how many splits are in each dataset, when there are more than 1.

Datasets	AC	DoC	ATC	COT	WF (ours)
Noises					
IN-C {75} 94	10.06	6.61	7.44	**2.23**	<u>4.79</u>
IN-$\overline{\text{C}}$ {50} 156	19.48	15.96	12.16	**3.17**	<u>7.35</u>
IN-3DCC {60} 115	11.83	<u>3.44</u>	8.15	**3.02**	3.66
CCC {27} 181	15.51	11.95	6.05	**2.04**	<u>2.80</u>
Domain Shifts					
Stylized 73	31.63	28.08	7.36	<u>12.18</u>	**3.81**
IN-V2 {3} 187	5.58	2.41	**0.45**	<u>2.68</u>	4.70
IN-Sketch 239	22.34	18.78	**0.15**	4.23	<u>1.71</u>
IN-R 99	23.21	19.65	**0.37**	2.44	<u>1.88</u>
IN-D Rusak *et al.* (2022b)					
⟼ Real	10.56	7.00	**1.35**	27.54	<u>3.18</u>
⟼ Painting	17.40	13.85	<u>3.27</u>	7.49	**2.12**
⟼ Clipart	21.27	17.72	**1.62**	4.52	<u>3.37</u>
⟼ Sketch	24.43	20.87	**0.61**	<u>0.71</u>	5.44
⟼ Infograph	54.12	50.57	36.26	**3.44**	<u>3.63</u>
⟼ Quickdraw	32.67	29.11	4.13	**1.60**	<u>2.57</u>
Cartoon & Drawing {2} 198	15.69	12.13	<u>4.42</u>	**1.62**	13.25
Adversarial Noises					
BG Challenge {8} 247	10.54	7.37	**4.88**	19.68	<u>6.92</u>
IN-A Hendrycks *et al.* (2021c)	45.12	41.57	**20.51**	30.38	<u>21.61</u>
IN-C Patch {75} 84	4.37	**0.16**	4.42	2.57	<u>1.60</u>
IN-Hard Taesiri *et al.* (2023)	29.71	26.15	<u>6.73</u>	15.33	**3.64**
Patch-IN {10} 175	<u>8.06</u>	**5.11**	**5.11**	10.13	8.87
IN-Obfuscations {3} 212	99.90	96.34	99.90	**0.12**	<u>4.58</u>
OOD/Other					
ObjectNet 14	34.59	31.03	<u>9.43</u>	10.40	**2.74**
NINCO 22	50.29	46.74	26.97	<u>20.28</u>	**18.07**
Average	26.02	22.29	11.81	<u>8.17</u>	**5.75**
Worst Case	99.90	96.34	99.90	<u>30.38</u>	**21.61**
Average (Worst Case Excluded)	22.66	18.92	7.81	<u>7.16</u>	**5.03**

Table 3.2: Mean Absolute Error between estimated accuracy and true accuracy, across different architectures. Using the same weighted-flips to-accuracy function, f, works across different architectures and models, without need for finetuning. For each model and dataset, the task is to estimate the accuracy of that model on the dataset. **Best** results are in bold; <u>second best</u> are underlined. AugMix: ◇ ANT: ‡ DeepAugment: ♠ Hendrycks *et al.* (2020a); Rusak *et al.* (2020b); Hendrycks *et al.* (2021a)

Datasets	RN-50	RN-18	RN-34	RN-50‡	RN-50◇	RN-50◇♠	RNXt-101	RNXt-101♠	ViT-B/16	MaxViT-T
IN-C	<u>4.79</u>	7.21	6.04	5.39	5.02	4.81	5.35	**4.12**	8.34	6.73
IN-$\overline{\text{C}}$	7.35	7.90	6.77	6.84	6.60	6.48	**5.60**	<u>5.63</u>	6.59	4.97
IN-3DCC	3.66	3.58	3.89	3.20	<u>3.07</u>	**2.98**	7.23	4.37	7.19	6.79
IN-V2	4.70	4.11	**3.37**	<u>3.67</u>	5.06	5.00	6.47	5.54	4.44	6.08
Real	3.18	2.83	**0.38**	2.72	6.59	3.30	4.24	<u>0.61</u>	1.02	3.40
Painting	2.12	5.36	0.78	**0.59**	7.62	2.51	12.02	1.12	3.02	<u>0.60</u>
Clipart	3.37	<u>1.59</u>	4.42	0.32	6.19	2.19	7.24	**0.53**	12.82	4.52
Sketch	5.44	**1.53**	3.93	6.18	9.73	3.60	10.75	<u>1.89</u>	11.04	10.88
Infograph	3.63	<u>1.76</u>	3.67	3.74	6.78	**0.28**	6.37	2.34	9.27	9.13
Quickdraw	2.57	2.34	2.24	2.20	2.53	1.27	2.27	1.21	2.36	2.31
Average	4.08	3.82	3.55	3.49	5.92	<u>3.10</u>	6.76	**2.74**	6.61	5.54

Versatility across Models and Architectures: To demonstrate the adaptability of the WF method, we tested it across various models and architectures, employing the **same** weighted-flips-to-accuracy function, f, used in our primary experiments (Table 3.1). Testing encompassed different ResNet variants, including models enhanced with noise augmentation techniques, such as ANT Rusak *et al.* (2020b), AugMix Hendrycks *et al.* (2020a), and DeepAugment Hendrycks *et al.* (2021a). Additionally, we evaluated a ResNext-101 Xie *et al.* (2017), ViTB-16 Dosovitskiy *et al.* (2010), and MaxViT-T Tu *et al.* (2022). The mean absolute errors between estimated and actual accuracies are reported in Table 3.2. Remarkably, 5 of the 8 models tested achieved a lower mean absolute error than the baseline model, RN-50, showing that f maintains its efficacy across different model architectures. When f is refitted on the architecture that WF is evaluated on, performance improves (see Appendix B.6).

Robustness to Dataset Choice: In Table 3.1, we derived the weighted-flips-to-accuracy function f using IN-C holdout and ImageNet validation noises. We further validated the robustness of the WF method by fitting f using a subset of the 23 datasets and then assessing its performance on the remaining datasets. As an added challenge, we excluded datasets used in the original configuration: IN-C Holdout and ImageNet-Validation. For each subset size, we

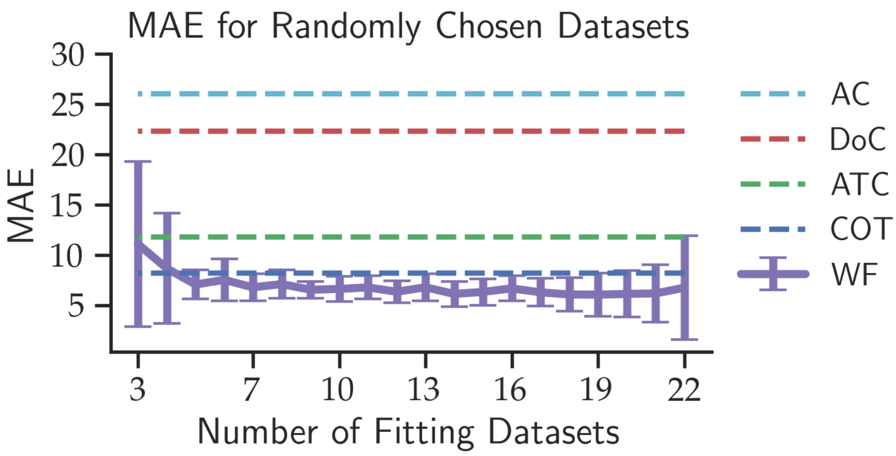


Figure 3.7: **WF outperforms other methods across almost all subset sizes**. Mean Absolute Error of WF when using a weighted-flips-to-accuracy function f to fit on random subsets of the 23 datasets in Table 3.1. For each point on the x-axis, we sample 50 fitting datasets for WF, and plot the average and the standard deviation of the MAE. For the other methods, we plot average MAE across all datasets

.

repeated the fitting and evaluation process 50 times. The results, plotted in Figure 3.7, illustrate that the WF method consistently outperforms COT across almost all subset sizes, reinforcing its resilience and reliability across a broad spectrum of datasets.

3.5 Related Work

To the best of our knowledge, the first time EM was shown to be useful for improving a classifier's accuracy was in Grandvalet and Bengio (2004). They showed how EM can be applied to a logistic regressor, and found it to be beneficial in cases where the data was corrupted by outliers. Following this, Lee (2013) proposed pseudo labeling as a means of improving classification accuracy on MNIST. Interestingly, t-SNE is used to show that pseudo labeling works partly by encouraging the model's embeddings to be better clustered, and away from the decision boundaries of the model. Moreoever, it is stated that pseudo labeling is equivalent to entropy regularization Grandvalet and Bengio (2004). Although this might be true in the settings considered then, pseudo labeling was shown to be less effective (and thus not equivalent) on larger-scale datasets, by Tent. Unlike previous work, we demonstrate that EM clusters by measuring the Silhouette score of the clusters themselves, allowing us to empirically evaluate

ImageNet scale datasets. Additionally, we show what happens when EM fails, which is not discussed in prior work, with the exception of Oliver *et al.* (2018), which shows how EM fails to adapt to a toy "two moons" dataset, because the model increases the magnitude of its output logits. This isn't the case in most TTA settings, as the final layer of the model isn't trained.

Minimizing entropy at test time was popularized by Tent Wang *et al.* (2020b), which demonstrated the effectiveness of EM on large-scale datasets, such as ImageNet-C.

Entropy minimization is ideal for domain adaptation: it can be used on a trained model, without retraining, and doesn't require balancing a proxy loss with a classification loss, as in Gidaris *et al.* (2018); Sun *et al.* (2020b); Gandelsman *et al.* (2022).

Though many prior works use losses that are based on entropy Wang *et al.* (2020b); Rusak *et al.* (2022a); Goyal *et al.* (2022); Mummadi *et al.* (2021); Wang *et al.* (2022); Niu *et al.* (2022); Cho *et al.* (2023); Press *et al.* (2023); Niu *et al.* (2023); Döbler *et al.* (2024); Marsden *et al.* (2024), little is known as to *why* it works. Additionally, entropy minimization, when used in TTA settings, is effective for only a limited number of iterations, before the classifier degrades to chance accuracy, shown in Press *et al.* (2023). Interestingly, this degradation of accuracy, named "collapse", differs from classical definitions of catastrophic forgetting in continual learning De Lange *et al.* (2021), in that the task itself does not change.

A plethora of methods have been used for adapting a trained classifier to out-of-domain data: from using an auxiliary loss to help learn the test domain Sun *et al.* (2019b, 2020b); Gandelsman *et al.* (2022) through simply re-estimating the mean and variance statistics Schneider *et al.* (2020); Nado *et al.* (2020) to using image augmentations Wang *et al.* (2022); Song *et al.* (2023); Chakrabarty *et al.* (2023). However, for their simplicity and success, entropy minimization-based methods are still the most widely used and successful in settings most relevant to this work.

Works that follow Tent improve EM by modifying the loss to be more robust to label noise Rusak *et al.* (2022a) or smoother Mummadi *et al.* (2021), or by adjusting the temperature of the output distribution Goyal *et al.* (2022). While testing on long sequences of images, both Wang *et al.* (2022) and Niu *et al.* (2022) show that Tent degrades in accuracy, the more iterations it does. Press *et al.* (2023) show that this is in fact true for all TTA methods apart from EATA Niu *et al.* (2022), which uses an L2 regularizer to constrain the adapting model's weights to be close to those of the pretrained model. Niu *et al.* (2023) study the effects of batch size, label shifts and other factors on adaptation; they propose a method to stabilize adaptation. Similarly, Döbler *et al.* (2024) also test entropy minimization-based methods in real-world conditions, and propose a new method based on a diversity and a weighted entropy loss. Entropy has also been used in semi-supervised settings: Sohn *et al.* (2020) propose augmentation and an entropy loss to train a classifier when only a few labels are available.

Analyzing which labels flip during training has been studied in Toneva *et al.* (2018), which explored which samples are forgotten during training. Another work, Deng *et al.* (2022) looked at how to reduce the amount of times a label flips during training. The agreement/disagreement between different models on ID data was shown to be linearly correlated to OOD accuracy and has been recently used to estimate accuracy in Miller *et al.* (2021); Jiang *et al.* (2021); Baek *et al.* (2022); Kim *et al.* (2023). These works are beyond the scope of this work, as they require access to multiple models and ID data, which is disallowed in most TTA settings Wang *et al.* (2020b); Niu *et al.* (2022); Yuan *et al.* (2023).

3.6 Conclusion

While EM is a cornerstone in many TTA methods, the mechanics of its success have remained enigmatic. This study sheds light on the transformative journey of input data embeddings under the EM adaption. It reveals a biphasic clustering process, where alignment with the training data's embedding clusters bolsters accuracy, followed by a subsequent phase where excessive divergence diminishes it.

Our work goes beyond deciphering the mystery behind entropy minimization; it also utilizes this knowledge to significantly refine the precision of model accuracy predictions in TTA contexts. This dual achievement underscores the potential of deep analytical approaches in enhancing the efficacy and applicability of machine learning models.

CHAPTER 4

CiteME: Can Language Models Accurately Cite Scientific Claims?

The previous two chapters benchmarked and analyzed methods to improve the performance of image classifiers. Following the success of large models like CLIP Radford *et al.* (2021), I struggled to find images that are both natural (realistic and not adversarial) and difficult for these models to classify; it seemed like image classification as a task was solved. For example, Vasudevan *et al.* (2022) show that nearly half of the errors made by a state-of-the-art model on ImageNet stemmed from incomplete or incorrect labels, and were not actual model errors. Further work Anzaku *et al.* (2024) also explains how the single-label regime of ImageNet can be misleading when it comes to estimating model performance. Given this, it felt like improving vision classifiers had run its course (at least for me), and I wanted to turn to problems that were further away from being solved.

To address this, I turned to language models. Similar to image classification models, I was interested in identifying realistic examples where language models fail. Around the same time, my lab-mate Andreas was working on using language models as paper recommendation tools. We decided to collaborate and began developing a benchmark and an agent together.

The ideas behind the work in this chapter are straightforward: we want to turn paper recommendation into a benchmark. Paper recommendation is inherently vague. For instance, if we have a system that takes a list of papers or topics as input, which papers should it recommend? The answer can be highly subjective and can vary from person to person.

To address this, we created a benchmark using citations directly from research papers. We collected text excerpts taken from research papers that cite only one paper, while also describing the cited paper in specific detail. Then, we removed the citation itself, and turned the set of excerpts into a task: given a text excerpt, the goal is to find the removed citation.

We found this task to be relatively easy for humans but challenging for language models. In order to boost the performance of language models, we proposed CiteAgent, which enables language models to search for and read papers. CiteAgent takes existing language models and improves their performane on our benchmark, without extra training or fine-tuning. However, even when using CiteAgent, state-of-the-art language models do not perform as well as expert humans.

The material in this chapter is adapted from Press *et al.* (2024a).

4.1 Introduction

Scientific discoveries are advancing at an ever-growing rate, with tens of thousands of new papers added just to arXiv every month arXiv (2024). This rapid progress has led to information overload within communities, making it nearly impossible for scientists to read all relevant papers. However, it remains a critical scholarship responsibility to check new claims and attribute credit to prior work accurately. Language models (LMs) have shown impressive abilities as assistants across tasks Dakhel *et al.* (2023), which leads us to explore the following task in this paper: *Can language models act as research assistants to help scientists deal with information overload?*

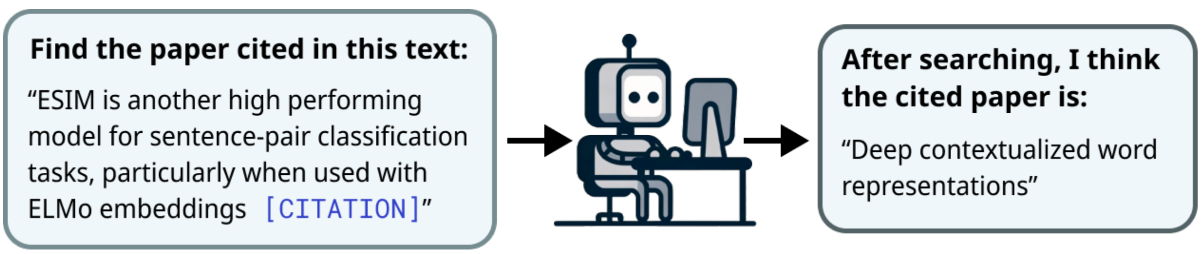


Figure 4.1: **Example of a CiteME instance.** The input (left) is an excerpt from a published paper with an anonymized citation; the target answer (right) is the title of the cited paper.

We make progress towards answering this question by evaluating the abilities of LMs in citation attribution (Färber and Jatowt, 2020; Metzler *et al.*, 2021). Given a text excerpt referencing a scientific claim, *citation attribution* is the task in which a system is asked to fetch the title of a referenced paper, as illustrated in Figure 4.1.

Current benchmarks are collected automatically, which leads to the dominance of ambiguous or unattributable text excerpts that make overly broad claims or are not used as evidence for any specific claim, as shown in Table 4.1. Furthermore, these benchmarks typically frame citation attribution as a retrieval task from a small set of pre-selected papers where only paper titles and abstracts can be viewed, not the full paper's content important for citation attribution (Cohen *et al.*, 2010; Lin, 2009).

Table 4.1: Percentage of reasonable, ambiguous, unattributable, and trivial excerpts across 4 citation datasets, as labeled by human experts. For a detailed breakdown of every analyzed sample, see Appendix C.1.

	Reasonable [%]	Ambiguous [%]	Unattributable [%]	Trivial [%]
FullTextPeerRead 110	24	26	34	16
ACL-200 21; 150	26	42	18	14
RefSeer 106; 150	24	28	32	16
arXiv 85	10	50	30	10
Average	21	36.5	28.5	14

To address these issues, we introduce *CiteME* (Citation for Model Evaluation), the first *manually curated* citation attribution benchmark with text excerpts that unambiguously reference a single paper. CiteMe's use of only unambiguous text excerpts eliminates the subjectivity that characterizes other benchmarks.

To evaluate CiteMe, we conduct benchmark tests that focus on *open-ended* citation attribution. Human evaluators confirm the lack of ambiguity, achieving 69.7% accuracy while taking just 38.2 seconds on average to find the referenced papers. The current state-of-the-art system, SPECTER2 Singh *et al.* (2022), experiences 0% accuracy on CiteME, highlighting the real-world difficulties of LM-based citation attribution. Similarly, current frontier LMs achieve performance of 4.2-18.5%, substantially beneath human performance. We conclude that current LMs cannot reliably link scientific claims to their sources.

To bridge this gap, we introduce CiteAgent, an autonomous system built on top of the GPT-4o Achiam *et al.* (2023) LM and the Semantic Scholar search engine Kinney *et al.* (2023). CiteAgent can search for and read papers repeatedly until it finds the referenced paper, mirroring how scientists perform this scholarship task to find targeted papers. CiteAgent correctly finds the right paper 35.3% of the time when evaluated on CiteME.

In summary, our main contributions are:

- CiteME, a challenging and human-curated benchmark of recent machine learning publications that evaluates the abilities of LMs to correctly attribute scientific claims. CiteME is both natural and challenging, even for SoTA LMs.

- CiteAgent, an LM-based agent that uses the Internet to attribute scientific claims. Our agent uses an existing LM without requiring additional training. It also uses a search engine, which makes it applicable to real-world settings and differentiates it from systems that can search only within a predetermined corpus of papers.

Future work that improves the accuracy of CiteME may lead to systems that can verify *all* claims an LM makes, not just those in the ML research domain. This could reduce the hallucination rate Zhang *et al.* (2023) and increase factuality Augenstein *et al.* (2023) of LM-generated text.

4.2 The CiteME Benchmark

We now present the CiteME benchmark, which we differentiate from other citation prediction benchmarks that are automatically curated, *i.e.,* curated without human supervision or feedback in selecting text excerpts Giles *et al.* (1998); Gehrke *et al.* (2003); Bird *et al.* (2008); Huang *et al.* (2014); Radev *et al.* (2013); Kang *et al.* (2018); Jeong *et al.* (2020); Gu *et al.* (2022b). For comparison, we study the quality of excerpts across four popular citation prediction benchmarks (FullTextPeerRead, Jeong *et al.* (2020), ACL-200 Bird *et al.* (2008); Medić and Šnajder (2020), RefSeer Huang *et al.* (2014); Medić and Šnajder (2020), and arXiv Gu *et al.* (2022b)). Specifically, we sample 50 excerpts from each dataset and categorize them using the following criteria:

(1) Attributable vs Unattributable. The cited paper should provide *evidence for the statement in the text excerpt*, i.e., be an attribution as opposed to a statement that does not clearly refer to supporting evidence. Excerpts that do not follow this criterion are termed *unattributable,* as in the example:

For all of our experiments, we use the hyperparameters from `[CITATION]`.

(2) Unambiguous vs Ambiguous. The cited text excerpt should not be overly broad. The ground truth cited papers should clearly be the *only possible reference* for the claim in the text excerpt. Excerpts that do not follow this criterion are termed *ambiguous*, as in the example:

`[CITATION1, CITATION2]` *explored paper recommendation using deep networks.*

(3) Non-Trivial vs Trivial. The text excerpt should not include author names or title acronyms, which simply tests LM memorization and retrieval. Excerpts that do not follow this criterion are termed *trivial*, as in the example:

SciBERT `[CITATION]` *is a BERT-model pretrained on scientific texts.*

(4) Reasonable vs Unreasonable. The text excerpt should be attributable, unambiguous and non-trivial. We term excerpts that do not follow this criterion *unreasonable*, but we categorize them according to the underlying issue (e.g., unattributable, ambiguous, or trivial). An example of a reasonable excerpt is:

We use the ICLR 2018–2022 database assembled by `[CITATION]`*, which includes 10,297 papers.*

In Table 4.1 (left), we demonstrate that most samples from all four datasets lack sufficient information for humans to identify the cited paper and are often labeled as ambiguous or unattributable. Additionally, an average of 17.5% of the samples are tagged as trivial because they include the title of the paper or its authors directly in the excerpt. Excerpts also frequently have formatting errors, making some nearly unreadable (see examples in Appendix C.1). Past work also notes similar artifacts Gu *et al.* (2022b); Jeong *et al.* (2020); Medić and Šnajder (2020), further supporting our claims. This analysis leads us to contend that performance on existing citation benchmarks might not reflect real-world performance of LM research assistants.

In response to these deficiencies, we created CiteME, a new benchmark with human expert curation for unambiguous citation references. CiteME contains carefully selected text excerpts, each containing a single, clear citation to ensure easy and accurate evaluation.

Curation. A team of 4 machine learning graduate students, henceforth referred to as "experts", were responsible for collecting text excerpts. The experts were instructed to find samples that (1) referenced a single paper and (2) provided sufficient context to find the cited paper with scant background knowledge. Each sample was checked for reasonableness; only those deemed reasonable by two or more experts were retained. Some excerpts were slightly modified to make them reasonable.

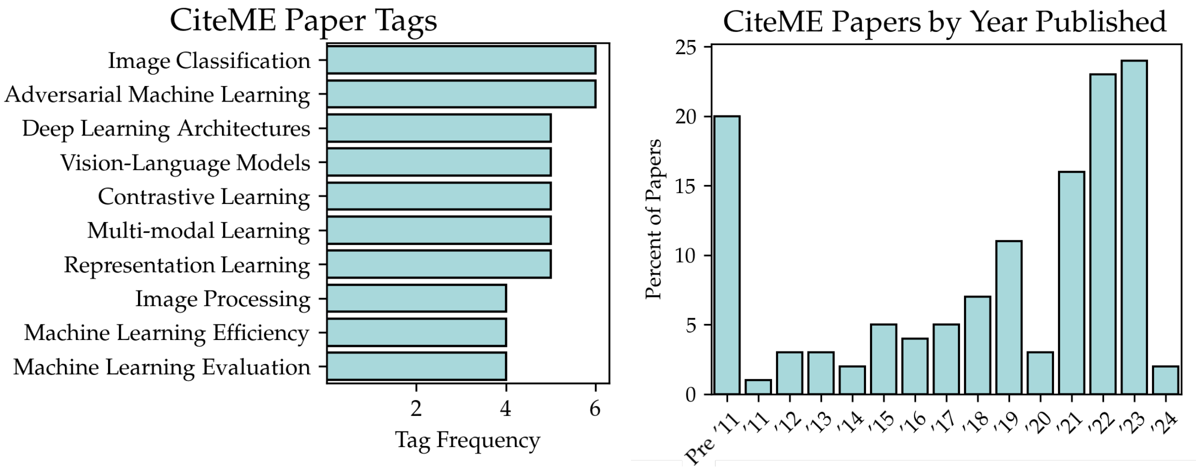


Figure 4.2: *(Left)* The top 10 most frequent labels of papers in CiteME, as identified by GPT-4. Overly broad tags like "Machine Learning" or "Deep Networks" were excluded (see Appendix C.4 for details). *(Right)* Most excerpts in CiteME are from recent papers.

Filtering Out the Easy Instances. To ensure that CiteMe is a challenging and robust dataset, we remove all dataset instances that GPT-4o can correctly answer. Filtering datasets by removing the samples that a strong model can correctly answer was previously done in Bamboogle Press *et al.* (2022) and the Graduate-Level Google-Proof Q&A Benchmark Rein *et al.* (2023). In our filtering process, GPT-4o was used with no Internet access or any other external tools. Therefore, it could answer only correctly specified papers that it memorized from its training process. We ran each sample through GPT-4o five times to cover its different outcomes. In the end, we filtered out 124 samples, leaving 130 samples in total.

Human Evaluation. To ensure that our benchmark instances are not unsolvable, we evaluate human performance on them. Using a random subset of 100 samples, we asked a group of 20 experts, who were not part of benchmark construction, to perform the task of finding the referenced papers given only the excerpt, with each expert given 5 random samples from CiteME and a maximum of two minutes to solve each instance (similar to Lála *et al.* (2023)). We observe that the experts found the correct citation 69.7% of the time, spending an average of only 38.2 seconds to do so. Note that this accuracy number does not represent the maximum-possible human performance since our annotators were limited to two minutes per question for budget reasons. Human accuracy may rise even higher given more time per instance. To check the experts' consistency, five more experts were asked to solve the same instances previously answered by the original experts. In 71% of the cases, both experts agreed on the answer, and at least one expert got to the right answer in 93% of cases.

Are 130 questions sufficient to evaluate LMs? Though traditional machine learning benchmarks usually contain thousands or even millions of test samples, recent work Chen *et al.* (2021); Press *et al.* (2022); Saharia *et al.* (2022); Wu *et al.* (2024a) shows that LM benchmarks can include only 100-200 samples and remain insightful. HumanEval Chen *et al.* (2021), for example, which consists of 164 programming problems, is among the most influential LM datasets today, appearing in virtually every SoTA LM paper recently published Ouyang *et al.* (2022); Achiam *et al.* (2023); Touvron *et al.* (2023); Chowdhery *et al.* (2023). Similarly, Bamboogle Press *et al.* (2022) contains 125 questions, DrawBench Saharia *et al.* (2022) contains 200 instances, and Plot2Code Wu *et al.* (2024a) contains 132 questions. This is in line with Prabhu *et al.* (2024); Polo *et al.* (2024), who show that benchmarks with many samples can be reduced to around 100 samples without sacrificing their utility. In addition, smaller benchmarks are advantageous because they are both cheaper to evaluate and impose a less significant environmental impact Schwartz *et al.* (2020).

4.3 CiteAgent

We now describe CiteAgent, an LM-based system that we built to mimic researcher performance of open-ended citation attribution. A researcher seeking the correct attribution for a claim might use a search engine, read several papers, refine the search query, and repeat until successful. To allow CiteAgent to perform these actions, we built it to use Semantic Scholar to search for and read papers. Unless specified otherwise, we refer to CiteAgent with the GPT-4o backbone

Table 4.2: Commands available to the model using our system.

Command	Description
`search(query, sort)`	Searches for a query; sorts results by relevance or by citation count; returns a list of papers, where each item consists of the paper ID, title, number of citations, and abstract.
`read(ID)`	Returns the full text of a paper, including title, author list, abstract, and the paper itself.
`select(ID)`	Selects a paper from the search results as the answer.

simply as CiteAgent throughout this paper.

Given a text excerpt, we prompt CiteAgent to perform one of a fixed set of custom commands and provide the output that the given command generated. CiteAgent then gives its rationale before performing another action, following Yao *et al.* (2022); Yang *et al.* (2024a). Figure 4.3 shows this process. We now describe the starting prompt and custom agent commands.

Prompt. Our prompt includes the task description, descriptions of available commands, and a demonstration *trajectory*, i.e., the series of actions that the system executes while solving an instance Yao *et al.* (2022); Yang *et al.* (2024a). The trajectory includes searching, reading a paper, and searching again (see Figure 4.4). We model our prompt on the SWE-Agent prompt Yang *et al.* (2024a).

Agent Commands. CiteAgent can respond to three custom commands (see Table 4.2). It always begins by executing the **`search`** command (sorting by relevance or citation count), which searches Semantic Scholar for a query and returns top results in a sorted order. After searching, CiteAgent can either search again, **`read`** one of the listed papers, or **`select`** a paper. It can perform up to 15 actions for every sample. Once a **`select`** action is taken, the session ends, and the selected paper is recorded.

Search. CiteAgent initiates a search command by querying Semantic Scholar Kinney *et al.* (2023). We chose the Selenium API Muthukadan (2011) over the Semantic Scholar API due to the former's significantly better re-ranked queries and its ability to provide a uniform interface for both our model and human trajectory annotators.

Selenium also lets us access features such as sorting search results by relevance and citation count, which our human trajectory annotators found particularly valuable.

To ensure correctness, we filter out search results published after the excerpt's source paper, and the source paper itself. We then give CiteAgent the top 10 search results, which include paper id, title, abstract, and citation count.

Read. `Read` command execution causes CiteAgent to retrieve the open-access PDF corresponding to the selected paper from Semantic Scholar. Using the PyPDF2 library Fenniak *et al.* (2024), our system extracts the text from the PDF, excluding visual figures. It then presents the text to CiteAgent, which generates a thought and a new command. If an open-access PDF link is unavailable, CiteAgent returns a message to that effect. We note that due to the limited context length of 8K tokens in the LLaMA-3 LM, we excluded the `read` action when using that model.

Select. `Select` command execution causes CiteAgent to choose a paper to attribute to the input text excerpt, which ends the run. If the number of actions reaches 14, CiteAgent is prompted to make a selection, forcefully concluding the run. This design choice ensures that all runs complete within a finite time and budget.

4.4 Experiment Setup

Below, we provide detailed implementation information for the baseline models and the various CiteAgent configurations we used for our evaluations.

SPECTER Models. We present the results of SPECTER Cohan *et al.* (2020) and SPECTER2 Singh *et al.* (2022) on CiteME as our baselines. SPECTER Cohan *et al.* (2020) encodes robust document-level representations for scientific texts, achieving high performance on citation prediction tasks without the need for fine-tuning. We use the Semantic Scholar SPECTER API[1] to embed the input text excerpts and the Semantic Scholar Datasets API[2] to embed all papers on Semantic Scholar, using these embeddings as our retrieval set.

SPECTER2 models Singh *et al.* (2022) introduce task-specific representations, each tailored to different tasks. For our experiments, we use the base customization of SPECTER2 from

[1] `https://github.com/allenai/paper-embedding-public-apis`
[2] `https://api.semanticscholar.org/api-docs/datasets`

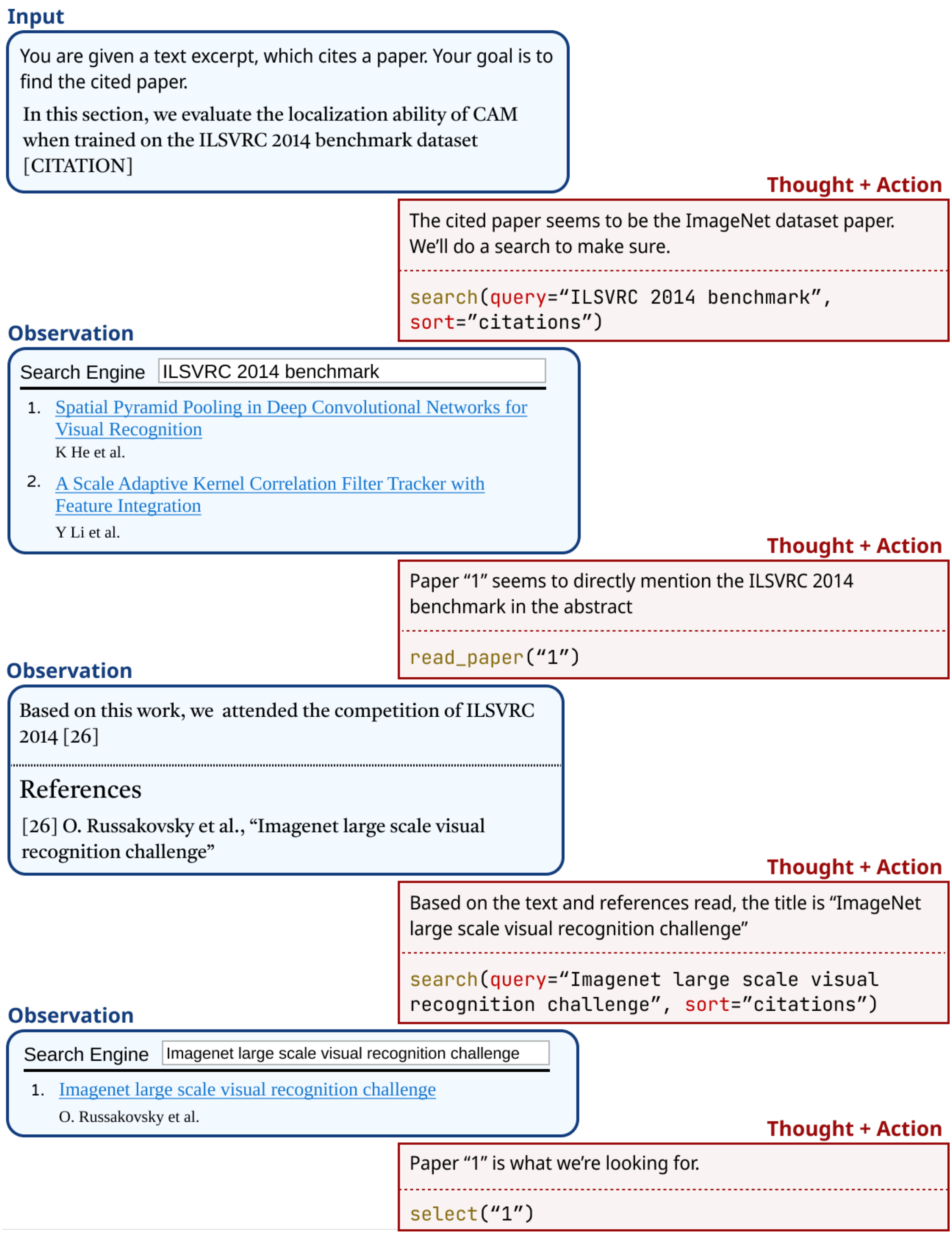


Figure 4.3: The demonstration trajectory we gave CiteAgent in the prompt.

Hugging Face[3] to embed text excerpts and the Semantic Scholar Datasets API to similarly embed all papers on Semantic Scholar, forming our retrieval set. We apply an exact kNN Lloyd (1982) match to identify the closest embedding, computing the cosine similarity between the embeddings of text excerpt and all available papers (title and abstract). Using exact kNN matches ensures no approximations/errors are introduced while matching queries. We embed the query text excerpt as title only and both title and abstract, but that did not change the performance of the SPECTER models.

CiteAgent. We run the CiteAgent system with three SoTA LMs as backbones: GPT-4o Achiam *et al.* (2023), Claude 3 Opus Anthropic (2024), and LLaMa-3-70B Touvron *et al.* (2023). We additionally ablate over three classes of commands (Table 4.2):

1. **Search and Read.** The model can perform both search and read commands.
2. **Search Only.** The model is not allowed to read papers but can perform searches.
3. **No Commands.** The model operates with no access to the interface for actions like searching and reading.

Each class of actions is evaluated with and without demonstrations trajectories in the prompt, resulting in six configurations per LM. With three LMs, two action classes, and the option to include or exclude demonstrations, we present a total of 12 CiteAgent ablations. We exclude LLaMa with both Search and Read because its context length is limited to 8k tokens. For all experiments, we use a temperature of 0.95, following Yang *et al.* (2024a), and provide our detailed prompts in Appendix C.5.

4.5 Results

Table 4.3: Performance of LMs (using our system) and retrieval methods on CiteME, summarized.

	GPT-4o	LLaMA-3-70B	Claude 3 Opus	SPECTER2	SPECTER1
Accuracy [%]	**35.3**	21.0	27.7	0	0

[3] `https://huggingface.co/allenai/specter2`

We present the evaluation results of the CiteME benchmark in Table 4.3. Our best model, CiteAgent (GPT-4o, search and read commands, and a demonstration in the prompt) achieves 35.3% accuracy, while the previous state-of-the-art models, SPECTER2 and SPECTER, achieve 0%. Human performance on the same task is 69.7% accuracy, with less than a minute of search time, indicating that a significant 34.4% gap remains.

Table 4.4: Accuracy (in %) of LMs and retrieval methods on CiteME. We test how the available commands and prompt demonstrations affect CiteME performance. LLaMA's context window is too small and therefore incompatible with the read command.

			Method				
			GPT-4o	LLaMA-3-70B	Claude 3 Opus	SPECTER2	SPECTER
Commands	**No Commands**	w/o Demo	0	4.2	15.1	0	0
		w/ Demo	7.6	5.9	18.5	–	–
	Search Only	w/o Demo	26.1	**21.0**	26.1	–	–
		w/ Demo	29.4	2.5	27.7	–	–
	Search and Read	w/o Demo	22.7	N/A	**27.7**	–	–
		w/ Demo	**35.3**	N/A	26.1	–	–

Performance across Language Models. Comparing the performance of LMs across columns in Table 4.4, GPT-4o demonstrates the highest accuracy when it has access to both read and search commands, outperforming other LMs by a wide margin. This finding aligns with previous research Yang *et al.* (2024a), which shows that GPT-4 powered agents excel in solving software issues. Notably, GPT-4o achieves high performance across settings even though CiteME consists exclusively of samples that GPT-4o cannot predict correctly without commands; its 0% performance without commands and demonstration trajectory is by design. However, LMs outperforming the SPECTER models purely by autoregressive generation provides evidence that LMs act as implicit knowledge bases with sufficient capacity (Petroni *et al.*, 2019).

Peformance across Demonstrations. Comparing the performance between w/o Demo and w/ Demo rows in Table 4.4, we observe that LLaMA and Claude surprisingly perform worse when provided with a demonstration trajectory in the prompt. This may be due to the increased prompt length, which complicates the detection of important information Liu *et al.* (2024). LLaMA-3-70b incurs a performance drop to 2.5% due to combined history extending beyond its context length, resulting in errors. However, GPT-4o effectively utilizes demonstrations, which improves its accuracy.

Performance across Commands. GPT-4o is the only LM whose accuracy improves with access to more commands, allowing it to read full papers. CiteAgent with GPT-4o creatively uses its commands across test samples, demonstrating command behaviors not shown in the demonstration trajectory (see Figure 4.4). It frequently refines its searches based on previous results and occasionally reads multiple papers before making a selection. In contrast, Claude 3 Opus is less effective in utilizing additional commands, likely due to difficulties in detecting important information Liu *et al.* (2024).

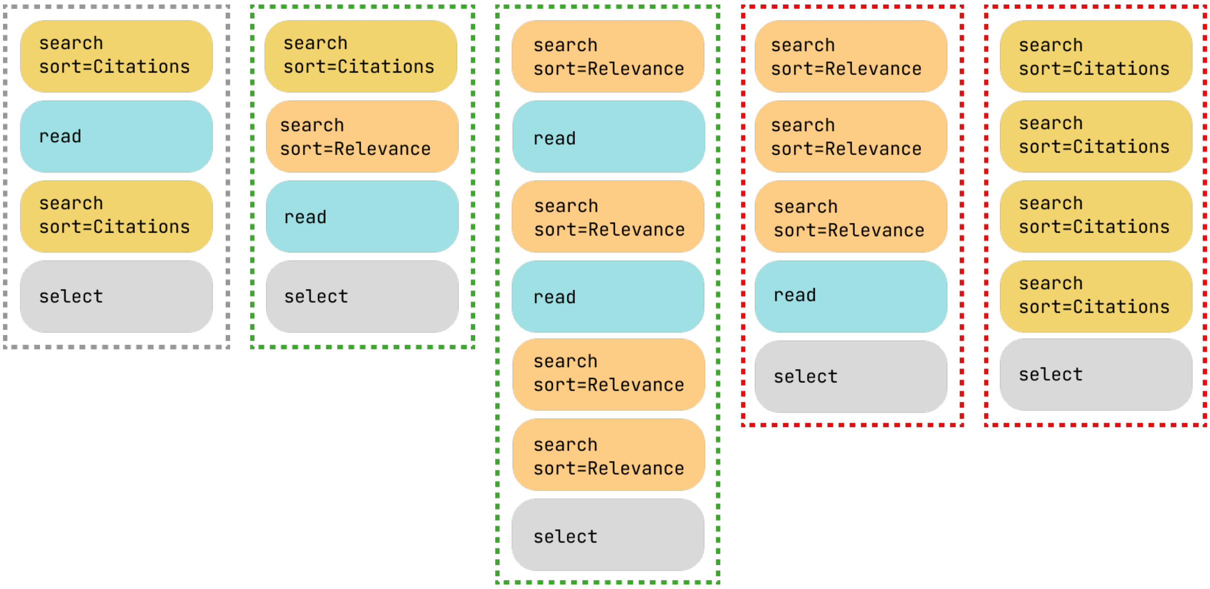


Figure 4.4: Five CiteAgent trajectories on five different samples. CiteAgent often exhibits behavior not shown in the demonstration given in the prompt, for example: searching by citation count and then by relevance, and searching multiple times in a row. Gray dotted box: prompt demonstration; green dotted boxes: CiteAgent succeeds; red dotted boxes: CiteAgent fails.

4.5.1 Error Analysis

To better identify CiteAgent's shortcomings, we analyze 50 randomly chosen CiteME samples from the best performing CiteAgent (using the GPT-4o backbone, with demonstrations, Search and Read commands) failed to solve correctly. We classify each error into three types based on CiteAgent's searches, its predicted paper and the justification provided:

Error Type 1: Misunderstands the Excerpt. This category accounts for 50% of the errors.

It occurs when CiteAgent focuses on irrelevant parts of the excerpt or omits critical details. For example, in the following excerpt:

> *The pioneering work of Reed et al. [37] approached text-guided image generation by training a conditional GAN* `[CITATION]`, *conditioned by text embeddings obtained from a pretrained encoder.*

CiteAgent searches for "Reed text-guided image generation conditional GAN" instead of "conditional GAN". It mistakes "Reed" as relevant to the current citation although it pertains to the previous one.

Error Type 2: Understands the Excerpt but Stops Prematurely. In 32% of cases, CiteAgent searches for the correct term, but it stops at a roughly matching paper instead of the exact match. For example, in the following excerpt:

> *Using Gaussian noise and blur, [CITATION] demonstrate the superior robustness of human vision to convolutional networks, even after networks are fine-tuned on Gaussian noise or blur.*

CiteAgent found a paper comparing human and machine robustness but missed that it did not cover fine-tuned networks. Notably, this paper referenced the correct target paper, meaning CiteAgent could have found the right answer with just one more step if it had properly understood the paper it was reading. Moreover, in 12.5% of such cases, the correct paper appeared in the search results but was not chosen by CiteAgent.

Error Type 3: Finds the Correct Citation but Stops Prematurely. The last 18% of errors occur when CiteAgent reads an abstract or paper and finds the correct citation; however, instead of doing another search, it selects the paper that cites the correct citation and stops searching. For example, in the following excerpt:

> `[CITATION]` *investigates transformers' theoretical expressiveness, showing that transformers cannot robustly model noncounter-free regular languages even when allowing infinite precision.*

CiteAgent finds a paper discussing the target paper and reports it, but it stops at the citing paper instead of searching for the correct target paper. For instance, it reports: *".. specifically* *mentioning Hahn's work on transformers' classification decisions becoming ineffective over longer input strings. This fits well with the description in the excerpt.."* but it selects the citing paper instead of finding Hahn's work, which is the correct target paper.

Technical Errors. Aside from comprehension errors that stem from a lack of understanding an excerpt, 5.8% of runs encountered technical issues. Occasionally, the LM formats responses incorrectly, making them unparseable by the system. Additionally, the Semantic Scholar API has inconsistencies, such as not providing open access PDF links when available or linking to non-existent web pages. Further details on these technical errors are provided in Appendix C.6.

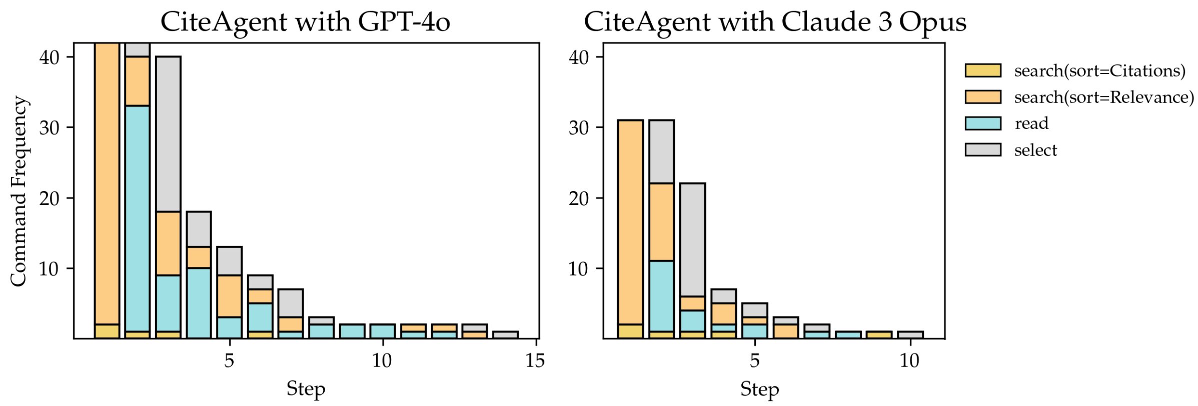


Figure 4.5: CiteAgent trajectories on samples that were correctly predicted reveals differences in model behavior. GPT-4o reads more frequently than Claude 3 Opus and can correctly predict papers even after performing many actions.

4.5.2 Analyzing the Succesful Runs

Manually examining the instances that were correctly predicted by GPT-4o and Claude 3 Opus (Figure 4.5) provides insights into how the LMs use commands they were given. First, we confirm the results presented in Table 4.4: GPT-4o frequently reads papers before it correctly predicts a citation. Second, when both LMs correctly predict a paper, they usually take just 5 steps or fewer to do so. This could stem from LMs loss of important details when given a long context window Liu *et al.* (2024).

CiteAgent's trajectories on CiteME enable us to analyze the shortcomings of GPT-4o and other SoTA LMs. These range from understanding fine details in text (Type 1 and Type 2 Errors), to not completely understanding the task (Type 3 Errors), to being unable to use commands

(Technical Errors). Correcting these errors could improve the utility of LMs on CiteME and for other related tasks.

4.5.3 Benchmarking Reasoning Capability Improvements with Latest Models

Table 4.5: Accuracy (in %) of newly released LMs on CiteME.

			Method			
			Claude-3.5-Sonnet	LLaMa-3.1-70B	o1-mini	o1-preview
Commands	**No Commands**	w/o Demo	8.4	3.4	16.0	38.7
		w/ Demo	9.2	8.4	10.9	–
	Search Only	w/o Demo	36.1	**29.4**	25.2	–
		w/ Demo	**43.7**	**29.4**	32.8	–
	Search and Read	w/o Demo	37.0	22.7	26.9	–
		w/ Demo	40.3	27.7	**34.5**	**61.3**

We compare the latest LLMs on the CiteME benchmark (Table 4.5) and find that Claude 3.5 Sonnet outperforms the previous best, Claude 3 Opus. This improvement stems from better generalization, as Sonnet achieves 9.2% without internet access, compared to Opus' 18.5%. Similarly, LLaMa-3.1-70B shows significant gains of 8% over LLaMa-3.0-70B, highlighting enhanced reasoning capabilities. However, GPT-o1, while performing well on CiteME, appears to have memorized 38.7% of the dataset, making its 61.3% benchmark performance less clear in terms of true improvement compared to GPT-4o.

4.6 Related Work

Recent work has made substantial progress in developing methods and datasets to assist researchers in paper writing and literature review Bhagavatula *et al.* (2018); Boyko *et al.* (2023); Wu *et al.* (2024b) or act as tutors (Chevalier *et al.*, 2024). Early work Learning (2011); Mayr (2014) showed that researchers automatically retrieved topics and papers considered highly relevant to their work. Other studies included methods that assist researchers in finding new ideas Gu and Krenn (2024), understanding certain topics Murthy *et al.* (2022), provide expert answers backed up by evidence (Malaviya *et al.*, 2023) or clarifying a paper's related work by supplementing it with more information and focus Chang *et al.* (2023); Palani *et al.* (2023).

Closer to our line of research, prior studies developed methods for substantiating specific claims using evidence from published papers Schuster *et al.* (2021); Wadden *et al.* (2021); Wright *et al.* (2022); Wadden *et al.* (2022); Ye *et al.* (2023); Cox *et al.* (????); Huang *et al.* (2024); Khalifa *et al.* (2024). Retrieval-augmented LMs (Lewis *et al.*, 2020; Borgeaud *et al.*, 2022; Gao *et al.*, 2023) are also popularly used to ground claims with real-world evidence (see (Mialon *et al.*, 2023) for a survey). Chen *et al.* (2023) built a web-based retrieval-augmented pipeline for fact verification; this contrasts with methods that use a static dataset for claim retrieval and verification Hanselowski *et al.* (2019); Atanasova (2024). Concurrent to this work, Ajith *et al.* (2024) build a retrieval benchmark consisting of questions about discoveries shown in specific machine learning papers.

Paper discovery is a crucial component of systems that automate scientific research as shown in Boiko *et al.* (2023); Lála *et al.* (2023); M. Bran *et al.* (2024); Miret and Krishnan (2024); Skarlinski *et al.* (2024). CiteME plays an important role in developing better tools for paper discovery, and provides a way to effectively measure their efficiency. Currently, these systems are tested as a whole, without isolating the tools responsible for scientific discovery. CiteME allows us to evaluate components within them independently – and we discover that current LM Agents are not yet ready for automated paper discovery, leading to serious gaps in end-to-end automated research pipelines.

In addition, most existing LM benchmarks are saturated, with most LMs scoring 80-95% on them Joshi *et al.* (2017); Hendrycks *et al.* (2021b); Cobbe *et al.* (2021). There is a need in the AI community to show what properties LMs currently lack, to show LM developers what aspects they should work on. On CiteME, the best LMs get less than 40%, clearly indicating to developers an important task that they could improve LMs on, while also providing an indicator they can use to track progress.

Context-aware Recommendation. Relevant to our research focus, McNee *et al.* (2002); Nallapati *et al.* (2008); He *et al.* (2010) take as input documents or parts thereof and recommend papers that are likely to be cited, often referred to as *context-aware citation recommendation*

Liu *et al.* (2015); Ebesu and Fang (2017); Yang *et al.* (2018); Färber and Sampath (2020); Jeong *et al.* (2020); Ohagi and Aizawa (2022); Gu *et al.* (2022b). The text inputs we use in CiteME resemble those used in Jeong *et al.* (2020); Ohagi and Aizawa (2022); Tang *et al.* (2023), which contain a few sentences with a masked out citation. However, CiteME differs because it uses excerpts containing only one unambiguous citation, making the context sufficient to identify the cited paper. Furthermore, our work explores agents with access to real-time paper information through tools like Semantic Scholar. This is crucial for real-time use since thousands of new papers are indexed by arXiv monthly (e.g., 8,895 papers in March 2024 under the `cs` category) arXiv (2024). Most previous approaches would be impractical due to the need for retraining with every new paper issuance.

Citation Attribution Datasets. A variety of datasets contain text excerpts from scientific papers and corresponding citations Giles *et al.* (1998); Gehrke *et al.* (2003); Bird *et al.* (2008); Huang *et al.* (2014); Radev *et al.* (2013); Kang *et al.* (2018); Jeong *et al.* (2020); Gu *et al.* (2022b). There are many crucial distinctions between the aforementioned datasets and CiteME, with the main one being that CiteME is composed of manually selected excerpts that clearly reference a paper. To our best knowledge, *CiteME is the only dataset that reports human accuracy on the benchmark.*

Additionally, the excerpts in CiteME are mostly taken from papers published in the last few years (see Figure 4.2), whereas other datasets contain older papers. For example, the arXiv dataset Gu *et al.* (2022b) includes papers from 1991-2020, and FullTextPaperRead Jeong *et al.* (2020) contains papers from 2007-2017. This currency is particularly relevant in rapidly evolving fields like machine learning. The key distinction between the dataset and methods we present compared to previous works is their *real life applicability*. Our agent is based on SoTA LMs, needs no extra training, and can use a search engine, all of which make it easily applicable to real-world settings.

CHAPTER 5

Conclusion

In this dissertation, we addressed a critical challenge in deploying neural networks: their inability to maintain robust performance when exposed to input data that differs from their training data. Variations such as shifts in lighting, object poses, and camera artifacts commonly appear in real-world inputs. Therefore, it is essential that deployed models either remain robust to these variations or effectively adapt to them without human intervention.

A significant portion of our work focused on test-time adaptation (TTA) methods for images, which improve model robustness without retraining or additional data. We explored how TTA techniques adjust pre-trained model parameters in real-time using unsupervised losses, demonstrating their potential to bridge the gap between controlled training environments and unpredictable real-world conditions. However, our analysis revealed that many existing TTA methods can lead to model degradation when subjected to prolonged adaptation, affecting their reliability in practical scenarios.

To rigorously evaluate the stability and effectiveness of TTA methods in image classification, we introduced the CCC benchmark. This comprehensive benchmark tests the resilience of adaptation techniques over extended image sequences, ensuring sustained performance across numerous model updates. Our findings indicate that most TTA methods eventually degrade in performance, highlighting the need for more robust adaptation strategies. Notably, a simple baseline approach of periodically resetting the model to its original pre-trained weights outperformed all methods evaluated.

Building on our evaluation of entropy-based TTA methods, we analyzed the mechanisms behind their initial success and subsequent failure. We found that entropy minimization initially clusters embeddings effectively, thus increasing classification accuracy. However, over time, this process can inadvertently move input image embeddings away from the original training embeddings, leading to performance declines. This precisely explains the behavior of the TTA methods evaluated on CCC.

Based on these insights, we proposed the Weighted Flips (WF) method, which predicts classifier accuracy without requiring labels or additional data. WF operates concurrently with adaptation, providing a practical tool for estimating model performance in realistic settings.

In the final chapter, we extended our investigation to language models by introducing a benchmark for literature recommendation. We demonstrated that current language models struggle to accurately infer referenced academic papers based solely on descriptive citations. By integrating an interface that enables language models to search and read papers, we showed how accuracy can be improved without retraining the models. This approach aligns with our overarching theme of enhancing model performance without additional training and with minimal additional resources.

In conclusion, this thesis demonstrates two things. First, comprehensive benchmarks that mirror realistic and challenging settings are essential to advancing the abilities of current models. Second, improving model performance on challenging benchmarks can be effictively done by giving pre-trained models tools to adapt to new inputs in real-time, without collecting additional data or performing extra training.

5.1 Future Directions

Today, machine learning models are widely used across many domains. Increasingly, language models are proving useful not only in laboratory settings but also in real-world applications. For example, SWE-Agent Yang *et al.* (2024a) is a language model-based agent capable of solving GitHub issues found in real code repositories Jimenez *et al.* (2023).

By developing new benchmarks that are realistic and relevant to real-world settings, we can clearly identify the gaps in language model capabilities. SWE-Bench Multimodal Yang *et al.* (2024b), for instance, demonstrates that language-model agents struggle with GitHub issues containing reference images; Tian *et al.* (2024) demonstrates the difficulty language models have in writing code based on algorithms from scientific papers, and Yoran *et al.* (2024) shows how web agents remain unreliable for slightly complex yet realistic queries. By thoroughly analyzing these capability gaps, we can continue enhancing the practical utility of language models.

CHAPTER A

Chapter 2 Appendix

A.1 2D Example Experiments and Analysis

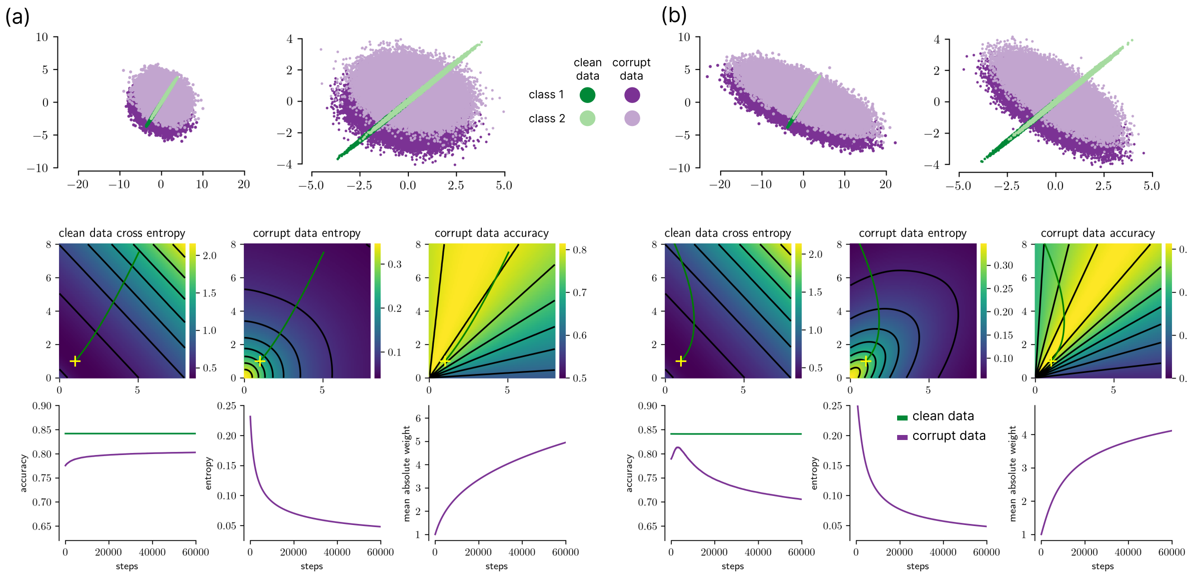


Figure A.1: Theoretical analysis of adaptation under distribution shift. Collapsing or non-collapsing behavior of entropy minimization can be reproduced with a simple 2d Gaussian binary classification example, a domain shift which slightly rotates the data and adds Gaussian noise, and a model which consists of a batch norm layer followed by logistic regression. **Top:** clean and corrupt data for two classes before (a) and after (b) batch norm. **Middle:** learning dynamics of entropy minimization in the 2d adaptation parameter space starting from initial parameters (yellow marker) over time. **Bottom:** accuracy, entropy, and size of adaptation weights over time.

In order to better understand collapse, we constructed a simple 2D Gaussian binary classification example.

Data. The 2D datasets are constructed as follows: the data is sampled by drawing sampled from two Gaussian blobs with identical variance corresponding to the two classes $\mathcal{N}(\mu_i, \Sigma)$. For

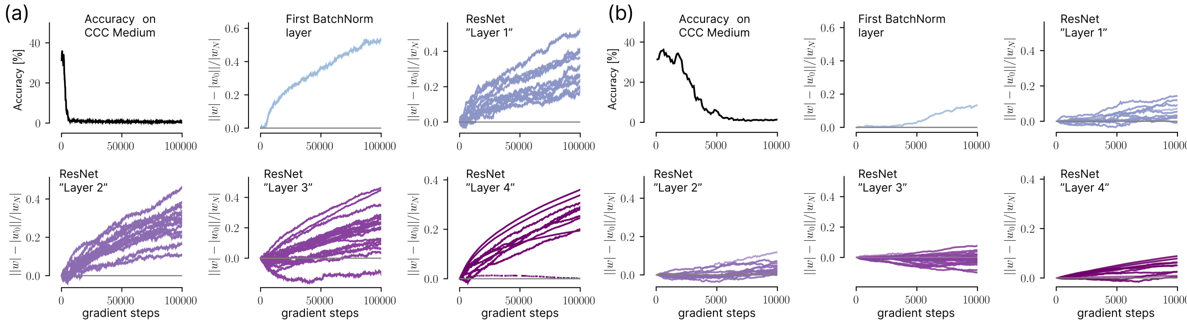


Figure A.2: Analysis of entropy minimization collapse on real data. (a) Consistent with the theoretical analysis in Fig. A.1, we find that the adaptation weights in all layers increase over time continually, even long after the collapse as indicated by Accuracy on CCC-Medium has happened. (b) shows a zoomed-in view where this increase is not yet apparent, well after the collapse.

the corrupted case, the data is then rotated by an angle θ and combined with additional additive Gaussian noise $\mathcal{N}(0, \Sigma_{\text{corrupt}})$. Finally, both in the clean and the corrupt data case, the data is rotated by an angle of $-\frac{\pi}{4}$ (which minimizes the effect of the batch normalization in the model).

$$Y \sim \text{Ber}(0.5) \tag{A.1}$$

$$X \sim \mathcal{N}(\mu_Y, \Sigma) \tag{A.2}$$

$$X_{\text{clean}} = R_{-\frac{\pi}{4}} X \tag{A.3}$$

$$X_{\text{corrupt}} \sim \mathcal{N}(R_{\theta_{c_1}} X_{\text{clean}}, R^{\top}_{-\frac{\pi}{4}} R^{\top}_{\theta_{c_2}} \tilde{\Sigma} R_{\theta_{c_2}} R_{-\frac{\pi}{4}}) \tag{A.4}$$

$$\text{with } \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}, \quad \sigma_1 \gg \sigma_2 \tag{A.5}$$

$$\tilde{\Sigma} = \begin{pmatrix} \tilde{\sigma}_1 & 0 \\ 0 & \tilde{\sigma}_2 \end{pmatrix}, \quad \tilde{\sigma}_2 \geq \tilde{\sigma}_1 \tag{A.6}$$

$$R_\theta = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \tag{A.7}$$

Note that for the clean data, we allocate most of the variance to the class dimension, σ_1. On the corrupted data, we add noise primarily perpendicular to the class dimension ($\tilde{\sigma}_2$), which is the main defining factor whether or not we observe collapsing behavior. $\theta_{c_1} \neq \theta_{c_2}$ serves to break the symmetry between signal and noise in the data, which results in Tent starting to deviate towards the noise direction.

In our example, we parametrize this model as follows: In condition 1, $\mu_i = (\pm 1, 0)$ and $\Sigma^2 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0.03 \end{smallmatrix}\right)$, $\theta = -\frac{\pi}{9}$ and $\Sigma^2_{\text{corrupt}} = R^T_{\theta_c} \left(\begin{smallmatrix} 0.25 & 0 \\ 0 & 4 \end{smallmatrix}\right) R_{\theta_c}$, where R_{θ_c} is the rotation matrix for an angle of $\theta_c = -\frac{\pi}{6}$. In condition 2, everything is the same except for $\Sigma^2_{\text{corrupt}} = R^T_{\theta_c} \left(\begin{smallmatrix} 0.25 & 0 \\ 0 & 25 \end{smallmatrix}\right) R_{\theta_c}$. In both conditions, we sample 300 000 data points which are equally distributed over both classes.

Model. The classification model consumes the dataset of shape $N \times 2$ and consists of a batchnorm layer ($\varepsilon = 0$) followed by a fully connected layer with two input channels, one output channels, no bias term and a sigmoid nonlinearity. Because our data is always centered, we do not learn the offset parameter of the affine adaptation in the batchnorm layer, but only the scale parameter.

Model training. The model is trained to minimize the binary cross-entropy on the clean data. We use batch gradient descent on the whole 300,000 sample dataset with a learning rate of 0.1 and no momentum. We decay the learning rate by a factor of 0.1 after 1000 and 2000 steps and stop training after 3000 steps.

Model Learning and Adaptation. We adapt the model on the corrupted data using Tent. More precisely, we optimize the scale parameter of the batch norm layer to minimize the entropy of the predictions using SGD with a learning rate of 0.01 and no momentum. We process the whole dataset in one batch and adapt for 60 000 steps.

Results. In the toy model, simple cases emerge where the loss does not result in collapse, or vice versa (Fig. A.1a and Fig. A.1b, respectively), mainly depending on the relation of signal and noise variances and directions.

The toy example furthermore predicts that the adapted parameters of a model should grow on the long run and indeed we were able to find exactly this effect when running ETA on a ResNet50 on CCC-Medium (Fig. A.2), suggesting that our minimal setup successfully reproduces the relevant aspects of the large scale case. However, the weight explosion becomes apparent only after the collapse happens, hence weight regularization is not enough to avoid the collapse (Fig. A.2(b)).

In the Fig. A.1(a) example, we find the direction of high target domain performance and stay there; in this case entropy minimization is stable. In the Fig. A.1(b) experiment, the domain shift adds more noise nearly orthogonal to the signal direction, which entropy minimization tries to use for making high confidence predictions: we still initially find the direction of high target domain performance, but traverse this region and continue into a direction of low entropy and low accuracy. This shows that even in a linear example, entropy minimization can show initial performance improvement and then collapse.

A.2 Path Finding Algorithm

Algorithm A.1 describes the pseudo code of the algorithm used to generate CCC. The algorithm is based on a set of Calibration Matrices, similar to the one shown in Figure 2.2. There exists a matrix m for every (n_a, n_b) pair, such that $m[i][j]$ is equal to accuracy of a pretrained ResNet-50 on the comination of noises (n_a, n_b), and severities $(i/5, j/5)$. We will release the full set of matrices upon publication.

Additionally, Algorithm A.2 uses a function $MinValidPath(s_1, s_2)$: this function returns the minimum path that starts at (s_1, s_2) and ends at $(0, s_j)$ for some s_k. The cost of a path is simply the average of all entries along the path. The minimum path is defined as the path with a cost closest to b_a in absolute terms. Lastly, a path is only valid if it starts with s_2 equal to 0, every transition either decreases s_1 by 0.25, or increases s_2 by 0.25, and stops once s_1 is equal to 0.

Algorithm A.1: Data Generation for CCC

```
function GenerateCCC(ba, k, T):
    # Baseline accuracy (ba), transition speed (k),
    # and total split size (T).
    t = 0

    # 1) Initialize the first two corruptions
    c1 = Uniform({1..15})
    c2 = Uniform({1..15})

    # 2) Calculate path (closest avg accuracy to ba)
    path = CalculatePath(c1, c2, ba)

    # Start index for path
    p = 0

    # 3) Generate until total T
    while true:
        # Extract severities from path
        s1, s2 = path[p]

        # Pick a random subset of ImageNetVal
        Subset = Uniform(ImageNetVal)

        # Apply chosen corruptions & severities
        apply_corruptions(Subset, (c1, s1), (c2, s2))

        # Save the corrupted subset
        save(Subset)

        # 4) Accumulate total images
        t = t + k
        if (t >= T):
            break

        # 5) If end of path, pick a new corruption
        if (p == length(path) - 1):
            c1 = c2
            c2 = Uniform({1..14})
            path = CalculatePath(c1, c2, ba)
            p = 0
        else:
            p = p + 1
end function
```

Algorithm A.2: CalculatePath

```
function CalculatePath(c1, c2, ba):
    # The two noises c1, c2, and the baseline accuracy ba

    m = CalibrationMatrix[c1][c2]
    (MinPath, MinCost) = MinValidPath(0, 0, m, ba)

    # Loop over severities in {0.25, 0.5, 0.75, ..., 5}
    for s1 in {0.25, 0.5, 0.75, ..., 5}:
        m = MinValidPath(1, 0, m, ba)
        (MinPath, MinCost) = MinValidPath(s1, 0, m, ba)

    return MinPath
end function
```

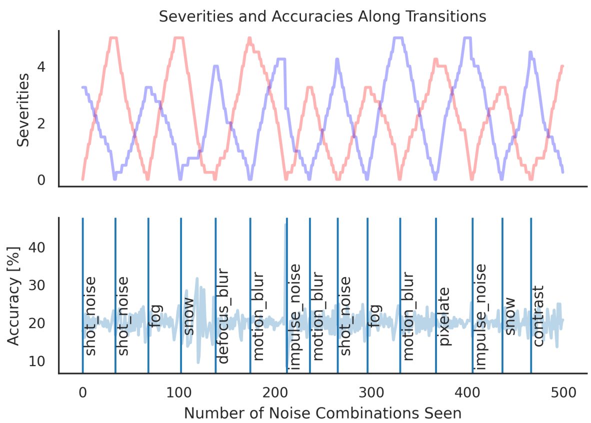
The resulting dataset features transitions between two different noises like the ones seen in Figure A.3 (a). We additionally plot the accuracy of a pretrained ResNet-50, alongisde the severities of the different noises in Figure A.3 (b).

We additionally share the following metadata about the length of traversals in CCC-Easy/Medium/Hard:

Table A.1: CCC traversal length statistics, for each CCC split.

	Min	Max	Mean	Median
CCC-Easy	11	36	22.8	23
CCC-Medium	21	41	33.9	34
CCC-Hard	41	41	41	41

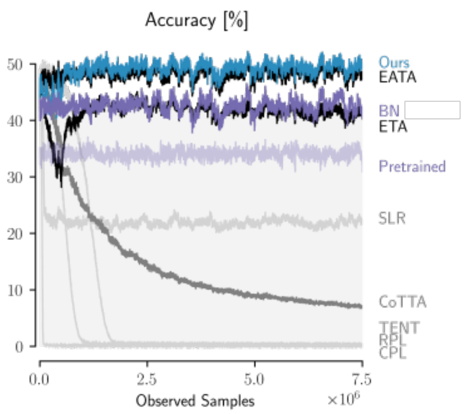
(a)



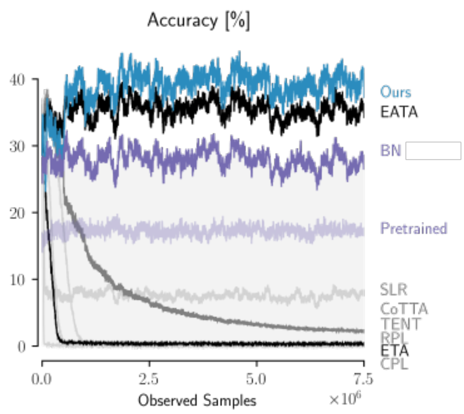
(b)

Figure A.3: **(a)** Visualization of CCC-Medium's smooth transition between Zoom Blur to Snow. Note: CCC additionally uses random flips and crops, which are not shown here. **(b)** As CCC transitions between noises, the severities of the first noise (red), and the second noise (blue) go up and down correspondingly, in order to keep the accuracy of a pretrained model stable.

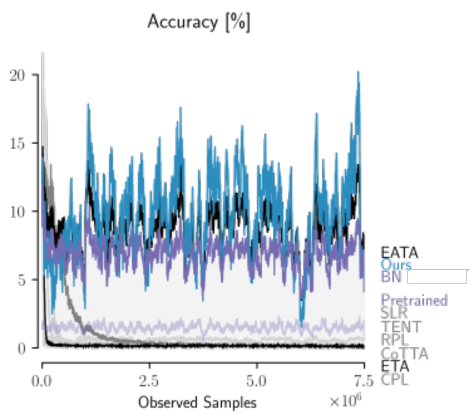
A.3 CCC Plots



(a) CCC Easy.



(b) CCC Medium.



(c) CCC Hard.

Figure A.4: Adaptation performance of all evaluated models using a ResNet-50 backbone. For all subplots, model performances are averaged over the 9 runs of the respective difficulty level.

A.4 EATA Implementation and Ablations

Our implementation of EATA differs from the official implementation[1]. The reason for this is that the official implementation uses clean ImageNet validation images to calculate the Fisher vector matrix for its regularizer[2]. This stands in contradiction with the method, which should not have access to the training distribution at test time, as shown in the paper in Table 1.

Instead of using $2,000$ ImageNet validation images, we calculate the Fisher matrix using the first $2,000$ images in our data stream. We conduct a hyperparameter search on the weight regularizer tradeoff parameter β:

β	25	50	100	250	500	1000	1500	2000
Acc. [%]	46.5	46.9	**47.1**	46.7	46.1	45.6	44.8	44.0

Table A.2: Accuracy of EATA on CIN-C holdout noises for different values of the weight regularizer loss.

Using the optimal value, 100 led to worse results than the default value, 2000, on CCC:

	CIN-C	CCC-Easy	CCC-Medium	CCC-Hard	CCC Avg
EATA-100	46.7	47.7	36.5	3.8	29.3
EATA-2000	41.8	48.2	35.4	8.7	30.8
Ours	46.5	49.3	38.9	9.6	32.6

Table A.3: Accuracy of EATA on CIN-C holdout noises for different values of the Fisher alpha.

In the end, we used the original value of 2000, as that was optimal on the CCC dataset.

In addition, we conducted a hyperparameter search for EATA on a ViT backbone. As shown in 2.4, EATA performs worse than a pretrained, non adapting baseline in this setting. To that end, we tried to stabilize the model by increasing the value of β, the hyperparameter that controls the weight of the anti-forgetting regularizer. As with the previous experiment, the original value of 2000 is optimal.

[1] `https://github.com/mr-eggplant/EATA`, version f739b3668c

[2] `https://github.com/mr-eggplant/EATA/blob/f739b3668c/main.py#L144`

β	2000	3000	4000
Acc. [%]	**38.5**	27.8	16.5

Table A.4: Accuracy of EATA on CCC-Medium using a ViT backbone, for different values of the regularizer, β.

A.5 Novelty of Resetting

Our work is the first to propose resetting to solve collapse in TTA methods. Notably, while prior work Wang *et al.* (2020b); Zhang *et al.* (2022); Niu *et al.* (2022) has briefly touched upon the concept of episodic resetting, the methodology and its application is significantly distinct and unrelated to collapse in TTA.

- **Tent** Wang *et al.* (2020b) mentions episodic in the context of overfitting to a single sample in segmentation (similar to Zhang *et al.* (2022)'s overfitting to a single sample). Resetting here is unrelated to collapse, as the paper doesn't discuss collapse at all.
- Although **MEMO** Zhang *et al.* (2022) uses resetting, it does so because it overfits to one image (and its augmentations) every step. MEMO doesn't discuss collapse or catastrophic forgetting. MEMO compares itself to a version of Tent that resets after every step (which they call Tent + episodic resetting), because MEMO without augmentations is similar to Tent + episodic resetting with a batch size of 1. (Note: MEMO is outperformed by BN/Tent/ETA when using the standard batch size of 64)
- **EATA** Niu *et al.* (2022) shows results for Tent + episodic resetting after every step in its tables, but provides no reasoning or discussion for doing this. Tent + episodic resetting is outperformed by regular Tent.

A.6 CIFAR10 Experiments

We conduct CIFAR10 experiments to show the need for ImageNet scale benchmarks. Tent, without an anti-collapse mechanism, does not collapse on CIFAR10-C, even after seeing 100 million images Fig. A.5a,b. Like ImageNet, CIFAR10-C's noises also exhibit high variance in difficulty Fig. A.5c.

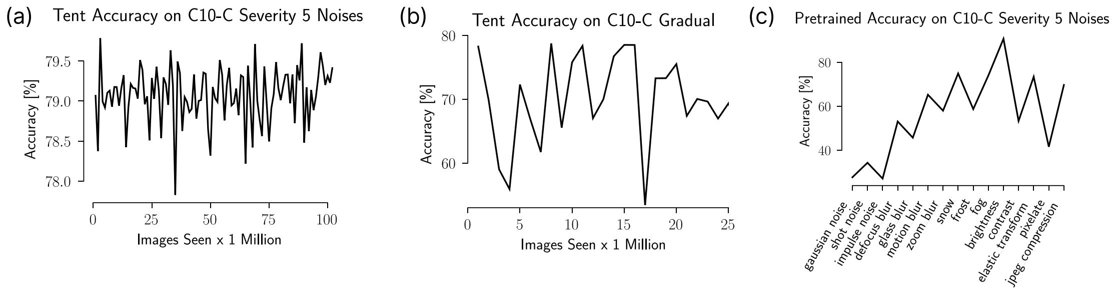


Figure A.5: **(a)** When tested on an infinite concatenation of severity 5 noises, Tent does not collapse even after seeing 100M CIFAR scale images. **(b)** Tent does not collapse to chance level when tested on a long term variant of CIFAR10-C gradual. **(c)** CIFAR10-C exhibits great variations between individual corruptions, similar to ImageNet-C.

A.7 Compute details

We conduct all experiments on Nvidia RTX 2080 TI GPUs with 12GB memory per device. All experiments except our study on larger models were conducted on a single GPU. For CoTTA experiments, we use data parallel training on 2 GPUs. A bulk of the compute spent for this work was on computing baseline accuracies on the calibration dataset, which contains 463M images.

A.8 Software and Dataset Licenses

A.8.1 Datasets

- ImageNet-C (Hendrycks and Dietterich, 2019a): Creative Commons Attribution 4.0 International,

  `https://zenodo.org/record/2235448`

- ImageNet-C (Hendrycks and Dietterich, 2019a), code for generating corruptions: Apache License 2.0

  `https://github.com/hendrycks/robustness`

- ImageNet-3D-CC (Kar *et al.*, 2022): CC-BY-NC 4.0 License

  `https://github.com/EPFL-VILAB/3DCommonCorruptions`

A.8.2 Models

- PyTorch's (Paszke *et al.*, 2019) Backbones

  `https://pytorch.org/vision/stable/models.html`

- Adaptive BN (Schneider *et al.*, 2020; Nado *et al.*, 2020):

  Apache License 2.0, `https://github.com/bethgelab/robustness`

- Tent (Wang *et al.*, 2020b): MIT License,

  `https://github.com/DequanWang/tent`

- RPL (Rusak *et al.*, 2021): Apache License 2.0,

  `https://github.com/bethgelab/robustness`

- CoTTA (Wang *et al.*, 2022): MIT License,

  `https://github.com/qinenergy/cotta`

- CPL Goyal *et al.* (2022): MIT License,

  `https://github.com/locuslab/tta_conjugate`

- EATA Niu *et al.* (2022): MIT License

  `https://github.com/mr-eggplant/EATA`

CHAPTER B

Chapter 3 Appendix

B.1 The Relationship between Entropy Minimization and Clustering

In this section, we explain the connection between entropy minimization and the Expectation-Maximization algorithm Dempster *et al.* (1977) with a mixture of Gaussians and show how the iterative entropy minimization objective leads to a clustering process similar to the Expectation-Maximization algorithm.

In the Expectation-Maximization algorithm for clustering, the latent variables represent the cluster assignments, and the algorithm alternates between estimating the cluster assignments (E-step) and updating the cluster parameters (M-step). The convergence of the EM algorithm in this setting has been formally established Dempster *et al.* (1977).

Poland and Shachter (1993) showed that for a random variable X with a given distribution and the mixture of random variables Y that derive from it, the objective of minimizing the "relative entropy" between X and Y generalizes the objective of the Expectation Maximization algorithm: to maximize the likelihood of the observations x drawn from Y's distribution.

In our setting, the iterative entropy minimization process corresponds to the Expectation Maximization algorithm, as iterative entropy minimization can also be seen as a form of "self-training" with minimization of the relative entropy (the DKL Kullback and Leibler (1951)) of the pseudo-labels (the model's predictions) Grandvalet and Bengio (2004). The forward pass of our training process serves two purposes: (1) it sets the "observations", which are the model's predictions, and (2) it acts as the E-step of the algorithm, estimating the distribution given the model parameters (the clustering assignment). The backpropagation step, which updates the model parameters (the cluster parameters), serves as the M-step and maximizes the likelihood under the current pseudo-label estimates Amini and Gallinari (2002). It is important to note that in our setting, the entropy minimization procedure involves changing both X and Y in each iteration, which may be different from the original Expectation Maximization algorithm.

Using these insights, we can provide a better explanation for the two-phase clustering phenomenon observed in our experiments. In the initial "success" phase, where the change in the embeddings is relatively small during the process, the entropy minimization effectively performs Expectation Maximization unsupervised clustering in the model's embedding space, guided by the smart initialization provided by the pre-trained model. The E-step estimates the pseudo-labels based on the current embedding structure, while the M-step updates the model to refine the embeddings and increase intra-cluster similarity. This process leads to the formation of well-separated clusters, as reflected by the increasing Silhouette score.

However, as the Expectation Maximization algorithm continues over many iterations in the "failure" phase or if there is bad initialization, it starts to overfit the model to the specific characteristics of the "new" test data. Unlike the regular Expectation Maximization algorithm, in our case, the data distribution (the observations) changes over time, which leads to a drift in the embeddings away from the initialized representations learned from the training data. This overfitting effect, which might even converge to a global minimum, is captured by the increasing Shift distance between the test data embeddings and the training class embeddings.

To support this explanation, we also provide visualizations of the prediction space to illustrate the clustering process and the eventual drift from the training embeddings. We used a mixture of Gaussians, and trained a GMM with the Expectation Maximization algorithm using maximum likelihood, where the means are initialized based on random samples. The covariance is used as the identity matrix, with the input samples being trainable and optimizing their location. In Figure B.1, each dot represents a sample colored by its original class, where the Xs are the centroids at each iteration. As we can see, with the "smart initialization" of the cluster centers, the points converge to the "right" clusters based on the original cluster centers. However, when we start the cluster centers with some shift, namely there is "wrong" initialization, the clusters start with good clustering but then converge to wrong solutions where they mix points with different classes.

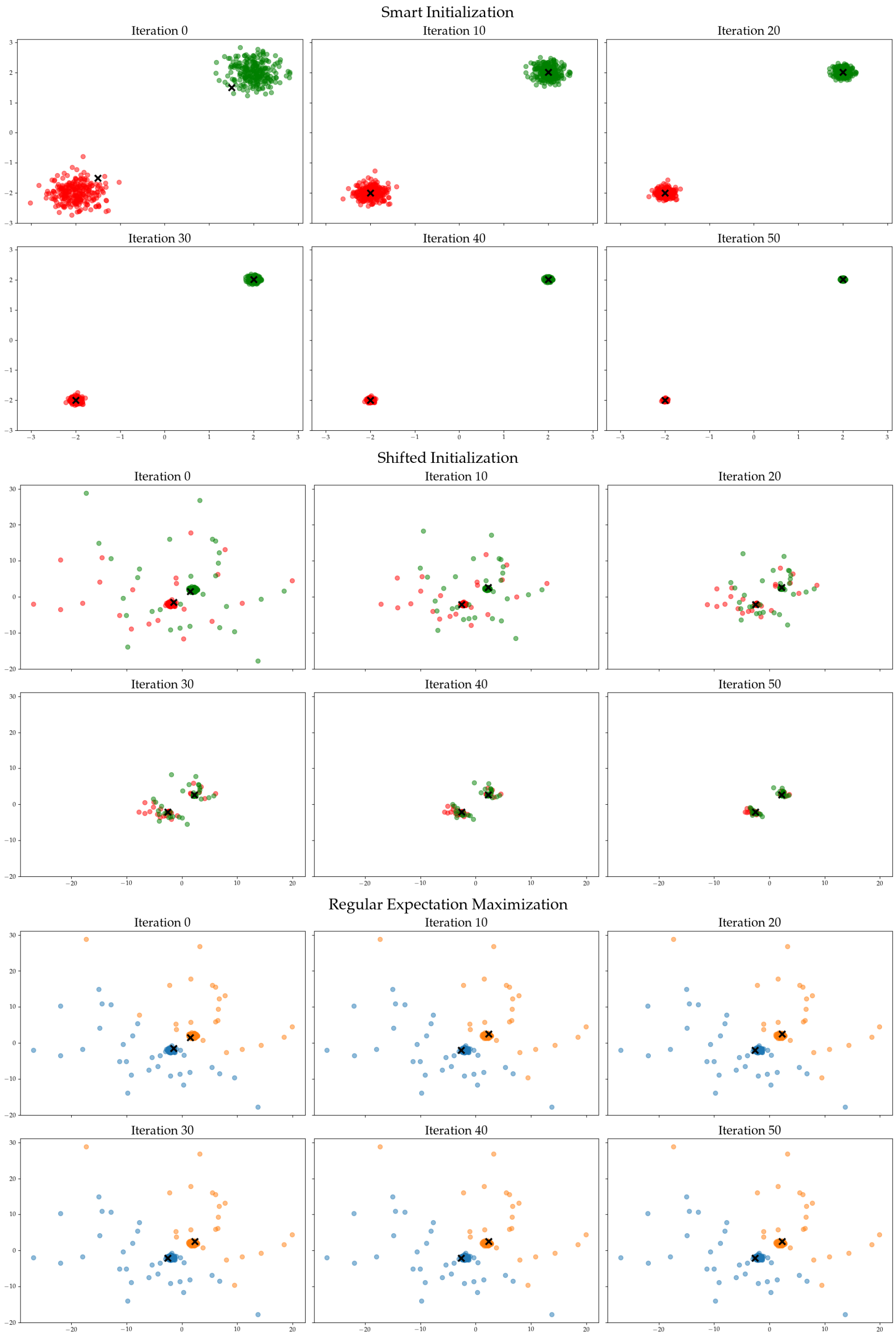


Figure B.1: **Top:** With the "smart initialization" of the cluster centers, the points converge to the "right" clusters based on the original cluster centers. **Middle:** When we start the cluster centers with some shift, namely there is "wrong" initialization, the clusters start with good clustering but then converge to wrong solutions where they mix points with different classes. **Bottom:** For reference, we also show the regular Expectation Maximization algorithm on the shifted dataset. The X's represent cluster centroids at each iteration.

B.2 Different Parameterizations of f

In this section, we test the different ways of parameterizing the weighted-flips-to-accuracy function, f. Firstly, we look at the effects of not weighing each flip, and then we look at linear and cubic interpolations between flips and accuracy (as opposed to quadratic interpolation, used in the rest of the paper). Our results in Table B.1 show that the optimal f is a weighted and interpolated quadratically, with the other variations not far behind. Importantly, all variations of f perform better than the second best performing method, COT Lu *et al.* (2023).

Table B.1: Mean Absolute Error between estimated accuracy, and true accuracy on a ResNet-50 model, for weighted and unweighted flips-to-accuracy functions, that are either linear, quadratic, or cubic interpolations of points.

Datasets	Unweighted Linear	Unweighted Quadratic	Weighted Linear	Weighted Quadratic	Weighted Cubic
Noises					
IN-C {75} 94	<u>4.95</u>	5.04	5.94	**4.79**	5.23
IN-$\overline{\text{C}}$ {50} 156	<u>7.19</u>	7.36	7.94	7.35	**7.01**
IN-3DCC {60} 115	<u>4.10</u>	4.12	4.33	**3.66**	4.25
CCC {27} 181	<u>2.97</u>	3.22	4.8	**2.80**	4.34
Domain Shifts					
Stylized 73	<u>7.12</u>	<u>7.12</u>	<u>7.12</u>	**3.81**	<u>7.12</u>
IN-V2 {3} 187	**3.55**	<u>3.71</u>	5.42	4.70	4.03
IN-Sketch 239	<u>1.11</u>	1.32	2.64	4.23	**0.23**
IN-R 99	<u>1.43</u>	1.67	3.01	1.88	**0.52**
IN-D 194					
→ Real	3.39	<u>3.16</u>	**2.04**	3.18	4.70
→ Painting	2.07	1.94	**0.34**	2.20	<u>0.85</u>
→ Clipart	<u>2.78</u>	3.08	5.12	3.37	**2.44**
→ Sketch	<u>6.12</u>	6.95	12.89	**5.44**	12.38
→ Infograph	<u>7.28</u>	8.76	10.35	**3.63**	10.35
→ Quickdraw	**0.79**	**0.79**	**0.79**	<u>2.57</u>	**0.79**
Cartoon & Drawing {2} 198	13.60	13.76	14.34	<u>13.25</u>	**12.96**
Adversarial Noises					
BG Challenge {8} 247	<u>7.19</u>	7.36	7.33	<u>6.92</u>	8.50
IN-A 101	23.70	23.53	**20.39**	<u>21.61</u>	22.91
IN-C Patch {75} 84	1.95	2.00	2.42	<u>1.60</u>	**1.48**
IN-Hard 220	5.27	4.92	**0.72**	3.64	<u>3.49</u>
Patch-IN {10} 175	**7.42**	<u>7.55</u>	9.02	8.87	7.98
IN-Obfuscations {3} 212	<u>0.20</u>	**0.10**	**0.10**	4.58	**0.10**
OOD/Other					
ObjectNet 14	<u>6.81</u>	<u>6.81</u>	<u>6.81</u>	**2.74**	<u>6.81</u>
NINCO 22	20.20	19.85	**14.98**	18.07	<u>17.73</u>
Average	<u>6.14</u>	6.27	6.47	**5.75**	6.36
Worst Case	23.70	23.53	**20.39**	<u>21.61</u>	22.91
Average (Worst Case Excluded)	<u>5.34</u>	5.48	5.84	**5.03**	5.60

B.3 WF with Limited Data

To further test WF's ability in a challenging setting, we look at how it performs under memory and data constraints. To this end, we test WF in the following scenarios: (1) WF is only allowed to store 100 samples for calculating flips, and (2) when whole dataset is limited to 100 samples for flip calculation and 1,000 samples for adaptation). We note that previous work assumes the existence of at least 2,000 test samples Niu *et al.* (2022). In both cases, we use the original weighted-flips-to-accuracy function, f, by multiplying the the weighted flips calculated on 100 samples by 10, and plugging the output into f. Even with only using 100 samples, WF is able to best the original implementation by a bit. Surprisingly, even with limited data and memory, WF manages to remain competitive with unconstrained methods, and is significantly ahead of COT, when it is constrained in a similar manner.

Table B.2: WF is effective in memory constrained settings. Without finetuning or refitting f, WF beats the original implementation, when only using 100 samples to calculate weighted flips (WF limited mem). In the limited memory/data setting, WF gets access to only 1000 samples in total, 100 of which are used for flip calculations. In this setting, COT gets access to 1,000 input samples and 1,000 in distribution samples. **Best** results are in bold; <u>second best</u> are underlined, $\{.\}$ indicates how many splits are in each dataset, when there are more than 1.

Datasets	COT original	WF original	WF limited mem	COT limited mem/data	WF limited mem/data
Noises					
IN-C {75} Hendrycks and Dietterich (2019a)	**2.23**	<u>4.79</u>	7.52	36.67	6.52
IN-$\overline{\text{C}}$ {50} Mintun *et al.* (2021)	**3.17**	7.35	8.34	40.55	<u>4.60</u>
IN-3DCC {60} Kar *et al.* (2022)	**3.02**	<u>3.66</u>	3.97	34.44	4.31
CCC {27} Press *et al.* (2023)	2.04	2.80	3.71	26.67	4.92
Domain Shifts					
Stylized Geirhos *et al.* (2019b)	12.18	<u>3.81</u>	<u>3.37</u>	38.84	**2.50**
IN-V2 {3} Recht *et al.* (2019)	**2.68**	4.70	4.00	43.96	<u>3.80</u>
IN-Sketch Wang *et al.* (2019)	4.23	<u>1.71</u>	**1.68**	12.46	3.39
IN-R Hendrycks *et al.* (2021a)	<u>2.44</u>	**1.88**	3.03	14.99	12.03
IN-D Rusak *et al.* (2022b)					
↦ Real	27.54	<u>3.18</u>	**1.73**	41.52	6.51
↦ Painting	7.49	<u>2.12</u>	**0.71**	26.21	18.44
↦ Clipart	<u>4.52</u>	**3.37**	5.91	15.98	8.10
↦ Sketch	**0.71**	5.44	6.30	12.65	<u>4.50</u>
↦ Infograph	3.44	3.63	**1.24**	4.57	<u>2.51</u>
↦ Quickdraw	<u>1.60</u>	2.57	2.80	**0.06**	2.46
Cartoon & Drawing {2} Salvador and Oberman (2022)	**1.62**	<u>13.25</u>	16.48	33.25	13.44
Adversarial Noises					
BG Challenge {8} Xiao *et al.* (2020)	19.68	<u>6.92</u>	**5.84**	32.84	10.15
IN-A Hendrycks *et al.* (2021c)	30.38	21.61	<u>16.75</u>	**15.30**	29.15
IN-C Patch {75} Gu *et al.* (2022a)	2.57	**1.60**	1.98	47.03	<u>1.92</u>
IN-Hard Taesiri *et al.* (2023)	15.33	<u>3.64</u>	**0.65**	5.83	14.73
Patch-IN {10} Pintor *et al.* (2023)	10.13	**8.87**	<u>9.09</u>	49.68	9.81
IN-Obfuscations {3} Stimberg *et al.* (2023)	<u>0.12</u>	4.58	4.67	**0.09**	8.93
OOD/Other					
ObjectNet Barbu *et al.* (2019)	10.40	2.74	**0.29**	<u>2.44</u>	2.74
NINCO Bitterwolf *et al.* (2023)	20.28	<u>18.07</u>	20.24	**13.05**	35.68
Average	8.17	<u>5.75</u>	**5.67**	23.87	9.40
Worst Case	30.38	<u>21.61</u>	**20.24**	49.68	35.68
Average (Worst Case Excluded)	7.16	<u>5.03</u>	**5.00**	22.70	8.21

B.4 Weighted Flips Ablations

B.4.1 Stopping Iteration Ablations

WF measures the amount of weighted flips from iteration 0 to iteration 1,000. This is done because RDumb resets the model to its pretrained state every 1,000 iterations, in order to avoid collapse Press *et al.* (2023). Here, we look at how measuring weighted flips before iteration 1,000 affects the performance of WF. Interestingly, using 500 iterations increases performance by a relative 26.89% as opposed to the 1,000 iterations used in the rest of the paper.

Datasets	Stopping Iteration 1000	500	250	100	50
IN-C	4.79	4.88	4.99	5.48	5.67
IN-$\overline{\text{C}}$	7.35	7.48	7.84	8.96	10.1
IN-3DCC	3.66	3.32	3.37	3.00	3.06
IN-V2	4.70	4.59	4.93	5.11	5.49
IN-D ⮯					
Real	3.18	0.36	0.96	2.82	5.01
Painting	2.12	1.21	4.02	11.28	14.8
Clipart	3.37	0.31	3.75	9.30	14.8
Sketch	5.44	2.49	0.33	5.61	9.26
Infograph	3.63	3.46	0.09	4.37	7.53
Quickdraw	2.57	1.73	7.55	17.54	25.3
Average	4.08	**2.98**	3.78	7.35	10.12

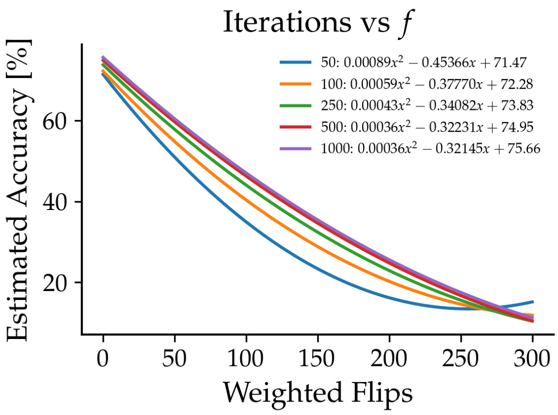


Figure B.2: **Left:** Mean Absolute Error between estimated accuracy and true accuracy, when measuring weighted flips between iteration 0 and various stopping iterations. **Right:** For different stopping iterations, interpolating between the points in the holdout set yields different weighted-flips-to-accuracy functions.

B.4.2 Holdout Set Size Ablations

Datasets	Holdout Set Size 1000	500	250	100	50
IN-C	4.79	4.77	5.52	6.07	7.6
IN-$\overline{\text{C}}$	7.35	5.91	6.74	7.06	8.4
IN-3DCC	3.66	5.10	4.34	5.17	5.2
IN-V2	4.70	5.13	5.48	3.47	9.1
IN-D ⮯					
Real	3.18	3.50	1.78	0.57	4.2
Painting	2.12	5.38	1.92	6.93	8.5
Clipart	3.37	6.69	7.96	8.79	7.7
Sketch	5.44	10.23	7.39	4.87	4.1
Infograph	3.63	6.15	0.88	3.05	3.7
Quickdraw	2.57	0.71	9.87	38.86	31.0
Average	**4.08**	5.36	5.19	8.48	9.04

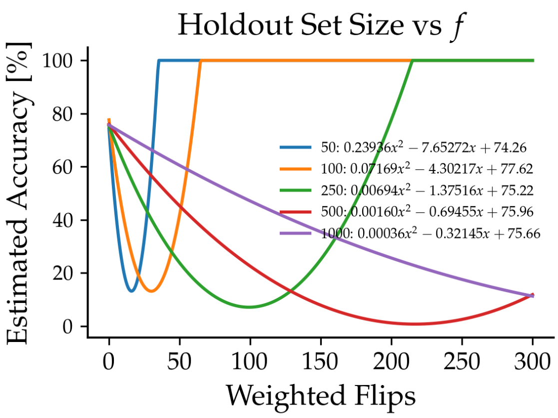


Figure B.3: **Left:** Mean Absolute Error between estimated accuracy and true accuracy, when measuring weighted flips on sets of images of different sizes. **Right:** For different holdout set sizes, interpolating between the points in the holdout set yields different weighted-flips-to-accuracy functions. f can only output values that are between 0 and 100.

B.5 WF with other TTA Methods

WF esimates the accuracy of a dataset as RDumb Press *et al.* (2023) is used to adapt to it. In this section, we show that WF work with a variety of different EM methods. To further showcase the versatility of WF, we do not finetune any method, and use the original weighted-flips-to-accuracy function f, for all experiments in Table B.3.

Table B.3: Mean Absolute Error between estimated accuracy and true accuracy, when adapting to data using a ResNet-50 backbone and different TTA methods: Tent Wang *et al.* (2020b), RPL Rusak *et al.* (2022a), and CPL Goyal *et al.* (2022). In all cases, the original weighted-flips-to-accuracy function f is used, highlighting the versatility of WF.

Datasets	RDumb	Tent	RPL	CPL
IN-C	4.79	6.75	6.85	5.14
IN-$\overline{\text{C}}$	7.35	9.68	7.20	7.44
IN-3DCC	3.66	2.92	3.99	3.72
IN-V2	4.70	3.80	3.82	4.42
IN-D ↴				
Real	3.18	5.15	5.15	0.31
Painting	2.12	7.59	7.59	0.03
Clipart	3.37	6.98	7.11	2.57
Sketch	5.44	3.30	3.52	3.86
Infograph	3.63	2.29	2.24	3.40
Quickdraw	2.57	2.24	2.24	2.37
Average	4.08	5.07	4.97	**3.33**

B.6 Additional Vision Transformer Experiments

To further analyze WF and the second best method, COT, we add additionally analysis using a ViT-B/16 model. The task is to estimate the accuracy of a ViT-B/16 on a variety of datasets. We compare between using the original weighted-flips-accuracy function, f, which was interpolated using data from a ResNet-50, and interpolating the function using ViT-B/16 data points. In both cases, the datasets used to interpolate are the same. Additionally, we compare to COT on this task.

Table B.4: Mean Absolute Error between estimated accuracy and true accuracy, when estimating the accuracy of a ViT-B/16 on different datasets.

Datasets	WF	WF (new f)	COT
IN-C	8.34	1.64	22.24
IN-$\overline{\text{C}}$	6.59	1.48	25.37
IN-3DCC	7.19	1.87	18.43
IN-V2	4.44	3.63	21.29
IN-D ↴			
Real	1.02	7.27	37.21
Painting	3.02	7.06	19.13
Clipart	12.82	0.25	13.58
Sketch	11.04	1.90	5.74
Infograph	9.27	3.34	1.16
Quickdraw	2.36	13.04	0.28
Average	6.61	**4.15**	16.44

B.7 Omitting Samples by Top-k Accuracy/Entropy Level

In addition to removing samples by Top-k accuracy, we also analyze the effects of removing samples according to their initial entropy level. We find that both experiments exhibit similar behaviour: it is possible to remove many Top-k/low entropy samples, without significantly affecting the accuracy gain of Tent (on a holdout set of Gaussian Noise 3).

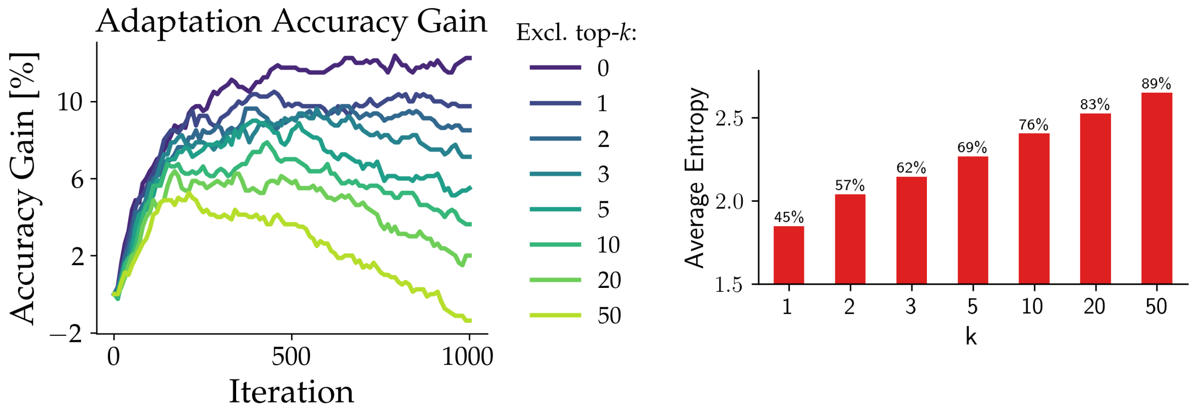


Figure B.4: **Left:** Average entropy across top-k samples for different values of k. The percentages shown are the fraction of images out of the whole dataset. The original dataset, Gaussian Noise 3, has an average entropy of 2.84. **Right:** The relative size of the datasets, when top-k samples are removed.

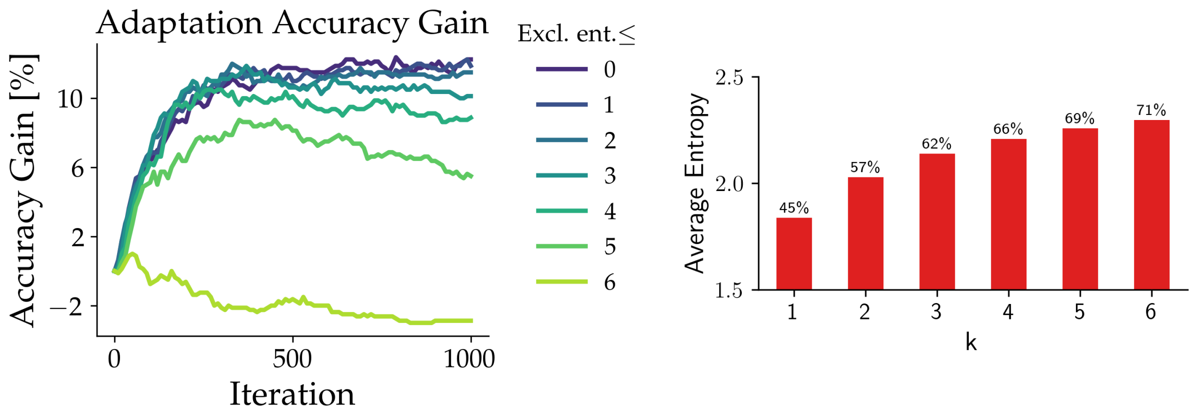


Figure B.5: **Left:** Accuracy gain per iteration on a holdout set, as Tent adapts to its inputs. Each line corresponds to a different experiment where we remove samples based on their initial entropy level. Similarly to Figure 3.2, it's possible to remove low entropy samples while barely hurting performance. When entropy ≤ 0, no images are excluded. **Right:** The relative size of the datasets and their average entropy, when samples with a entropy level $\leq k$ are removed.

B.8 Silhouette score, Shift distance, and Accuracy Throughout Entropy Minimization

In Figure 3.4, we looked at the changes of Silhouette scores/Shift distances for each phase in EM. Here, we show how these scores, along with accuracy, change in every iteration of Tent. For each one of the datasets analyzed, we group noises based on severity level, and plot their averages and standard deviations, for every iteration.

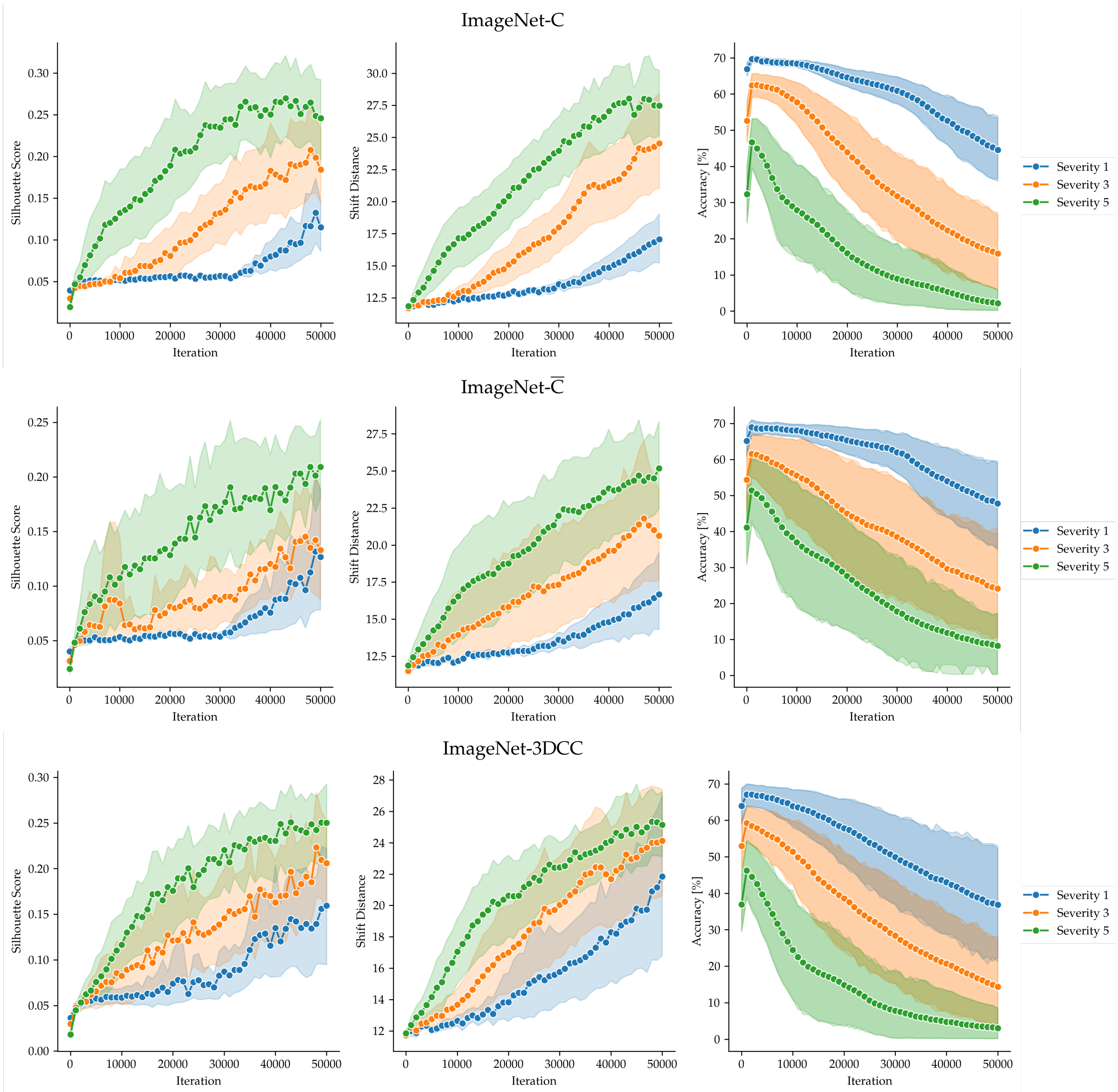


Figure B.6: Changes in Silhouette scores, Shift distances, and Accuracies as Tent adapts to its inputs. We group together noises by severity level, and average the data for every iteration.

B.9 WF on CIFAR10/100

WF is not as effecitve on CIFAR10 Krizhevsky *et al.* (2009) as it is on ImageNet Deng *et al.* (2009) scale datasets. CIFAR10 is an outlier in entropy minimization: for example, Press *et al.* (2023) showed that Tent doesn't degrade in accuracy, even after 100 million CIFAR10 images seen. We nonetheless run our method on CIFAR10. On average, we see only 0-5 label flips per dataset on C10-C. This is far from what we see ImageNet-scale datasets we tested.

Like in the paper, we interpolate a weighted-flips-to-accuracy function f on the holdout set and get:

$$f(x) = -249.36x^2 - 87.39x + 77.01$$

which has a MAE of 16.64 on the C10 validation set.

We repeat this for CIFAR100 and get:

$$f(x) = 0.000322x^2 - 0.287x + 99.54$$

which has a MAE of 9.10 on the C100 validation set.

Apart from refitting f, we did not tune any other parameter in these two experiments.

B.10 RDumb

WF uses RDumb Press *et al.* (2023) to estimate accuracy. We go over the implementation of the method in brief. RDumb is based on ETA Niu *et al.* (2022), wherein the model is reset to its pretrained state every 1,000 iterations. Rdumb optimizes the BatchNorm Ioffe and Szegedy (2015) parameters, Θ of a given classifier f.

The loss optimized is entropy, with two filtration steps: the first, in which samples with high entropy are filtered out, and the second, in which samples that produce logits similar to previous samples are filtered out.

For a sample x, the first filtration is given by:

$$S^{ent}(x) = \frac{1}{\exp[E(x;\Theta) - E_0]} \cdot \mathbb{I}_{E(x;\Theta)<E_0}(x),$$

with $E_0 = 0.4 \times \ln 10^3$.

The second filtration is given by:

$$S^{div}(x) = \mathbb{I}_{\{cos(f_o(x), m^{-1})<\varepsilon\}}(x)$$

where $cos()$ is the cosine similarity, and m^t is an exponential moving average of the logits of previously seen samples at iteration t:

$$m^t = \begin{cases} y^1, & \text{if } t = 1 \\ \alpha y^t + (1-\alpha)m^{t-1}, & \text{if } t > 1 \end{cases}$$

and y^t is the average model prediction on a batch of inputs at step t, and $\alpha = 0.9$.

Put together with entropy minimization, the optimization formula becomes:

$$\min_{\hat{\Theta}} -S^{ent}(x) \cdot S^{div}(x) \sum_{y \in C} f_{\Theta}(y|x) \log f_{\Theta}(y|x)$$

RDumb uses a SGD with a learning rate of 2.5×10^{-4}, and a batch size of 64, and is reset to its pre-trained state every 1,000 iterations.

B.11 Software Licenses

- ImageNet-C (Hendrycks and Dietterich, 2019a)Apache License 2.0
  https://github.com/hendrycks/robustness
- ImageNet-R (Hendrycks *et al.*, 2021a) MIT License
  https://github.com/hendrycks/imagenet-r
- ImageNet-3D-CC (Kar *et al.*, 2022): CC-BY-NC 4.0 License
  https://github.com/EPFL-VILAB/3DCommonCorruptions
- ImageNet-$\overline{\text{C}}$ (Mintun *et al.*, 2021): MIT License
  https://github.com/facebookresearch/augmentation-corruption
- ImageNet-V2 (Recht *et al.*, 2019): MIT License
  https://github.com/modestyachts/ImageNetV2
- Backgrounds Challenge (Xiao *et al.*, 2020):
  https://github.com/MadryLab/backgrounds_challenge
- CCC (Press *et al.*, 2023): MIT License
  https://github.com/oripress/CCC
- Stylized ImageNet Geirhos *et al.* (2019b): MIT License
  https://github.com/rgeirhos/Stylized-ImageNet
- NINCO Bitterwolf *et al.* (2023): MIT License https://github.com/j-cb/NINCO
- ImageNet-D Rusak *et al.* (2022b): Apache License 2.0
  https://github.com/bethgelab/robustness
- ObjectNet Barbu *et al.* (2019): MIT License https://objectnet.dev/
- Shift Happens Benchmark: Apache License 2.0 https://github.com/shift-happens-benchmark/icml-2022

CHAPTER C

Chapter 4 Appendix

C.1 Excerpts from Citation Datasets

To demonstrate the problematic nature of automatically sourced text excerpts, we randomly choose 10 excerpts from FullTextPeerRead, ACL-200, RefSeer, and arXiv. We tag each sample chosen with one of 4 tags, as summarised in Table 1 in the main paper. We show each sample as it appears verbatim, using the datasets that appear in the official repository[1] of Gu *et al.* (2022b).

ACL-200 Bird *et al.* (2008); Medić and Šnajder (2020)

- m which the data was extracted (original). We used a combination of automatic (e.g. BLEU–4 (OTHERCIT), METEOR (OTHERCIT)) and human metrics (using crowdsourcing) to evaluate the output (see generally, TARGETCIT . However, in the interest of space, we will restrict the discussion to a human judgment task on output preferences. We found this evaluation task to be most informative for system improvement. The ta

  Unattributable

- n Section 2 that it is more difficult to extract keyphrases correctly from longer documents. Second, recent unsupervised approaches have rivaled their supervised counterparts in performance (OTHERCIT; TARGETCIT b). For example, KP-Miner (OTHERCIT), an unsupervised system, ranked third in the SemEval-2010 shared task with an F-score of 25.2, which is comparable to the best supervised system scoring 27.5. 5 An

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one

[1] https://github.com/nianlonggu/Local-Citation-Recommendation

paper.

- rams include unigrams for all feature definitions and bigrams for selected ones. Figure 3b shows a sample of the actual extended set. We use two datasets, one prepared for the CoNLL 2000 shared task (TARGETCIT and another prepared for the BioNLP/NLPBA 2004 shared task (OTHERCIT). They represent two different tagging tasks, chunking and named entity recognition, respectively. The CoNLL 2000 chunking dataset

  Trivial

- ipts were from meetings, seminars and interviews. Some authors have also referred to this phenomenon as Ellipsis because of the elliptical form of the NSU [OTHERCIT, Fern´andez et al., 2004, OTHERCIT, TARGETCIT , OTHERCIT]. While the statistical approaches 336 have been investigated for the purpose of ellipsis detection [Fern´andez et al., 2004, OTHERCIT], it has been a common practice to use rules – syntact

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- e source language is morphologically poor, such as English, and the target language is morphologically rich, such as Russian, i.e., language pairs with a high degree of surface realization ambiguity (TARGETCIT . To address this problem we propose a general approach based on bilingual neural networks (BNN) exploiting source-side contextual information. This paper makes a number of contributions: Unlike previ

  Reasonable

- n our approach and the one described in (OTHERCIT). Such a similarity is calculated by using the WordNet::Similarity tool (OTHERCIT), and, concretely, the Wu-Palmer measure, as defined in Equation 1 (TARGETCIT . 2N3 Sim(C1, C2) ? (1) N1 + N2 + 2N3 where C1 and C2 are the synsets whose similarity we want to calculate, C3 is their least common superconcept, N1 is the number of nodes on the path from C1 to C3,

  Reasonable

- ch detected image object a visual attribute and a spatial relationship to the other objects in the image. The spatial relationships are translated into selected prepositions in the resulting captions. TARGETCIT used manually segmented and labeled images and introduced visual dependency representations (VDRs) that describe spatial relationships between the image objects. The captions are generated using templ

Reasonable

- ous open source machine translation systems. The widely used Moses system (OTHERCIT) implements the standard phrase-based translation model. Parsingbased translation models are implemented by Joshua (TARGETCIT , SAMT (OTHERCIT), and cdec (OTHERCIT). Cunei (OTHERCIT) implements statistical example-based translation. OTHERCIT and OTHERCIT respectively provide additional open-source implementations of phrase-b

  Trivial

- and test set, we had about 1000 sentences each with 10 reference translations taken from the NIST 2002 MT evaluation. All Chinese data was re-segmented with the CRF-based Stanford Chinese segmenter (TARGETCIT that is trained on the segmentation of the Chinese Treebank for consistency. The parser used in Section 3 was used to parse the training data so that null elements could be recovered from the trees.

  Trivial

- rdering between nodes), their means of creation, and the scoring method used to extract the best consensus output from the lattice (OTHERCIT). In speech processing, phoneme or word lattices (OTHERCIT; TARGETCIT are used as an interface between speech recognition and understanding. Lat1318 Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1318–1327, Uppsala, Sweden

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

RefSeer Huang *et al.* (2014); Medić and Šnajder (2020)

- . Their experiments suggested that view independence does indeed affect the performance of co-training; but that CT, when compared to other algorithms that use labeled and unlabeled data, such as EM (TARGETCIT ; OTHERCIT), may still prove e#ective even when an explicit feature split is unknown, provided that there is enough implicit redundancy in the data. In contrast to previous investigations of

Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- eeded is NP-hard. On the other hand, if the permutation π avoids the pattern 1-2-3, no shuffles are needed if k $\geq$ 5 (this is the result that every triangle free circle graph is 5-colorable, see again TARGETCIT). It becomes clear once more why circle graphs "frustrated mathematicians for some years" OTHERCIT, and still continue to do so. 5 Stacking Constraints We finally consider the generalization in which ite

  Reasonable

- a small number of details they have many things in common, especially the process of motion compensation and the DCT. Due to similar motion compensation the motion vector (MV) can be reused very well TARGETCIT . Furthermore, the equivalent usage of the DCT of block size ? ? makes a transcoder implementation within the DCT-domain possible OTHERCIT. With the standardization of H.264 the task of heterogeneous trans

  Reasonable

- tioned Transactions ? Lingxiang Xiang Michael L. Scott Department of Computer Science, University of Rochester lxiang, scott@cs.rochester.edu 1. Introduction Twenty years after the initial proposal TARGETCIT , hardware transactional memory is becoming commonplace. All commercial versions to date—and all that are likely to emerge in the near future—are best effort implementations: a transaction may abort a

  Reasonable

- local values generating a cluster are uniformly distributed in the range of [μ_{ij} - $\sigma_{ij} \times 0.01$, μ_{ij} + $\sigma_{ij} \times 0.01$]. ? Irrelevant feature f ? j $\in S_i$: We uniformly generate values in the entire range TARGETCIT . We then synthetically generate co-occurrence scores. While the co-occurrence score can be arbitrarily generated, it is non-trivial to decide the ground-truth clusters when featurebased and co-occurr

  Unattributable

- for visualizing the messagesow between objects in terms of method invocations. The scenario diagrams are generated from event traces and linked to other sources of information. Jerding and colleagues TARGETCIT , OTHERCIT focus on the interactions between program components at runtime. They observed that recurring interaction pattern can be used in the abstraction process for program understanding. The authors d

  Trivial: Though the cited excerpt cites more than one paper, that author name is given.

- Many multimedia services, such as audio-video conferencing or video playback, have associated with them performance requirements that must be met to guarantee acceptable service to the users. TARGETCIT describes the requirements that some typical applications place on networks. The Tenet Real-Time Protocol Suite [Ferrari92] is one approach to providing these real-time performance guarantees in pac

  Unattributable

- y of the controlled system is jeopardized. Several scheduling paradigms have been developed to support the analysis of a task set and determine if a schedule is feasible, e.g., rate-monotone analysis TARGETCIT . These scheduling paradigms rely on the assumption that the worst-case execution time (WCET) of hard real-time tasks be known a-priori. If the WCET of all tasks is known, it can be determined if a sc

  Reasonable

- Recommended for acceptance by L. Quan. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0308-1003. æ recovered TARGETCIT , OTHERCIT. Note that these calibration techniques can be used for both central and noncentral catadioptric cameras. 2. Self-calibration. This kind of calibration techniques uses only point correspo

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- ic controller in which a single action is associated with each node, and an observation results in a deterministic transition to a successor node (OTHERCIT; Hansen 1998; TARGETCIT a). In other cases, it is a stochastic controller in which actions are selected based on a probability distribution associated with each node, and an observation results in a probabilistic transition

Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

arXiv Gu *et al.* (2022b)

- In this study we parallelized the computation of gradients to improve the efficiency, and for large datasets further improvements can be obtained by using random minibatches to perform the inversion TARGETCIT . Such a strategy can be applied to any variational inference method (e.g. also ADVI) since variational methods solve an optimization rather than a stochastic sampling problem. In comparison, this st

  Unattributable

- e been shown to provide superior generative quality, but VAEs have a number of advantages which include outlier robustness, improved training stability and interpretable, disentangled representations TARGETCIT . Disentangled representations are generally conceived to be representations in which each element relates to an independent (and usually semantically meaningful) generative factor OTHERCIT OTHERCIT . Achieving a di

  Reasonable

- tion (NTF) OTHERCIT . For example, NMF/NTF-based ML methods have been successfully used for analysis of Monte Carlo simulated fission chamber's output signals OTHERCIT , for compression of scientific simulation data TARGETCIT , and for a variety of other applications OTHERCIT . To avoid confusion, we should emphasize that in this paper the term tensor is used to define two different types of mathematical objects. We use tensors t

  Unattributable

- insight about the generalization to the multipartite scenario, but also since the recovery problem for a tripartite probability distribution given all the three possible bipartite marginals is open OTHERCIT TARGETCIT OTHERCIT . Moreover, moving to the quantum scenario, also the compatibility problem for just a couple of overlapping marginals is open OTHERCIT OTHERCIT . We are then going to assume the set of the two given marginal densit

Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- seen that the proxy-SU(3) symmetry suggests N = 116 as the point of the prolate-to-oblate shape/phase transition, in agreement with existing exprerimental evidence OTHERCIT OTHERCIT OTHERCIT OTHERCIT OTHERCIT and microscopic calculations OTHERCIT OTHERCIT OTHERCIT TARGETCIT OTHERCIT . Table 1 . Comparison between SU(3) irreps for U(6), U(10), U(15), and U(21), obtained by the code UNTOU3 OTHERCIT , contained in the relevant U(n) irrep for M valence protons or M valence neutrons. Above

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- h cannot be explained by the traditional expected utility theory. In the context of decision-theoretic systems, Nadendla et al. have presented detection rules employed by prospect theoretic agents in TARGETCIT under different scenarios based on decision costs. In particular, the authors have focused on two types of prospect theoretic agents, namely optimists and pessimists, and have shown that the prospect

  Trivial: The name of the author of the referenced paper appears in the excerpt.

- .) (3) $\psi(\wedge$ S) does depend on the isotopy class of the collection. Its image in the space A($\star$ k 1 ,... ,kμ) , however, does not. These issues, and the above proof, are discussed in full detail in TARGETCIT . We remark that, in the form presented, this theorem does not depend on the two pieces of heavy machinery employed by OTHERCIT -it depends on neither the adapted Kirby-Fenn-Rourke theorem nor the OTHERCIT calculati

  Unattributable

- ed to follows an addition rule 2ND 2 = analogous to that found for frequency conversion. A series of recent experiments demonstrated a more complex transfer of OAM in the generation of Raman sideband TARGETCIT OTHERCIT OTHERCIT . This process was found to follow a now wellestablished OAM-algebra for Stokes and anti-Stokes orders and was definitively verified through phase measurements in a simultaneous Young double slit e

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- BMD. An important tool to assess the performance of decoding metrics is the generalized mutual information (GMI) OTHERCIT Sec. 2.4]. An interpretation of uniform BMD and bit-shaped BMD as a GMI are given in TARGETCIT and OTHERCIT , respectively. In OTHERCIT Sec. 4.2.4], the GMI is evaluated for a bit-metric. It is observed that the GMI increases when the bits are dependent. We call this approach shaped GMI. Besides the GMI, oth

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

- cay products dilute faster than matter, the expansion rate can be reduced around z $\sim$ 2.3. However, the simplest such model, a dark matter component decaying into dark radiation with constant lifetime TARGETCIT OTHERCIT , is in conflict with observations of the late integrated SachsWolfe effect and lensing power spectrum OTHERCIT OTHERCIT . Moreover, we find Ω ExDE becomes positive again at z < 1.5. Thus any decaying component mus

  Ambiguous: The citation is ambiguous by definition, as the excerpt cites more than one paper.

FullTextPeerRead Jeong *et al.* (2020)

- tion function: r=g.The typical training criterion for autoencoders is minimizing the reconstruction error, Σx∈XL with respect to some loss L, typically either squared error or the binary cross-entropy TARGETCIT .Denoising autoencoders are an extension of autoencoders trained to reconstruct a clean version of an input from its corrupted version . The denoising task requires the network to learn representatio

  Ambiguous: Although Bengio (2013) is cited, it could be argued that the original paper that used cross entropy as a loss Cox (1958) should be used.

- al matrices of parameters, and show that it outperforms the random counterpart when applied to the problem of replacing one of the fully connected layers of a convolutional neural network for ImageNet TARGETCIT . Interestingly, while the random variant is competitive in simple applications , the adaptive variant has a considerable advantage in more demanding applications .The adaptive SELLs, including Adapti

Trivial

- eneous information networks. Recently, u peek_meaning:NTF . peek_catcode:NTF a . . published a question answering algorithm that converts a given question into a vector space model to find the answer TARGETCIT , but, like neural network based models 2013 , the learned model is generally uninterpretable. peek_meaning:NTF . peek_catcode:NTF a . . proposed T-verifier, a search engine based fact checker 2011

  Ambiguous: The cited paper is Guu *et al.* (2015), while Iyyer *et al.* (2014) also fits the description given.

- he graph's main component correctly. The state-of-the-art described in gives a lowest value at 58, with the best algorithms around 60, while algorithms regularized spectral methods such as the one in TARGETCIT obtain about 80 errors.The current result should also extend directly to a slowly growing number of communities . It would be interesting to extend the current approach to smaller sized communities or

  Unattributable

- amming approach that was used in all other structural tractability results that were known before, and as we have seen this is no coincidence. Instead, B-acyclic #SAT lies outside the STV-framework of TARGETCIT that explains all old results in a uniform way.We close this paper with several open problems that we feel should be explored in the future. First, our algorithm for #SAT is specifically designed for

  Unattributable

- our method on a fully-connected network , we compare our method with on this dataset. CIFAR and SVHN dataset, we evaluate our method on three popular network architectures: VGGNet, Net and DenseNet TARGETCIT . The VGGNet is originally designed for ImageNet classification. For our experiment a variation of the original VGGNet for CIFAR dataset is taken from . For Net, a 164-layer pre-activation Net with bo

Trivial

- ars, various probabilistic extensions of description logics have been investigated, see, for instance,.The one that is closest to our approach is the type 1 extension of ALC proposed in the appendixof TARGETCIT . Briefly, This difference is the main reason why the ExpTime algorithm proposed by tz and Schrödercannot be transferred to our setting. It does not suffice to consider the satisfiable types independ

  Unattributable

- h we compute through current input and the previous hidden state. The final output of hidden state would be calculated based on memory cell and forget gate.In our experiment we used model discussed in TARGETCIT .t x is feature vector for tth word in a sentence and hl is previous hidden state then computation of hidden and output layer of LSTM would be.Where σ is sigmoid activation function, $\star$ is a element

  Unattributable

- e use of conditional LSTMs in the generation component of neural network -based dialogue systems which depend on multiple conditioning sources and optimising multiple metrics.ral conversational agents TARGETCIT are direct extensions of the sequence-to-sequence model in which a conversation is cast as a source to target transduction problem.wever, these models are still far from real world applications becau

  Ambigiuous: The cited paper is Vinyals and Le (2015), though Tang *et al.* (2019) also fits the description given.

- onsistent with previous findings.As a comparison we also include test performances of a BNN with a Gaussian approximation , a BNN with HMC, and a sparse Gaussian process model with 50 inducing points TARGETCIT . In test-LL metric our best dropout model out-performs the Gaussian approximation method on almost all datasets, and for some datasets is on par with HMC which is the current gold standard for yesian

  Ambiguous: The cited paper is Bui *et al.* (2016), while Burt *et al.* (2020) also fits the description given.

C.1.1 Automatic Ambiguity Analysis

In addition to the manual analysis above, we conducted an automated analysis of the ambiguous category. Specifically, we identified excerpts that cited multiple papers simultaneously (e.g., `\cite{paper1, paper2, paper3}`) where one of the cited papers is the target. This analysis allowed us to establish a lower bound on ambiguous excerpts across all benchmarks (Table C.1). These excerpts can not serve well as questions since they have multiple different correct answers, whereas the respective benchmarks only include one correct target answer (as in CiteME).

Table C.1: Dataset ambiguity percentages from an automatic analysis. We note that this is just a lower bound estimate, as the automatic parsing is only able to detect a subset of the ambiguous excerpts. Still, these findings are consistent with our previous results, and show that previous benchmarks contain vast quantities of ambiguous excerpts.

Dataset	**Ambiguous [%]**
arXiv	54.96
ACL	27.20
RefSeer	12.61

FulllTextPeerRead automatically deletes all other citations, so this was not possible to do in their case. We have updated Table 1 in the revised draft with the results with the expanded 50-sample sets and included the automatic evaluation data.

C.2 Additional Comparison to Existing Benchmarks

We additionally compare CiteME to previous benchmarks based on information found in Gu *et al.* (2022b). Importatnly, CiteME differs from previous work in that the query set, from which the answers come from, is by far the largest with 218 million papers. Additionally, CiteME makes the entire paper available to the model, and not just a snippet. These two factors make CiteME able to mimic the experience a research would have when looking for papers.

Table C.2: Comparison of previous benchmarks and CiteME based on query set size, availability of full paper text, and date range.

Dataset	Query Set Size	Full Paper Text	Date Range
FullTextPeerRead Jeong *et al.* (2020)	5K	✗	'07-'17
ACL-200 Bird *et al.* (2008); Medić and Šnajder (2020)	20K	✗	'09-'15
RefSeer Huang *et al.* (2014); Medić and Šnajder (2020)	625K	✗	Unk - '14
arXiv Gu *et al.* (2022b)	1.7M	✗	'91-'20
CiteME (Ours)	218M	✓	'08-'24

C.3 CiteAgent Results By Year

Language models may perform better on papers they encountered during training, with a drop in performance on newer papers, leading to better performance from more recently released models. To test this, we compare the results of our CiteAgent on excerpts from papers published before 2024 versus on excerpts from papers published in 2024. We note that the cutoff dates for Claude 3 Opus, Claude 3.5 Sonnet and GPT-4o are August 2023, August 2023 and October 2023 respectively. The results, shown in Table C.3, show that this is indeed true for the LMs analyzed in this paper.

C.4 Verifying GPT-4 Paper Tags

We asked GPT-4 to generate 3 general tags that describe every paper in CiteME. We manually verify that the tags automatically generated by GPT-4 are overwhelmingly correct. Here, we

Table C.3: Accuracy of CiteAgent models (in %) on questions where the target papers were published either before 2024 and during 2024

Model	Before 2024	2024
CiteAgent + GPT-4	36.99%	32.61%
CiteAgent + Claude 3 Opus	28.77%	21.74%
CiteAgent + Claude 3.5 Sonnet	42.47%	36.96%

give a few examples of papers, and their matching tags:

- **Paper Name:** PaLI: A Jointly-Scaled Multilingual Language-Image Model

  Tags: Multimodal AI Models, Vision-Language Integration, Scalable Machine Learning

- **Paper Name:** Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets

  Tags: Neural Network Generalization, Deep Learning Optimization, Algorithmic Data Analysis

- **Paper Name:** Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack

  Tags: Adversarial Machine Learning, Neural Network Security, Robustness Evaluation]

- **Paper Name:** Mamba-R: Vision Mamba ALSO Needs Registers

  Tags: Computer Vision Models, Image Processing Techniques, Neural Network Architectures

- **Mass-Editing Memory in a Transformer**

  Tags: Language Model Updating, Natural Language Processing, Machine Learning Techniques

C.5 Example Trajectory

Please note that the abstracts retrieved in the paper search have been redacted, and only the first two of the ten results are displayed to condense the text. Additionally, the content retrieved by the paper reading action has been abbreviated. The system prompt is omitted for brevity. Complete and exact prompts are available in the published code.

User:

You are now given an excerpt. Find me the paper cited in the excerpt, using the tools described above. Please make sure that the paper you select really corresponds to the excerpt: there will be details mentioned in the excerpt that should appear in the paper. If you read an abstract and it seems like it could be the paper we're looking for, read the paper to make sure. Also: sometimes you'll read a paper that cites the paper we're looking for. In such cases, please go to the references in order to find the full name of the paper we're looking for, and search for it, and then select it.

Our implementation for ImageNet follows the practice in [21, 41]. The image is resized with its shorter side randomly sampled in [256, 480] for scale augmentation [41]. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted [21]. The standard color augmentation in [21] is used. We adopt batch normalization (BN) [CITATION] right after each convolution and before activation.

Assistant:

```
{
    "reason": "The excerpt mentions batch normalization (BN) and places
              it right after each convolution and before activation.
              I'll search for 'batch normalization' to find the
              appropriate citation.",
    "action": {
        "name": "search_relevance",
        "query": "batch normalization"
    }
}
```

User:

- Paper ID: 995c5f5e62614fcb4d2796ad2faab969da51713e
  Title: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
  Abstract: <redacted>
  Citation Count: 39617

- Paper ID: 713bbd21bd4780fc92a69645a8bee2a005cc90dd
  Title: Comparison of normalization algorithms for cross-batch color segmentation of histopathological images
  Abstract: <redacted>
  Citation Count: 18

<Remaining 8 Results>

Assistant:


```
{
    "reason": "The paper with ID 995c5f5e62614fcb4d2796ad2faab969da51713e
              titled 'Batch Normalization: Accelerating Deep Network
              Training by Reducing Internal Covariate Shift' is explicitly
              about batch normalization and aligns perfectly with the
              mention of batch normalization in the given excerpt.",
    "action": {
        "name": "read",
        "paper_id": "995c5f5e62614fcb4d2796ad2faab969da51713e"
    }
}
```


User:

arXiv:1502.03167v3 [cs.LG] 2 Mar 2015BatchNormalization:

AcceleratingDeepNetworkTrainingb y ReducingInternalCovariateShift

Sergey Ioffe GoogleInc., sioffe@google.comChristianSzegedy GoogleInc., szegedy@google.com Abstract TrainingDeepNeuralNetworksiscomplicatedbythefact that the distributionofeach layer’sinputschangesduring training, as the parametersof the previouslayers change. This slows down the training by requiringlower learning

<Remaining Content of the Paper>

Assistant:

```
{
    "reason": "The paper Batch Normalization: Accelerating Deep Network
               Training by Reducing Internal Covariate Shift by Sergey
               Ioffe and Christian Szegedy precisely matches the
               description in the excerpt given. It details the
               application of batch normalization, placing it right after
               each convolution and before activation.",
    "action": {
        "name": "select",
        "paper_id": "995c5f5e62614fcb4d2796ad2faab969da51713e"
    }
}
```

C.6 Technical Errors

In this section, we break down the types of technical errors that can occur when running CiteAgent with a GPT-4o or Claude 3 Opus backbone.

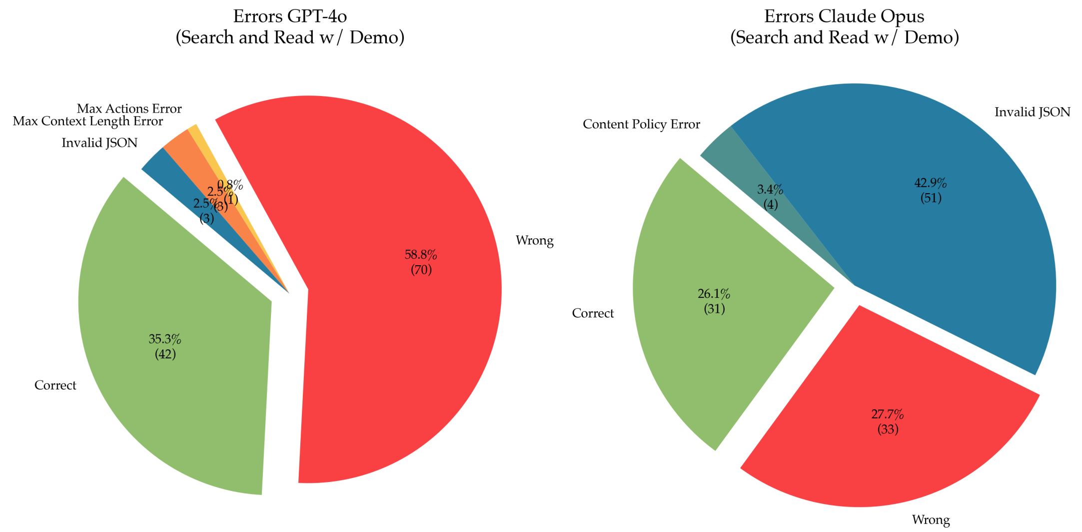


Figure C.1: Different technical errors for the CiteAgent with `Search` and `Read` command with Demo comparing the GPT-4o and Claude Opus backbone. Claude Opus has a significantly higher error rate. It struggles to adhere to the expected JSON fromat and in four cases the content filter was triggered.

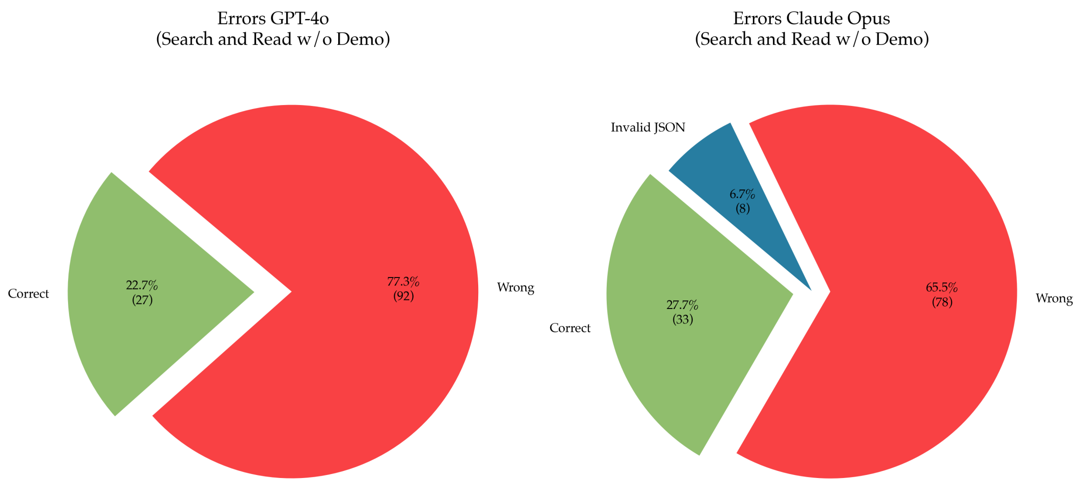


Figure C.2: Different technical errors for the CiteAgent with `Search` and `Read` command without Demo comparing the GPT-4o and Claude Opus backbone. Because there is no demo the system prompt is much shorter just containing the task description and the format instructions. One can see that the JSON error rate for Claude Opus is now drastically reduced. GPT-4o also exhibits a smaller error rate but its performance is degraded.

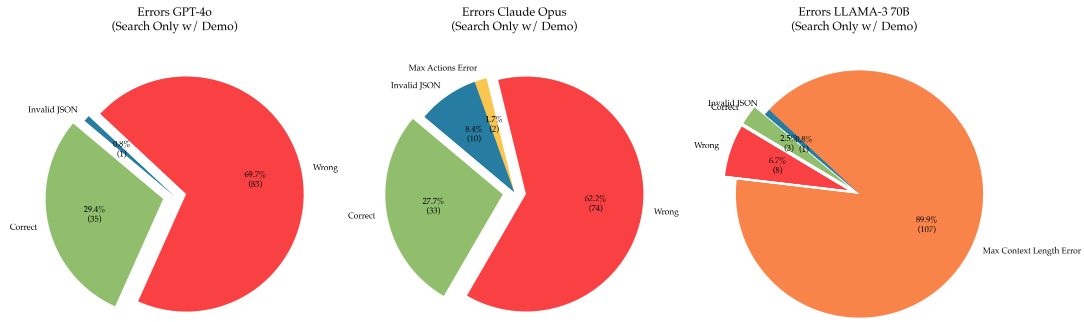


Figure C.3: Different technical errors for the CiteAgent with `Search Only` command with Demo comparing the GPT-4o, Claude Opus and LLaMA-3 70B backbone. The system prompt containing the Demo takes up a considerable amount of LLaMA-3's context length therefore just a few actions lead to the model running out of context.

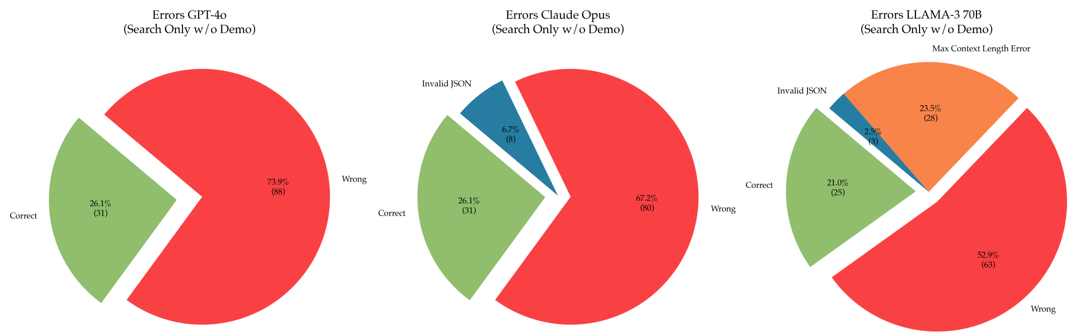


Figure C.4: Different technical errors for the CiteAgent with `Search Only` command without Demo comparing the GPT-4o, Claude Opus and LLaMA-3 70B backbone.

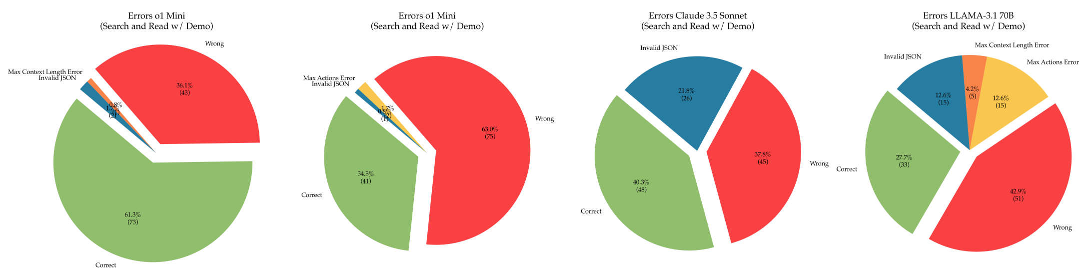


Figure C.5: Different technical errors for the CiteAgent with `Search` and `Read` command with Demo comparing the o1-Preview, o1-Mini, Claude 3.5 Sonnet and LLaMA-3.1 70B backbone.

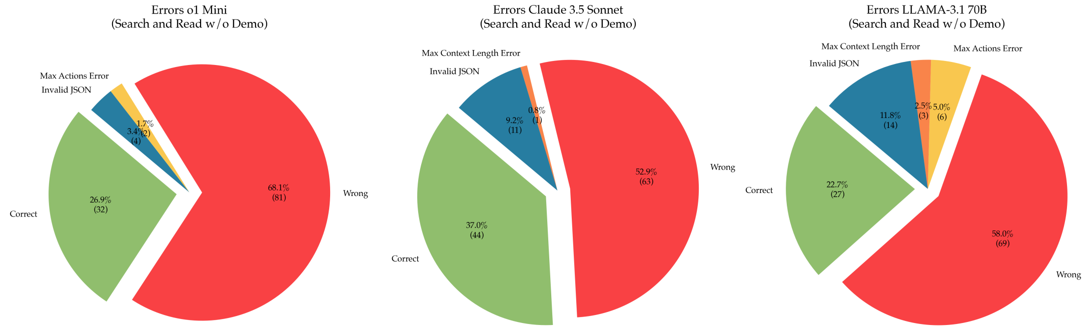


Figure C.6: Different technical errors for the CiteAgent with `Search` and `Read` command without Demo comparing the o1-Mini, Claude 3.5 Sonnet and LLaMA-3.1 70B backbone.

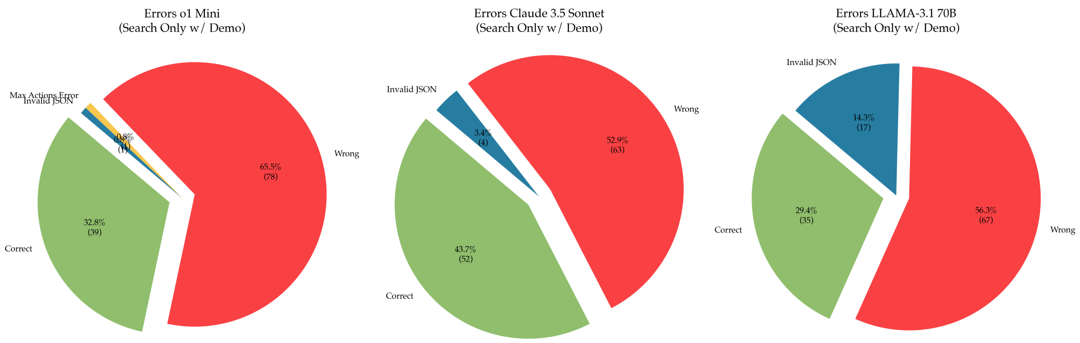


Figure C.7: Different technical errors for the CiteAgent with `Search Only` command with Demo comparing the o1-Mini, Claude 3.5 Sonnet and LLaMA-3.1 70B backbone.

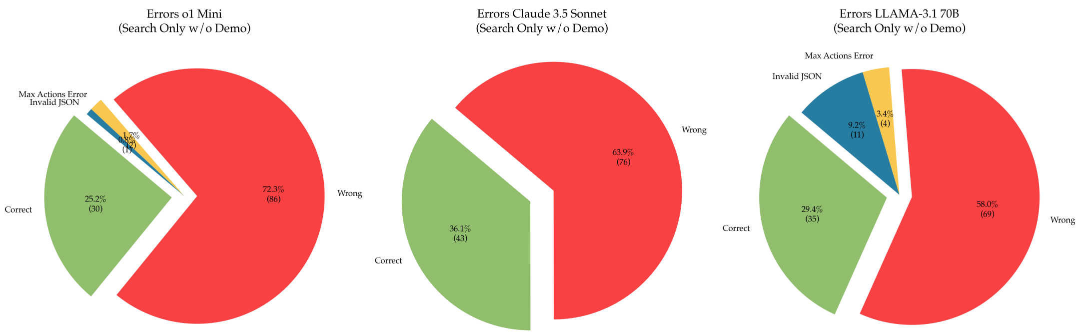


Figure C.8: Different technical errors for the CiteAgent with `Search Only` command without Demo comparing the o1-Mini, Claude 3.5 Sonnet and LLaMA-3.1 70B backbone.

C.7 Price and Duration Distribution

In this section, we break down runtimes and costs associated with running CiteAgent with a GPT-4o or Claude 3 Opus backbone.

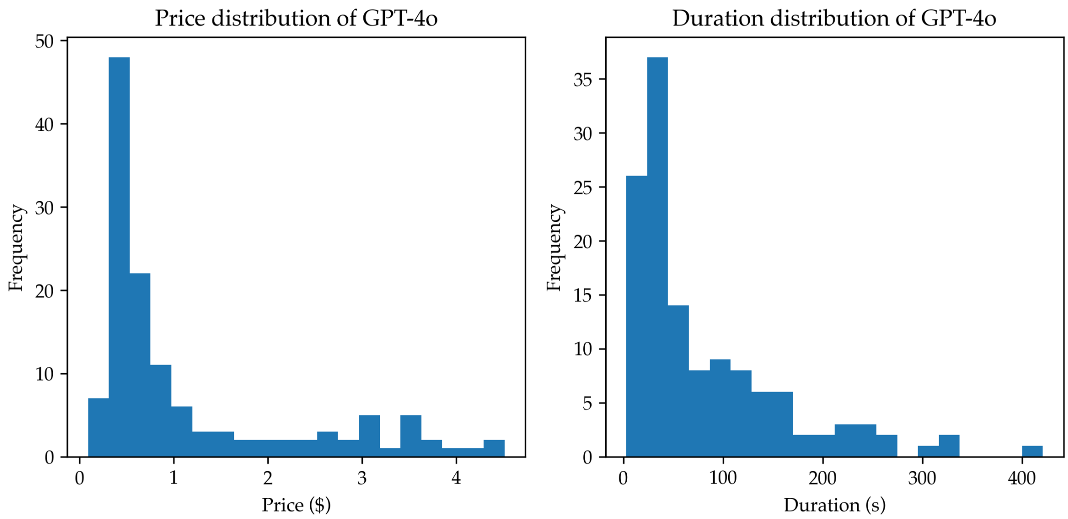


Figure C.9: Price and duration distribution on CiteME with the `Read` and `Search` command with Demo for the GPT-4o backbone. The average price is ∼\$1.2 per run or ∼\$150 in total. The average duration is 82.9 s per citation or 10772 s in total.

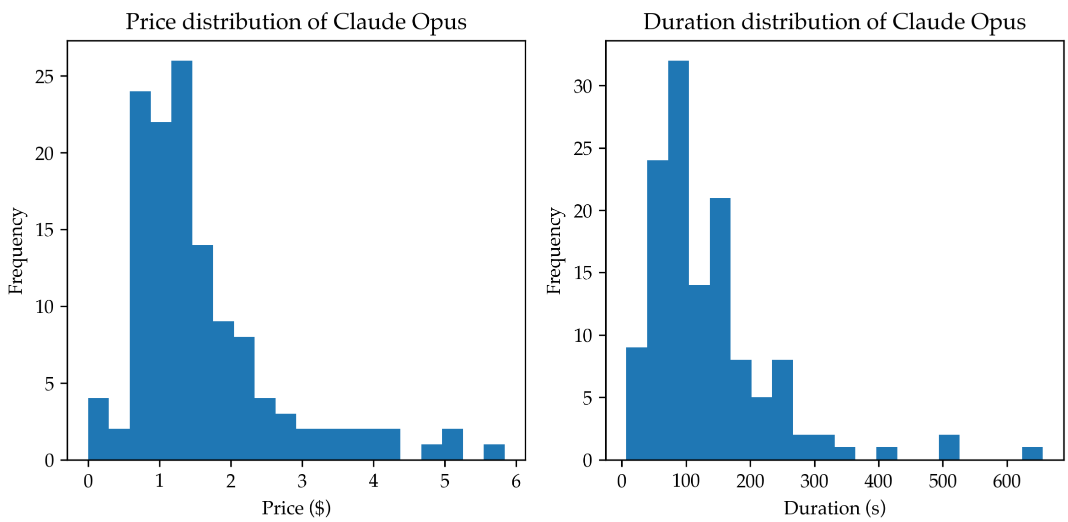


Figure C.10: Price and duration distribution on CiteME with the `Read` and `Search` command with Demo for the Claude Opus backbone. The average price is ∼\$1.6 per run or ∼\$206 in total. The average duration is 136.0 s per citation or 17675 s in total.

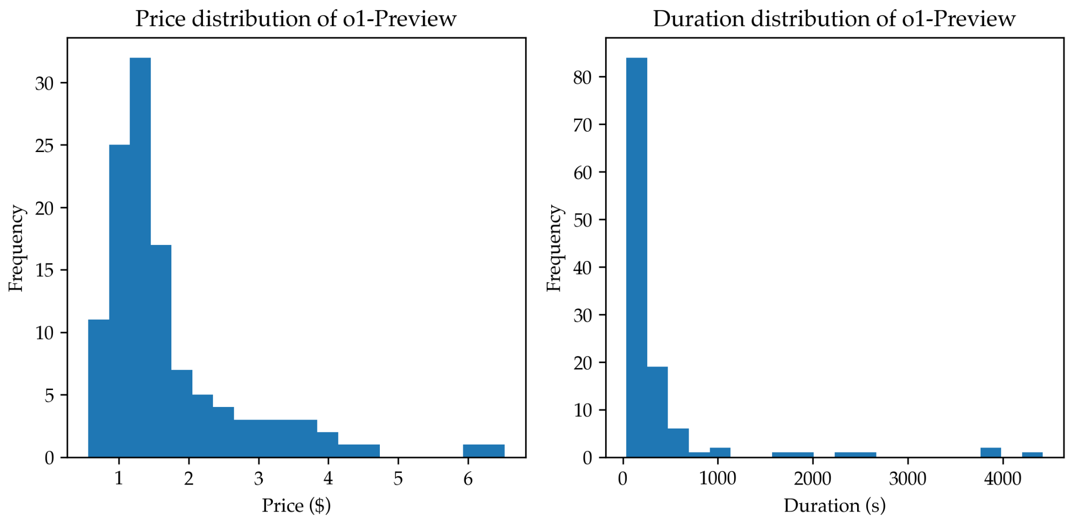


Figure C.11: Price and duration distribution on CiteME with the `Read` and `Search` command with Demo for the o1-Preview backbone. The average price is ∼\$1.7 per run or ∼\$205 in total. The average duration is 369.8 s per citation or 44006 s in total.

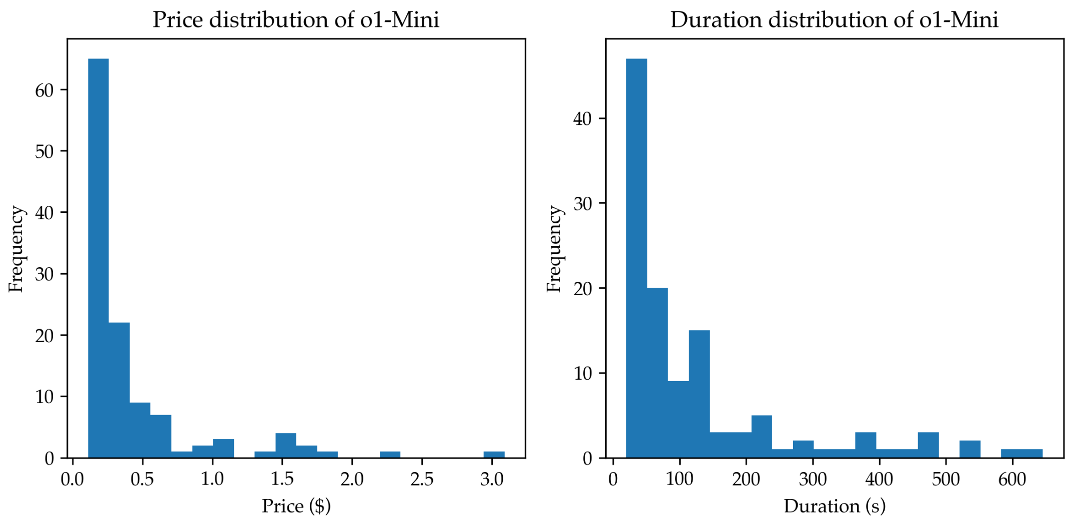


Figure C.12: Price and duration distribution on CiteME with the `Read` and `Search` command with Demo for the 01-Mini backbone. The average price is ∼\$0.4 per run or ∼\$50 in total. The average duration is 125.1 s per citation or 14886 s in total.

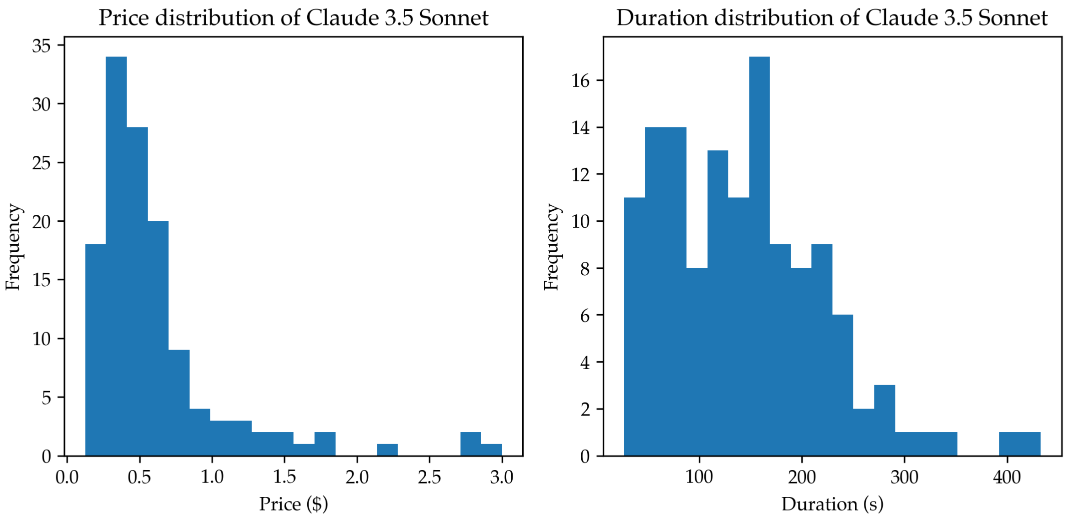


Figure C.13: Price and duration distribution on CiteME with the `Read` and `Search` command with Demo for the Claude 3.5 Sonnet backbone. The average price is ∼\$0.6 per run or ∼\$80 in total. The average duration is 143.7 s per citation or 18686 s in total.

Bibliography

Abnar, S., Berg, R. v. d., Ghiasi, G., Dehghani, M., Kalchbrenner, N., and Sedghi, H. (2021). Gradual domain adaptation in the wild: When intermediate distributions are absent. *arXiv preprint arXiv:2106.06080*.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., *et al.* (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ajith, A., Xia, M., Chevalier, A., Goyal, T., Chen, D., and Gao, T. (2024). Litsearch: A retrieval benchmark for scientific literature search. *arXiv preprint arXiv:2407.18940*.

AlBadawy, E. A., Saha, A., and Mazurowski, M. A. (2018). Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical physics*, **45**(3), 1150–1158.

Alcorn, M. A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.-S., and Nguyen, A. (2019). Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4845–4854.

Amini, M.-R. and Gallinari, P. (2002). Semi-supervised logistic regression. In *ECAI*, volume 2, page 11.

Anthropic (2024). Introducing the next generation of claude.

Anzaku, E. T., Mousavi, S. A., Van Messem, A., and De Neve, W. (2024). Re-assessing imagenet: How aligned is its single-label assumption with its multi-label nature? *arXiv preprint arXiv:2412.18409*.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

arXiv (2024). arxiv monthly submission statistics. Accessed: 2024-05-27.

Atanasova, P. (2024). Generating fact checking explanations. In *Accountable and Explainable Methods for Complex Reasoning over Text*, pages 83–103. Springer.

Augenstein, I., Baldwin, T., Cha, M., Chakraborty, T., Ciampaglia, G. L., Corney, D., DiResta, R., Ferrara, E., Hale, S., Halevy, A., *et al.* (2023). Factuality challenges in the era of large language models. *arXiv preprint arXiv:2310.05189*.

Baek, C., Jiang, Y., Raghunathan, A., and Kolter, J. Z. (2022). Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, **35**, 19274–19289.

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., and Katz, B. (2019). Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9448–9458.

Barocas, S., Hardt, M., and Narayanan, A. (2023). *Fairness and machine learning: Limitations and opportunities*. MIT press.

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473.

Ben-Shaul, I., Shwartz-Ziv, R., Galanti, T., Dekel, S., and LeCun, Y. (2023). Reverse engineering self-supervised learning. *arXiv preprint arXiv:2305.15614*.

Bengio, Y. (2013). Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pages 1–37. Springer.

Bhagavatula, C., Feldman, S., Power, R., and Ammar, W. (2018). Content-based citation recommendation. *arXiv preprint arXiv:1802.08301*.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. pages 387–402.

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, and D. Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Bitterwolf, J., Müller, M., and Hein, M. (2023). In or out? fixing imagenet out-of-distribution detection evaluation. *arXiv preprint arXiv:2306.00826*.

Bobu, A., Tzeng, E., Hoffman, J., and Darrell, T. (2018). Adapting to continuously shifting domains. *Workshop Track - ICLR 2018*.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. (2023). Autonomous chemical research with large language models. *Nature*, **624**(7992), 570–578.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., *et al.* (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G. B., Lespiau, J.-B., Damoc, B., Clark, A., *et al.* (2022). Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. *Advances in neural information processing systems*, **20**.

Boyko, J., Cohen, J., Fox, N., Veiga, M. H., Li, J. I., Liu, J., Modenesi, B., Rauch, A. H., Reid, K. N., Tribedi, S., *et al.* (2023). An interdisciplinary outlook on large language models for scientific research. *arXiv preprint arXiv:2311.04929*.

Brendel, W. and Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*.

Bui, T., Hernández-Lobato, D., Hernandez-Lobato, J., Li, Y., and Turner, R. (2016). Deep gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, pages 1472–1481. PMLR.

Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.

Burt, D. R., Rasmussen, C. E., and Van Der Wilk, M. (2020). Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, **21**(131), 1–63.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Castro, D. C., Walker, I., and Glocker, B. (2020). Causality matters in medical imaging. *Nature Communications*, **11**(1), 3673.

Chakrabarty, G., Sreenivas, M., and Biswas, S. (2023). Santa: Source anchoring network and target alignment for continual test time adaptation. *Transactions on Machine Learning Research*.

Chang, J. C., Zhang, A. X., Bragg, J., Head, A., Lo, K., Downey, D., and Weld, D. S. (2023). Citesee: Augmenting citations in scientific papers with persistent and personalized historical context. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Chen, J., Kim, G., Sriram, A., Durrett, G., and Choi, E. (2023). Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., *et al.* (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chevalier, A., Geng, J., Wettig, A., Chen, H., Mizera, S., Annala, T., Aragon, M. J., Fanlo, A. R., Frieder, S., Machado, S., *et al.* (2024). Language models as science tutors. *arXiv preprint arXiv:2402.11111*.

Cho, Y., Kim, Y., and Lee, D. (2023). Beyond entropy: Style transfer guided single image continual test-time adaptation. *arXiv preprint arXiv:2311.18270*.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., *et al.* (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, **24**(240), 1–113.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., *et al.* (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Cohan, A., Feldman, S., Beltagy, I., Downey, D., and Weld, D. S. (2020). Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*.

Cohen, K. B., Johnson, H. L., Verspoor, K., Roeder, C., and Hunter, L. E. (2010). The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC bioinformatics*, **11**, 1–10.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **20**(2), 215–232.

Cox, S., Hammerling, M., Lála, J., Laurent, J., Rodriques, S., Rubashkin, M., and White, A. (????). Wikicrow: Automating synthesis of human scientific knowledge.

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation policies from data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dai, D. and Van Gool, L. (2018). Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824. IEEE.

Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., and Jiang, Z. M. J. (2023). Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, **203**, 111734.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, **44**(7), 3366–3385.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, **39**(1), 1–22.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Deng, X., Xiao, Y., Long, B., and Zhang, Z. (2022). Reducing flipping errors in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6506–6514.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Döbler, M., Marencke, F., Marsden, R. A., and Yang, B. (2024). Diversity-aware buffer for coping with temporally correlated data streams in online test-time adaptation. *arXiv preprint arXiv:2401.00989*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.* (2010). An image is worth 16x16 words: Transformers for image recognition at scale. arxiv 2020. *arXiv preprint arXiv:2010.11929*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.* (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ebesu, T. and Fang, Y. (2017). Neural citation network for context-aware citation recommendation. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 1093–1096.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, **542**(7639), 115–118.

Färber, M. and Jatowt, A. (2020). Citation recommendation: approaches and datasets. *International Journal on Digital Libraries*, **21**(4), 375–405.

Färber, M. and Sampath, A. (2020). Hybridcite: A hybrid model for context-aware citation recommendation. In *Proceedings of the ACM/IEEE joint conference on digital libraries in 2020*, pages 117–126.

Feng, F., Chan, R. H., Shi, X., Zhang, Y., and She, Q. (2019). Challenges in task incremental learning for assistive robotics. *IEEE Access*, **8**, 3434–3441.

Fenniak, M., Stamy, M., pubpub zz, Thoma, M., Peveler, M., exiled-kingcc, and pypdf Contributors (2024). The pypdf library. See https://pypdf.readthedocs.io/en/latest/meta/CONTRIBUTORS.html for all contributors.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, **46**(4), 1–37.

Gandelsman, Y., Sun, Y., Chen, X., and Efros, A. (2022). Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, **35**, 29374–29385.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, **17**(1), 2096–2030.

Gao, T., Yen, H., Yu, J., and Chen, D. (2023). Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Garg, S., Balakrishnan, S., Lipton, Z. C., Neyshabur, B., and Sedghi, H. (2022). Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*.

Gehrke, J., Ginsparg, P., and Kleinberg, J. (2003). Overview of the 2003 kdd cup. *Acm Sigkdd Explorations Newsletter*, **5**(2), 149–151.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018a). Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, **31**.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018b). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019a). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019b). Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *ArXiv preprint*, **abs/2004.07780**.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998). Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98.

Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., and Lee, S.-J. (2022). Note: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems*.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goyal, S., Sun, M., Raghunathan, A., and Kolter, Z. (2022). Test-time adaptation via conjugate pseudo-labels. *arXiv preprint arXiv:2207.09640*.

Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, **17**.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, **37**(3), 362–386.

Gu, J., Tresp, V., and Qin, Y. (2022a). Evaluating model robustness to patch perturbations. In *ICML 2022 Shift Happens Workshop*.

Gu, N., Gao, Y., and Hahnloser, R. H. (2022b). Local citation recommendation with hierarchical-attention text encoder and scibert-based reranking. In *European conference on information retrieval*, pages 274–288. Springer.

Gu, X. and Krenn, M. (2024). Generation and human-expert evaluation of interesting research ideas using knowledge graphs and large language models. *arXiv preprint arXiv:2405.17044*.

Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., and Schmidt, L. (2021). Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1134–1144.

Guu, K., Miller, J., and Liang, P. (2015). Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*.

Han, J., Liang, X., Xu, H., Chen, K., Lanqing, H., Mao, J., Ye, C., Zhang, W., Li, Z., Liang, X., *et al.* (2021a). Soda10m: A large-scale 2d self/semi-supervised object detection dataset for autonomous driving. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Han, X., Papyan, V., and Donoho, D. L. (2021b). Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*.

Hanselowski, A., Stab, C., Schulz, C., Li, Z., and Gurevych, I. (2019). A richly annotated corpus for different tasks in automated fact-checking. *arXiv preprint arXiv:1911.01214*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

He, Q., Pei, J., Kifer, D., Mitra, P., and Giles, L. (2010). Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430.

Hendrycks, D. and Dietterich, T. (2019a). Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Hendrycks, D. and Dietterich, T. G. (2019b). Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. (2020a). Augmix: A simple data processing method to improve robustness and uncertainty. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., *et al.* (2020b). The many faces of robustness: A critical analysis of out-of-distribution generalization. *ArXiv preprint*, **abs/2006.16241**.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., *et al.* (2021a). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. (2021b). Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. (2021c). Natural adversarial examples. *CVPR*.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., *et al.* (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, **29**(6), 82–97.

Hoffman, J., Darrell, T., and Saenko, K. (2014). Continuous manifold based adaptation for evolving visual domains. In *Computer Vision and Pattern Recognition (CVPR)*.

Hoi, S. C., Sahoo, D., Lu, J., and Zhao, P. (2021). Online learning: A comprehensive survey. *Neurocomputing*, **459**, 249–289.

Huang, C., Wu, Z., Hu, Y., and Wang, W. (2024). Training language models to generate text with citations via fine-grained rewards. *arXiv preprint arXiv:2402.04315*.

Huang, W., Wu, Z., Mitra, P., and Giles, C. L. (2014). Refseer: A citation recommendation system. In *IEEE/ACM joint conference on digital libraries*, pages 371–374. IEEE.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, **32**.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org.

Iyyer, M., Boyd-Graber, J., Claudino, L., Socher, R., and Daumé III, H. (2014). A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 633–644.

Jeong, C., Jang, S., Park, E., and Choi, S. (2020). A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, **124**, 1907–1922.

Jiang, Y., Nagarajan, V., Baek, C., and Kolter, J. Z. (2021). Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. (2023). Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kang, D., Ammar, W., Dalvi, B., Van Zuylen, M., Kohlmeier, S., Hovy, E., and Schwartz, R. (2018). A dataset of peer reviews (peerread): Collection, insights and nlp applications. *arXiv preprint arXiv:1804.09635*.

Kar, O. F., Yeo, T., Atanov, A., and Zamir, A. (2022). 3d common corruptions and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18963–18974.

Khalifa, M., Wadden, D., Strubell, E., Lee, H., Wang, L., Beltagy, I., and Peng, H. (2024). Source-aware training enables knowledge attribution in language models. *arXiv preprint arXiv:2404.01019*.

Kim, E., Sun, M., Raghunathan, A., and Kolter, Z. (2023). Reliable test-time adaptation via agreement-on-the-line. *arXiv preprint arXiv:2310.04941*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kinney, R., Anastasiades, C., Authur, R., Beltagy, I., Bragg, J., Buraczynski, A., Cachola, I., Candra, S., Chandrasekhar, Y., Cohan, A., *et al.* (2023). The semantic scholar open data platform. *arXiv preprint arXiv:2301.10140*.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., *et al.* (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR.

Krizhevsky, A., Hinton, G., *et al.* (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, **2**(1-2), 83–97.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, **22**(1), 79–86.

Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., Rodriques, S. G., and White, A. D. (2023). Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, **10**(1), 1096.

Learning, M. (2011). Apolo: Making sense of large network data by combining.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, **2**.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, **521**(7553), 436–444.

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop : Challenges in Representation Learning (WREPL)*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., *et al.* (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, **33**, 9459–9474.

Liang, J., Hu, D., Wang, Y., He, R., and Feng, J. (2021). Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lin, J. (2009). Is searching full text more effective than searching abstracts? *BMC bioinformatics*, **10**, 1–15.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, **42**, 60–88.

Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T. M., and Xia, F. (2015). Context-based collaborative filtering for citation recommendation. *Ieee Access*, **3**, 1695–1703.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, **12**, 157–173.

Liu, Z., Miao, Z., Pan, X., Zhan, X., Lin, D., Yu, S. X., and Gong, B. (2020). Open compound domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12406–12415.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, **28**(2), 129–137.

Lomonaco, V. and Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 17–26. PMLR.

Lu, Y., Wang, Z., Zhai, R., Kolouri, S., Campbell, J., and Sycara, K. (2023). Predicting out-of-distribution error with confidence optimal transport. *arXiv preprint arXiv:2302.05018*.

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. (2024). Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Malaviya, C., Lee, S., Chen, S., Sieber, E., Yatskar, M., and Roth, D. (2023). Expertqa: Expert-curated questions and attributed answers. *arXiv preprint arXiv:2309.07852*.

Marsden, R. A., Döbler, M., and Yang, B. (2024). Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2555–2565.

Mayr, P. (2014). Are topic-specific search term, journal name and author name recommendations relevant for researchers? *arXiv preprint arXiv:1408.4440*.

McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A., and Riedl, J. (2002). On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 116–125.

Medić, Z. and Šnajder, J. (2020). Improved local citation recommendation based on context enhanced with global information. In *Proceedings of the first workshop on scholarly document processing*, pages 97–103.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, **54**(6), 1–35.

Metzler, D., Tay, Y., Bahri, D., and Najork, M. (2021). Rethinking search: making domain experts out of dilettantes. In *Acm sigir forum*, volume 55, pages 1–27. ACM New York, NY, USA.

Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., *et al.* (2023). Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., and Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. *ArXiv preprint*, **abs/1907.07484**.

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., and Schmidt, L. (2021). Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR.

Mintun, E., Kirillov, A., and Xie, S. (2021). On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, **34**, 3571–3583.

Miret, S. and Krishnan, N. (2024). Are llms ready for real-world materials discovery? *arXiv preprint arXiv:2402.05200*.

Motiian, S., Jones, Q., Iranmanesh, S., and Doretto, G. (2017). Few-shot adversarial domain adaptation. *Advances in neural information processing systems*, **30**.

Mummadi, C. K., Hutmacher, R., Rambach, K., Levinkov, E., Brox, T., and Metzen, J. H. (2021). Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*.

Murthy, S. K., Lo, K., King, D., Bhagavatula, C., Kuehl, B., Johnson, S., Borchardt, J., Weld, D. S., Hope, T., and Downey, D. (2022). Accord: A multi-document approach to generating diverse descriptions of scientific concepts. *arXiv preprint arXiv:2205.06982*.

Muthukadan, B. (2011). Selenium with python. `https://selenium-python.readthedocs.io/`.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. (2020). Evaluating prediction-time batch normalization for robustness under covariate shift. *ArXiv preprint*, **abs/2006.10963**.

Nallapati, R. M., Ahmed, A., Xing, E. P., and Cohen, W. W. (2008). Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. (2017). Exploring generalization in deep learning. *Advances in neural information processing systems*, **30**.

Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., and Tan, M. (2022). Efficient test-time model adaptation without forgetting. *arXiv preprint arXiv:2204.02610*.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., and Tan, M. (2023). Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*.

Ohagi, M. and Aizawa, A. (2022). Pre-trained transformer-based citation context-aware citation network embeddings. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, pages 1–5.

Oliver, A., Odena, A., Raffel, C., Cubuk, E., and Goodfellow, I. (2018). Realistic evaluation of semi-supervised learning algortihms. In *International conference on learning representations*, pages 1–15.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al.* (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, **35**, 27730–27744.

Palani, S., Naik, A., Downey, D., Zhang, A. X., Bragg, J., and Chang, J. C. (2023). Relatedly: Scaffolding literature reviews with existing related work sections. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Papyan, V., Han, X., and Donoho, D. L. (2020). Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, **117**(40), 24652–24663.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural networks*, **113**, 54–71.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., *et al.* (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, **32**.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. (2019). Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Pintor, M., Angioni, D., Sotgiu, A., Demetrio, L., Demontis, A., Biggio, B., and Roli, F. (2023). Imagenet-patch: A dataset for benchmarking machine learning robustness against adversarial patches. *Pattern Recognition*, **134**, 109064.

Poland, W. B. and Shachter, R. D. (1993). Mixtures of gaussians and minimum relative entropy techniques for modeling continuous uncertainties. In *Uncertainty in Artificial Intelligence*, pages 183–190. Elsevier.

Polo, F. M., Weber, L., Choshen, L., Sun, Y., Xu, G., and Yurochkin, M. (2024). tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*.

Prabhu, A., Torr, P. H., and Dokania, P. K. (2020). Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer.

Prabhu, A., Udandarao, V., Torr, P., Bethge, M., Bibi, A., and Albanie, S. (2024). Lifelong benchmarks: Efficient model evaluation in an era of rapid progress. *arXiv preprint arXiv:2402.19472*.

Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., and Lewis, M. (2022). Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Press, O., Schneider, S., Kümmerer, M., and Bethge, M. (2023). Rdumb: A simple approach that questions our progress in continual test-time adaptation. *Advances in Neural Information Processing Systems*, **36**.

Press, O., Hochlehnert, A., Prabhu, A., Udandarao, V., Press, O., and Bethge, M. (2024a). Citeme: Can language models accurately cite scientific claims? In *Proceedings of the Thirty-Eighth Conference on Neural Information Processing Systems (NeurIPS 2024), Datasets and Benchmarks Track*.

Press, O., Shwartz-Ziv, R., LeCun, Y., and Bethge, M. (2024b). The entropy enigma: Success and failure of entropy minimization. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 41064–41085. PMLR.

Radev, D. R., Muthukrishnan, P., Qazvinian, V., and Abu-Jbara, A. (2013). The acl anthology network corpus. *Language Resources and Evaluation*, **47**, 919–944.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.* (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, **97**(2), 285.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and

Bowman, S. R. (2023). Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, **20**, 53–65.

Rusak, E., Schott, L., Zimmermann, R., Bitterwolf, J., Bringmann, O., Bethge, M., and Brendel, W. (2020a). Increasing the robustness of dnns against image corruptions by playing the game of noise. *ArXiv preprint*, **abs/2001.06057**.

Rusak, E., Schott, L., Zimmermann, R. S., Bitterwolf, J., Bringmann, O., Bethge, M., and Brendel, W. (2020b). A simple way to make neural networks robust against diverse image corruptions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 53–69. Springer.

Rusak, E., Schneider, S., Pachitariu, G., Eck, L., Gehler, P. V., Bringmann, O., Brendel, W., and Bethge, M. (2021). If your data distribution shifts, use self-learning. *Transactions of Machine Learning Research*.

Rusak, E., Schneider, S., Pachitariu, G., Eck, L., Gehler, P. V., Bringmann, O., Brendel, W., and Bethge, M. (2022a). If your data distribution shifts, use self-learning. *Transactions on Machine Learning Research*.

Rusak, E., Schneider, S., Gehler, P. V., Bringmann, O., Brendel, W., and Bethge, M. (2022b). Imagenet-d: A new challenging robustness dataset inspired by domain adaptation. In *ICML 2022 Shift Happens Workshop*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.* (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., *et al.* (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, **35**, 36479–36494.

Salvador, T. and Oberman, A. M. (2022). Imagenet-cartoon and imagenet-drawing: two domain shift datasets for imagenet. In *ICML 2022 Shift Happens Workshop*.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, **61**, 85–117.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. (2020). Improving robustness against common corruptions by covariate shift adaptation. In *Advances in neural information processing systems*.

Schuster, T., Fisch, A., and Barzilay, R. (2021). Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2020). Green ai. *Communications of the ACM*, **63**(12), 54–63.

Shi, X., Li, D., Zhao, P., Tian, Q., Tian, Y., Long, Q., Zhu, C., Song, J., Qiao, F., Song, L., Guo, Y., Wang, Z., Zhang, Y., Qin, B., Yang, W., Wang, F., Chan, R. H. M., and She, Q. (2020). Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM. In *2020 International Conference on Robotics and Automation (ICRA)*, pages 3139–3145.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, **90**(2), 227–244.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, **6**(1), 1–48.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., *et al.* (2016). Mastering the game of go with deep neural networks and tree search. *nature*, **529**(7587), 484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., *et al.* (2017). Mastering the game of go without human knowledge. *nature*, **550**(7676), 354–359.

Singh, A., D'Arcy, M., Cohan, A., Downey, D., and Feldman, S. (2022). Scirepeval: A multi-format benchmark for scientific document representations. In *Conference on Empirical Methods in Natural Language Processing*.

Skarlinski, M. D., Cox, S., Laurent, J. M., Braza, J. D., Hinks, M., Hammerling, M. J., Ponnapati, M., Rodriques, S. G., and White, A. D. (2024). Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, **33**, 596–608.

Song, J., Lee, J., Kweon, I. S., and Choi, S. (2023). Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929.

Stimberg, F., Chakrabarti, A., Lu, C.-T., Hazimeh, H., Stretcu, O., Qiao, W., Liu, Y., Kaya, M., Rashtchian, C., Fuxman, A., *et al.* (2023). Benchmarking robustness to adversarial image obfuscations. *arXiv preprint arXiv:2301.12993*.

Stock, P. and Cisse, M. (2018). Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European conference on computer vision (ECCV)*, pages 498–512.

Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, **8**(5).

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., and Anguelov, D. (2020a).

Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A., and Hardt, M. (2019a). Test-time training for out-of-distribution generalization. *ArXiv preprint*, **abs/1909.13231**.

Sun, Y., Tzeng, E., Darrell, T., and Efros, A. A. (2019b). Unsupervised domain adaptation through self-supervision. *ArXiv preprint*, **abs/1909.11825**.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. (2020b). Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Taesiri, M. R., Nguyen, G., Habchi, S., Bezemer, C.-P., and Nguyen, A. (2023). Zoom is what you need: An empirical study of the power of zoom and spatial biases in image classification. *arXiv preprint arXiv:2304.05538*.

Tang, J., Zhao, T., Xiong, C., Liang, X., Xing, E. P., and Hu, Z. (2019). Target-guided open-domain conversation. *arXiv preprint arXiv:1905.11553*.

Tang, M., Yao, S., Yang, J., and Narasimhan, K. (2023). Referral augmentation for zero-shot information retrieval. *arXiv preprint arXiv:2305.15098*.

Teney, D., Lin, Y., Oh, S. J., and Abbasnejad, E. (2022). Id and ood performance are sometimes inversely correlated on real-world datasets. *arXiv preprint arXiv:2209.00613*.

Tian, M., Gao, L., Zhang, S. D., Chen, X., Fan, C., Guo, X., Haas, R., Ji, P., Krongchon, K., Li, Y., *et al.* (2024). Scicode: A research coding benchmark curated by scientists. *arXiv preprint arXiv:2407.13168*.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. (2018). An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.

Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., *et al.* (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., and Li, Y. (2022). Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 459–479. Springer.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176.

Van de Ven, G. M. and Tolias, A. S. (2019). Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

Vapnik, V. (1991). Principles of risk minimization for learning theory. *Advances in neural information processing systems*, **4**.

Vasudevan, V., Caine, B., Gontijo Lopes, R., Fridovich-Keil, S., and Roelofs, R. (2022). When does dough become a bagel? analyzing the remaining mistakes on imagenet. *Advances in Neural Information Processing Systems*, **35**, 6720–6734.

Vinyals, O. and Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Volk, G., Müller, S., Von Bernuth, A., Hospach, D., and Bringmann, O. (2019). Towards robust cnn-based object detection through augmentation with synthetic rain variations. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pages 285–292. IEEE.

Wadden, D., Lo, K., Wang, L. L., Cohan, A., Beltagy, I., and Hajishirzi, H. (2021). Multivers: Improving scientific claim verification with weak supervision and full-document context. *arXiv preprint arXiv:2112.01640*.

Wadden, D., Lo, K., Kuehl, B., Cohan, A., Beltagy, I., Wang, L. L., and Hajishirzi, H. (2022). Scifact-open: Towards open-domain scientific claim verification. *arXiv preprint arXiv:2210.13777*.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2020a). Fully test-time adaptation by entropy minimization. *ArXiv preprint*, **abs/2006.10726**.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2020b). Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. (2019). Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. (2022). Continual test-time domain adaptation. *arXiv preprint arXiv:2203.13591*.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., and Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, **30**(4), 964–994.

Wiens, J., Saria, S., Sendak, M., Ghassemi, M., Liu, V. X., Doshi-Velez, F., Jung, K., Heller, K., Kale, D., Mohammed, A., *et al.* (2019). Do no harm: a roadmap for responsible machine learning for health care. *Nature medicine*, **25**(9), 1337–1340.

Wright, D., Wadden, D., Lo, K., Kuehl, B., Cohan, A., Augenstein, I., and Wang, L. L. (2022). Generating scientific claims for zero-shot scientific fact checking. *arXiv preprint arXiv:2203.12990*.

Wu, C., Ge, Y., Guo, Q., Wang, J., Liang, Z., Lu, Z., Shan, Y., and Luo, P. (2024a). Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots.

Wu, J. F., Hyk, A., McCormick, K., Ye, C., Astarita, S., Baral, E., Ciuca, J., Cranney, J., Field, A., Iyer, K. G., Koehn, P., Kotler, J., Kruk, S. J., Ntampaka, M., O'Neill, C., Peek, J., Sharma, S., and Yunus, M. (2024b). Designing an evaluation framework for large language models in astronomy research.

Wulfmeier, M., Bewley, A., and Posner, I. (2018). Incremental adversarial domain adaptation for continually changing environments. In *2018 IEEE International conference on robotics and automation (ICRA)*, pages 4489–4495. IEEE.

Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. (2020). Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*.

Xie, Q., Luong, M., Hovy, E. H., and Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. IEEE.

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5987–5995. IEEE Computer Society.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. (2024a). Swe-agent: Agent computer interfaces enable software engineering language models.

Yang, J., Jimenez, C. E., Zhang, A. L., Lieret, K., Yang, J., Wu, X., Press, O., Muennighoff, N., Synnaeve, G., Narasimhan, K. R., *et al.* (2024b). Swe-bench multimodal: Do ai systems generalize to visual software domains? *arXiv preprint arXiv:2410.03859*.

Yang, L., Zheng, Y., Cai, X., Dai, H., Mu, D., Guo, L., and Dai, T. (2018). A lstm based model for personalized context-aware citation recommendation. *IEEE access*, **6**, 59618–59627.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2022). React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Ye, X., Sun, R., Arik, S. Ö., and Pfister, T. (2023). Effective large language model adaptation for improved grounding. *arXiv preprint arXiv:2311.09533*.

Yoran, O., Amouyal, S. J., Malaviya, C., Bogin, B., Press, O., and Berant, J. (2024). Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuan, L., Xie, B., and Li, S. (2023). Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932.

Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., and Vincentelli, A. S. (2021). Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13834–13844.

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, **15**(11), e1002683.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhang, M., Levine, S., and Finn, C. (2022). Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, **35**, 38629–38642.

Zhang, M., Press, O., Merrill, W., Liu, A., and Smith, N. A. (2023). How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.