Learning to Generalize Across Distribution Shifts

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Frederik Joshua Träuble

aus Schwäbisch Gmünd

Tübingen

2024

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

Tag der mündlichen Qualifikation:	13.03.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Bernhard Schölkopf
2. Berichterstatter:	Prof. Dr. Georg Martius
3. Berichterstatterin:	Prof. Dr. Dhanya Sridhar

Summary

Biological intelligence demonstrates impressive abilities to learn and adapt to new situations. In contrast, the success of machine learning (ML) often relies on the assumption that data is independent and identically distributed *(i.i.d.)* during training and testing. When faced with data that deviates from this *i.i.d.* assumption, ML models are challenged with the experience of what is called *Out-of-Distribution (OOD) shifts*. These shifts represent changes in the data's underlying patterns, to which ML models are notably less adaptable compared to human capabilities. However, real-world data and applications are fundamentally characterized by a variety of such OOD shifts.

In this dissertation, we focus on improving the generalization of ML models, especially those based on deep neural networks, across a wide range of realistic out-of-distribution scenarios. A central approach to achieving this is learning abstract representations that robustly infer the underlying generative factors and associated causal mechanisms of observations. Learning such factorizing representations – ideally without knowing the factors a priori – is the primary goal of *disentanglement learning* to then obtain more adaptable ML models in a further step. Another prerequisite for better generalization is that ML models are adaptable and resilient within dynamic settings, a challenge that mainly concerns *reinforcement learning (RL)* and *continual learning*. This work will extensively deal with how such methods behave under distribution shifts and present new approaches to address various weaknesses. We primarily deal with generalization problems in computer vision and robotics using empirical, experimental, as well as theoretical methods.

In the first part of this dissertation, we deal with distribution shifts in settings whose underlying factors are known. We show the limitations of current methods in unsupervised learning, especially in *disentanglement learning* when the underlying factors contain various correlations during training. Although correlations represent a central characteristic of real-world data, they have been largely neglected when developing and testing disentanglement methods. Subsequently, we introduce a new benchmark environment for learning causal structures in robotics. This robotics environment, *CausalWorld*, allows for systematically assessing the adaptability of RL agents under various degrees of changes in their environment. *CausalWorld* enables doing interventions on any underlying variables. Then, we investigate the effectiveness of representation learning for OOD generalization within this complex and challenging robotics and RL setting. In addition to systematically investigating distribution shifts within the simulated environment, we also study the transfer of representations from simulation to reality. The presented results of these works are underpinned by large-scale empirical studies analyzing the influence of several representation metrics on different forms of OOD generalization.

In the second part of this dissertation, we will focus on learning scenarios that deviate from standard distribution shifts. We address challenges inherent in scenarios of continual learning, mainly focusing on adaptive learning and mitigating catastrophic forgetting by introducing a new information bottleneck mechanism – so-called *Discrete Key-Value Bottlenecks*. The proposed bottleneck mechanism allows for sparse, localized, and context-dependent model updates, thereby enabling favorable adaptability and generalization capabilities under strong distribution shifts at training time. Finally, we explore a particular challenge in label shifts of classification models. Here, we introduce the *Prediction Update Problem*, a problem that arises when ML predictions are to be updated by supposedly better models. To reduce the effect of negative flips, i.e., incorrect predictions

that were previously correct, we present a probabilistic Bayesian approach to efficiently address backward compatibility and adaptation under inevitable shifts in the distribution of predictions by updating ML models.

Zusammenfassung

Biologische Intelligenz zeigt beeindruckende Fähigkeiten zum Lernen und zur Anpassung an neue Situationen. Im Gegensatz dazu beruht der Erfolg des maschinellen Lernens (ML) oft auf der Annahme, dass die Daten während des Trainings und der Tests unabhängig und identisch verteilt sind (*i.i.d.*). Bei Konfrontation mit Daten, die nicht der *i.i.d.* Annahme entsprechen, stehen ML-Modelle vor der Herausforderung, sogenannten *Out-of-Distribution Verschiebungen (OOD Shifts)* zu bewältigen. Diese Verschiebungen stellen Veränderungen in den zugrunde liegenden Datenmustern dar, an die sich ML-Modelle im Vergleich zu menschlichen Fähigkeiten deutlich weniger anpassen können. Daten und Anwendungen in der realen Welt sind jedoch grundsätzlich durch eine Vielzahl solcher OOD-Shifts gekennzeichnet.

In dieser Dissertation konzentrieren wir uns darauf die Generalisierung von ML-Modellen, insbesondere basierend auf tiefen neuronalen Netzwerken, entlang vielfältiger realistischer Out-of-Distribution Szenarien besser zu verstehen. Ein zentraler Ansatz hierfür ist das Erlernen von abstrakten Repräsentationen, welche die Faktoren und damit zusammenhängende kausale Mechanismen robust inferieren sollen, die den Beobachtungen zu Grunde liegen. Das Erlernen derartiger faktorisierender Repräsentationen – idealerweise ohne die Faktoren im Voraus zu kennen – ist das primäre Ziel von *disentanglement learning* um damit in einem weiteren Schritt anpassungsfähigere Systeme zu erhalten. Eine weitere Voraussetzung für bessere Generalisierung ist es, dass ML-Modelle anpassungs- und widerstandsfähig innerhalb dynamischer Settings sind, eine Herausforderung welche vor allem das bestärkende Lernen (*reinforcement learning - RL*) und kontinuierliche Lernen betrifft. Innerhalb dieser Arbeit werden wir uns umfassend damit auseinandersetzen, wie sich derartige Methoden unter Verteilungsverschiebungen verhalten und präsentieren neue Ansätze, diverse Schwachstellen anzugehen. Wir behandeln vorrangig Generalisierungs-Probleme in der Computer Vision und Robotik unter Zuhilfenahme empirischer und experimenteller sowie theoretischer Methoden.

Im ersten Teil dieser Dissertation befassen wir uns mit Verteilungsänderungen in Szenarien, deren zu Grunde liegende Faktoren bekannt sind. Wir zeigen die Grenzen aktueller Methoden im Bereich des unüberwachten Lernens auf, insbesondere im Bereich des *disentanglement learnings*, wenn diese Faktoren verschiedene Korrelationen während des Trainings enthalten. Obwohl Korrelationen eine zentrale Charakteristik realer Daten darstellen, wurden sie in der Entwicklung und Evaluierung neuer Disentanglement-Methoden größtenteils vernachlässigt. Anschließend stellen wir eine neue Benchmark-Umgebung zum Erlernen von kausalen Strukturen in der Robotik vor. Diese Robotik-Umgebung, *CausalWorld*, ermöglicht es die Anpassungsfähigkeit von RL-Agenten unter verschiedenen Änderungsgraden ihrer Umgebung systematisch zu bewerten. *CausalWorld* erlaubt es auf beliebige zu Grunde liegende Faktoren zu intervenieren. Daraufhin werden wir die Effektivität des Repräsentationslernens für OOD-Generalisierung innerhalb dieses komplexen und herausfordernden Robotik- und RL-Settings untersuchen. Neben systematischen Untersuchung von Verteilungsverschiebungen innerhalb der simulierten Umgebung widmen wir uns darüber hinaus auch dem Transfer von Repräsentationen von der Simulation auf die Realität. Die vorgestellten Ergebnisse dieser Arbeiten untermauern wir mit Hilfe von groß angelegten empirischen Studien, die den Einfluss mehrerer Repräsentationsmetriken auf verschiedene Formen der OOD-Generalisierung analysieren.

Im zweiten Teil dieser Dissertation werden wir uns auf Lernszenarien konzentrieren, die von standardmäßigen Verteilungsverschiebungen abweichen. Wir gehen Herausforderungen an, die in Szenarien des kontinuierlichen Lernens inhärent sind, insbesondere konzentrieren wir uns auf adaptives Lernen und auf die Minderung des katastrophalen Vergessens durch die Einführung von einem neuen Informations-Engpass-Mechanismus – sogenannte *Discrete Key-Value Bottlenecks*. Der vorgeschlagene Engpass-Mechanismus ermöglicht spärliche, lokalisierte und kontextabhängige Aktualisierungen des Modells, wodurch günstige Anpassungs- und Generalisierungsfähigkeiten bei starken Verteilungsverschiebungen während des Trainings ermöglicht werden. Schließlich erforschen wir eine besondere Herausforderung bei Verschiebungen in den Vorhersagen von Klassifikationsmodellen. Diesbezüglich führen wir das *Prediction Update Problem* ein, ein Problem, das entsteht, wenn ML-Vorhersagen durch vermeintlich bessere Modelle aktualisiert werden sollen. Um den Effekt negativer Flips, d.h. falscher Vorhersagen, die zuvor korrekt waren, zu reduzieren, präsentieren wir einen probabilistischen bayes'schen Ansatz, um die Rückwärtskompatibilität und Anpassung unter unvermeidlichen Verschiebungen in der Verteilung der Vorhersagen durch die Aktualisierung von ML-Modellen effizient anzugehen.

Acknowledgements

This dissertation would not have been possible without the many individuals that have supported me throughout my life. The past four years of my PhD have been transformative, shaping me both academically and personally in ways I could have never anticipated.

First and foremost, I would like to thank Bernhard for giving me the chance to do my PhD in his lab working alongside incredibly talented and motivated colleagues as well as the unwavering support, guidance and trust he was putting into me from the first day. I am thankful for the unconditional and extensive freedom you have given me in pursuing and exploring so many different research directions. I consider myself incredibly fortunate to have had you support me through so many steps over the course of my PhD.

Second, I am especially grateful to my second examiner Georg Martius for all the work I have caused you as well as Matthias Bethge, both of whom have been on my Thesis Advisory Committee. Thank you for your insightful comments and constructive criticisms along this way that have greatly improved and influenced my research. Thank you for dedicating your time towards science and in particularly mentoring and guiding the next generation of scientists.

A special thank you goes to Stefan Bauer, having played a unique and pivotal role in supervising and mentoring me throughout the first years of my PhD. I can't list on one page how much I have learned from you and will be always thankful for all your professional, academic and personal support.

These thanks extend to Francesco Locatello and Anirudh Goyal. I am thankful for all your in-depth mentoring and countless research discussions we had. I learned so many research skills from every one of you which I am benefit from until to date.

I am also particularly grateful to Yoshua Bengio for your invaluable feedback on many research projects and the many virtual research discussions we had. Thank you for giving me the unique opportunity to become a tiny part of your amazing research group as visiting PhD researcher at Mila in 2021 and 2022.

My research would not have been possible without the collaborations and intellectual engagements I have been privileged to experience. The two most important persons standing out here were Andrea Dittadi and Nasim Rahaman. I consider myself incredibly fortunate for having met you, learning so much from both of you during all the projects we worked on together and the many times we were motivating each other. I hope this friendship will last forever! I want to extend these thanks to the many collaborators I had the privilige to work with during my PhD including and in alphabetical order: Vaibhav Agrawal, Ossama Ahmed, Olivier Bachem, Yukun Chen, Elliot Creager, Cian Eastwood, Niki Kilbertus, Kenji Kawaguchi, Armin Kekić, Julius von Kügelgen, Hugo Larochelle, Li Erran Li, Davide Mambelli, Mike Mozer, Alex Neitz, Andrei Nicolicioiu, Lukas Schott, Martin Weiss, Felix Widmaier, Ole Winther and Manuel Wüthrich.

I want to thank the entire Empirical Inference department. The warm and open environment created by my colleagues has been nothing short of exceptional. The friendships, intellectual discussions, and the supportive atmosphere have not only contributed to my research but have also played a pivotal role in my personal growth and many new friendships. Jonas W, Heiner, Luigi, Simon,

Yucen, Max D, Alex, Giovanni, Yassine, Timmy, Jonas K, Max M, Felix, Sabrina, Ann-Sophie and Volker.

I want to thank my managers and mentors during two wonderful internships I was fortunate to experience despite the pandemic. Peter Gehler and Patrick Schwab, you have both been fantastic and excellent managers and role models to me at AWS AI and GSK AI. Thanks to Matthäus Kleindessner, Ron Schwessinger, Andreas Georgiou, Lachlan Stuart and Arash Mehrjou for having been exceptional colleagues it was always fun to work with! I also wish to thank AWS for having helped us with compute resources to undertake several experiments in this thesis.

I am grateful for having a supportive circle of friends who have been there for me during the highs and lows of this journey most importantly: Philipp, Freddy, Sophia and Jannik.

Last but by no means least, I owe my deepest gratitude to my family. Their care and support in the background have been my constant source of strength. My mom and dad, my brothers Korbinian, Leo, Jakob, as well as Andrea and Roland for supporting me on every single point and decision in my life.

Saving the most heartfelt acknowledgment for last, I owe an immeasurable debt of gratitude to Sarah. Without your unwavering support and love none of these past four years would have been possible. I cannot envision any of this — my research, my growth, my happiness — without you by my side. Thank you for being with me every step of the way.

Preface

This dissertation primarily consists of the following publications:

1. **Chapter 2**
   On Disentangled Representations Learned from Correlated Data
   Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf and Stefan Bauer.
   ICML 2021: International Conference on Machine Learning (*long oral*).
   Träuble et al. [1]

2. **Chapter 3**
   CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning
   Ossama Ahmed*, **Frederik Träuble***, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf and Stefan Bauer.
   ICLR 2021 International Conference for Learning Representations.
   Ahmed et al. [2]

3. **Chapter 4**
   On the Transfer of Disentangled Representations in Realistic Settings
   Andrea Dittadi*, **Frederik Träuble***, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer and Bernhard Schölkopf
   ICLR 2021: International Conference on Learning Representations.
   Dittadi et al. [3]

4. **Chapter 5**
   The Role of Pretrained Representations for the OOD Generalization of RL Agents
   Frederik Träuble*, Andrea Dittadi*, Manuel Wüthrich, Felix Widmaier, Peter Vincent Gehler, Ole Winther, Francesco Locatello, Olivier Bachem, Bernhard Schölkopf and Stefan Bauer.
   ICLR 2022: International Conference on Learning Representations.
   Träuble et al. [4]

5. **Chapter 6**
   Discrete Key-Value Bottleneck
   Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael Curtis Mozer, Kenji Kawaguchi, Yoshua Bengio and Bernhard Schölkopf.
   ICML 2023: International Conference on Machine Learning.
   Träuble et al. [5]

6. **Chapter 7**
   Backward-Compatible Prediction Updates: A Probabilistic Approach
   Frederik Träuble, Julius Von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf and Peter Gehler.
   NeurIPS 2021: Advances in Neural Information Processing Systems.
   Träuble et al. [6]

* Shared First Author. Equal Contribution.

During my Ph.D. I additionally contributed to the following works which are not part of the main part of this thesis. A subset of them is being further contextualized with regards to this dissertation in Appendix A.

7. *Boxhead: A Dataset for Learning Hierarchical Representations*
   Yukun Chen, Andrea Dittadi, **Frederik Träuble**, Stefan Bauer and Bernhard Schölkopf.
   NeurIPS 2021 Workshop: Workshop on Shared Visual Representations in Human and Machine Intelligence (SVRHM)
   Chen et al. [7]

8. *Visual Representation Learning Does Not Generalize Strongly Within the Same Domain*
   Lukas Schott, Julius von Kügelgen, **Frederik Träuble**, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello and Wieland Brendel.
   ICLR 2022: International Conference on Learning Representations
   Schott et al. [8]

9. *Compositional Multi-Object Reinforcement Learning with Linear Relation Networks*
   Davide Mambelli*, **Frederik Träuble***, Stefan Bauer, Bernhard Schölkopf and Francesco Locatello.
   ICLR 2022 Workshop: Workshop on the Elements of Reasoning: Objects, Structure and Causality
   Mambelli et al. [9]

10. *A General Purpose Neural Architecture for Geospatial Systems*
    Nasim Rahaman*, Martin Weiss*, **Frederik Träuble**, Francesco Locatello, Alexandre Lacoste, Yoshua Bengio, Chris Pal, Li Erran Li and Bernhard Schölkopf.
    NeurIPS 2022 Workshop: Workshop on AI for Humanitarian Assistance and Disaster Response (*long oral*)
    Rahaman et al. [10]

11. *DCI-ES: An Extended Disentanglement Framework with Connections to Identifiability*
    Cian Eastwood*, Andrei Liviu Nicolicioiu*, Julius Von Kügelgen*, Armin Kekić, **Frederik Träuble**, Andrea Dittadi and Bernhard Schölkopf.
    ICLR 2023: International Conference on Learning Representations
    Eastwood et al. [11]

12. *Unlearning via Sparse Representations*
    Vedant Shah, **Frederik Träuble**, Ashish Malik, Hugo Larochelle, Michael Mozer, Sanjeev Arora, Yoshua Bengio and Anirudh Goyal.
    Preprint. Shah et al. [12]

I additionally contributed to the following work which is under journal review.

13. *Multi-Megabase Scale Genome Interpretation with Genetic Language Models*
    Frederik Träuble*, Lachlan Stuart*, Andreas Georgiou*, Pascal Notin, Arash Mehrjou, Ron Schwessinger, Mathieu Chevalley, Kim Branson, Bernhard Schölkopf, Cornelia van Duijn, Debora Marks and Patrick Schwab.
    Preprint. Träuble et al. [13]

Contents

Summary iii

Zusammenfassung v

Contents xi

1 Introduction 1
- 1.1 Distribution Shifts . . . 2
- 1.2 Generalization and Adaptation Strategies . . . 5
- 1.3 Outline . . . 8

2 On Disentangled Representations Learned from Correlated Data 13
- 2.1 Introduction . . . 15
- 2.2 Background . . . 17
- 2.3 The Problem with Correlated Data . . . 18
- 2.4 Unsupervised Disentanglement under Correlated Factors . . . 21
- 2.5 Recovering Factorized Latent Codes . . . 26
- 2.6 Other Related Work . . . 29
- 2.7 Conclusion . . . 30

3 A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning 31
- 3.1 Introduction . . . 33
- 3.2 CausalWorld Benchmark . . . 35
- 3.3 Related Work . . . 40
- 3.4 Experiments . . . 41
- 3.5 Conclusion . . . 44

4 On the Transfer of Disentangled Representations in Realistic Settings 47
- 4.1 Introduction . . . 49
- 4.2 Related Work . . . 50
- 4.3 Scaling Disentangled Representations to Complex Scenarios . . . 52
- 4.4 Framework for the Evaluation of OOD Generalization . . . 56
- 4.5 Benefits and Transfer of Structured Representations . . . 57
- 4.6 Conclusion . . . 60

5 The Role of Pretrained Representations for the OOD Generalization of RL Agents 63
- 5.1 Introduction . . . 65
- 5.2 Background . . . 67
- 5.3 Study Design . . . 69
- 5.4 Results . . . 72
- 5.5 Other Related Work . . . 79
- 5.6 Conclusion . . . 79

6 Discrete Key-Value Bottleneck **81**
6.1 Introduction . . . 83
6.2 Encode, Processing via Discrete Key-Value Bottleneck, Decode . . . 85
6.3 Theoretical Analysis . . . 86
6.4 Related Work . . . 89
6.5 Experiments . . . 90
6.6 Limitations and Extensions . . . 96
6.7 Conclusion . . . 97

7 Backward-Compatible Prediction Updates: A Probabilistic Approach **99**
7.1 Introduction . . . 101
7.2 The Prediction Updates Problem Setting . . . 104
7.3 Our Method . . . 105
7.4 Discussion: Extensions and Limitations . . . 109
7.5 Related Work . . . 110
7.6 Experiments . . . 111
7.7 Conclusion . . . 119

8 Conclusion **121**
8.1 Recap . . . 121
8.2 Key Contributions . . . 121
8.3 Broader Context and Future Directions . . . 124
8.4 Personal Reflections . . . 128

Appendix **129**

A Related Projects **131**
A.1 Multi-Megabase Scale Genome Interpretation with Genetic Language Models . . . 131
A.2 Compositional Multi-Object Reinforcement Learning with Linear Relation Networks 133
A.3 Visual Representation Learning Does Not Generalize Strongly Within the Same Domain . . . 135

B Appendix to Chapter 2 **137**
B.1 Proof of Proposition . . . 137
B.2 Implementation Details . . . 138
B.3 Additional Results - Unsupervised Disentanglement under Correlated Factors . . . 142
B.4 Additional Results - Recovering Factorized Latent Code . . . 149

C Appendix to Chapter 3 **155**
C.1 Observations . . . 155
C.2 TriFinger Platform . . . 157
C.3 Task Generators . . . 157
C.4 Training Details . . . 160
C.5 Additional Results . . . 160
C.6 Causality in Reinforcement Learning . . . 162

D Appendix to Chapter 4 **163**
D.1 Implementation Details . . . 163
D.2 Additional Results . . . 166

E Appendix to Chapter 5 **173**
E.1 Implementation Details . . . 173
E.2 Additional Results . . . 174

F Appendix to Chapter 6 **183**
F.1 Proof of Theorem . . . 183
F.2 Why do random down-projection matrices help? . . . 186
F.3 Implementation Details. . . . 186
F.4 Additional Experiments . . . 188
F.5 Additional Analysis . . . 193

G Appendix to Chapter 7 **199**
G.1 Assumption of Conditionally-Independent Classifiers . . . 199
G.2 Backward Compatibility Scores . . . 199
G.3 Robustness to Random and Adversarial Model Sequences . . . 200
G.4 Role of Confusion Matrix Estimates . . . 202
G.5 Experiments on CIFAR-10 . . . 205
G.6 On Peaking Accuracies . . . 208
G.7 Role of the Selection Strategy . . . 208
G.8 Evolution of the Stored Label Distribution . . . 212
G.9 Reducing re-evaluations matters at scale . . . 213
G.10 Result Tables of Main Experiments . . . 213
G.11 On Utilizing Uncalibrated Soft Labels . . . 217

Bibliography **239**

1 Introduction

1.1 Distribution Shifts . . . 2
1.2 Generalization and Adaptation Strategies . 5
1.3 Outline 8

Imagine a child in a playroom, effortlessly constructing a detailed structure from an assortment of blocks – cubes, spheres, and pyramids – each differing in size, weight, and texture. This seemingly simple activity demonstrates a remarkable human ability: to perceive, adapt, and manipulate physical objects in complex ways, guided by a vast variety of sensory input and cognitive skills. Such an intuitive and adaptive approach to problem-solving represents a significant challenge for artificial intelligence (AI) [14, 15].

Since the beginnings of computer science, building machines that reason and behave like humans has been the grand goal in artificial intelligence research [16, 17]. Machine learning (ML), a cornerstone of AI, attempts to achieve this by creating models that predict a desired output Y from given inputs X. These inputs might be images, sensor readings, or any other observational data, while the output could range from class labels to the next actions of a robot. Propelled by the ever-increasing availability of computing and data resources, artificial neural networks have emerged as the backbone of recent advances in AI [18].

Neural networks transform raw inputs X through a sequence of simple yet interconnected functions to an output variable Y, thereby approximating complex functions [19–22]. Fundamental to their learning process is the concept of back-propagation, a method used to train these networks by adjusting their parameters in response to errors [23, 24]. This training approach, coupled with the development of functional layers like convolutional functions, residual blocks, and attention mechanisms, has enabled particularly deep neural networks to learn exceptionally powerful tasks and intermediate representations of data [25, 26]. This has led to profound impacts on AI, with neural networks setting new benchmarks and solving challenging tasks across a diverse range of fields such as image processing [25, 27–29], natural language processing [26, 30–33], strategy games and cognition [34–36], medicine and biology [37, 38], and robotics and control [39–41].

Despite these advances, AI systems still struggle to emulate the adaptability and generalization capabilities inherent in humans across their whole spectrum. From the change in lighting conditions affecting visual perception to the conceptual drift in task requirements, these shifts represent critical challenges in current machine learning models [42, 43].

[14]: Fischer (1980), 'A theory of cognitive development: The control and construction of hierarchies of skills.'
[15]: Gopnik et al. (1999), *The scientist in the crib: Minds, brains, and how children learn.*
[16]: Turing (1950), 'Computing Machinery and Intelligence'
[17]: Rosenblatt (1958), 'The perceptron: a probabilistic model for information storage and organization in the brain.'

[18]: LeCun et al. (2015), 'Deep learning'

[23]: Rumelhart et al. (1986), 'Learning representations by back-propagating errors'
[24]: LeCun et al. (1998), 'Gradient-based learning applied to document recognition'

[25]: He et al. (2016), 'Deep residual learning for image recognition'
[26]: Vaswani et al. (2017), 'Attention is all you need'

[42]: Hendrycks et al. (2021), 'The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization'
[43]: Geirhos et al. (2020), 'Shortcut learning in deep neural networks'

1.1 Distribution Shifts

The success of many machine learning models, including neural networks, is primarily attributed to their performance in data settings that are *i.i.d. - independent and identically distributed*. The *i.i.d.* assumption states that the data points during training and testing are drawn from the same joint probability distribution $P(X, Y)$. Whenever this assumption holds, statistical learning theory provides a theoretical framework to better understand a learned function's generalization under new data [44]. Given enough training data, modern neural networks excel when the training and test data share the same probability distribution, or more formally, $P_{\text{train}}(X, Y) = P_{\text{test}}(X, Y)$. See Figure 1.1 (left) for an example. The *i.i.d.* assumption simplifies the process of learning and predicting, as it assumes that the patterns learned from the training data will apply equally to new, unseen data points (X, Y). To better understand the basis of such shared (and not shared) patterns, we need a more precise characterization of the joint data distribution and the generating processes involved. To begin with, the joint data distribution can be viewed as a decomposition of $P(X, Y) = P(Y|X)P(X)$ with the marginal distribution $P(X)$ capturing the process that generates a new input X and the conditional distribution $P(Y|X)$ capturing the probability of outcome Y given X. Learning under *i.i.d.* conditions assumes that both of these quantities are static, i.e., our training and testing data was generated from the same static $P(X)$, and the task-conditional $P(Y|X)$ remains fixed.

[44]: Vapnik (1999), *The nature of statistical learning theory*

Although machine learning has undoubtedly made significant advancements within this *i.i.d.* setting, if these conditions are violated – often summarized as *Out-of-Distribution Shifts* – these models frequently encounter challenges and struggle in new, unfamiliar scenarios. [42, 43]. Still, for building truly reliable ML models, they should likewise be as adaptive and generalizing as humans when they face entirely new situations, i.e., if $P_{\text{train}}(X, Y) \neq P_{\text{test}}(X, Y)$. A child who has learned to build a tower using red cubes can easily transfer their skills to building towers with green cubes or even creating a tower using blocks of different shapes and sizes. Humans quickly adapt to new requirements by utilizing their prior knowledge in different contexts. Nevertheless, why is it that human intelligence and cognition are highly adaptive and machine learning is not (yet)?

To approach this question, it is worth recognizing that we make use of a much more detailed background knowledge and understanding of how the world around us works. We do not assume that we will always observe what we are used to; we intuitively know $P(Y|X)$ can change, and so can $P(X)$. We have a mental model of the world that we extend and update based on additional

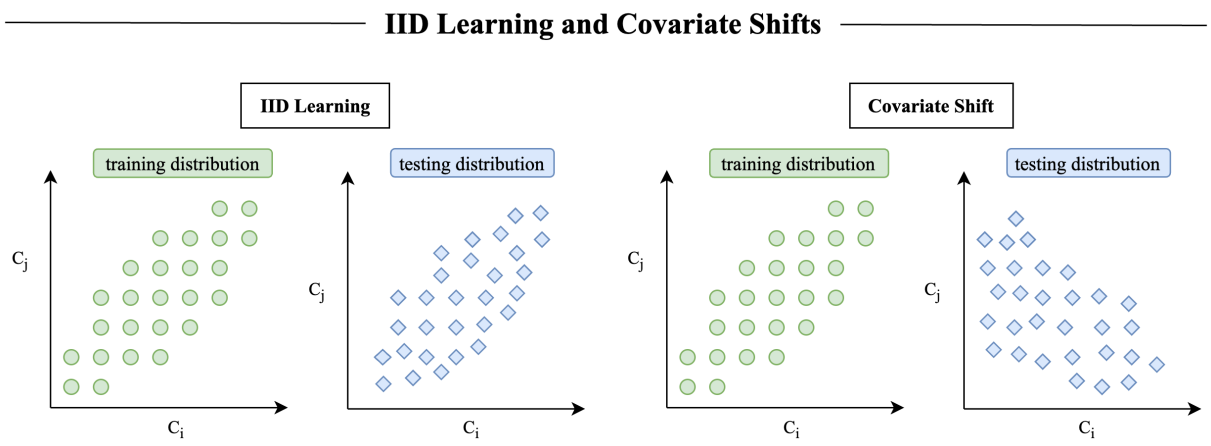


Figure 1.1: A schematic illustration of an *i.i.d.* learning scenario with $P_{\text{train}}(C_1, C_2) = P_{\text{test}}(C_1, C_2)$ and a *covariate shift* in the distribution of the latent variables with $P_{\text{train}}(C_1, C_2) \neq P_{\text{test}}(C_1, C_2)$.

information. We know that underlying variables and relations among them affect our observation, such as the lighting conditions or the physical and visual properties of different objects. Assuming that the marginal input data distribution $P(X)$ is the result of an underlying data-generative process $P(X|C)$ with underlying variables C, i.e., $P(X) = \int P(X|C)P(C)\,dC$, we can already anticipate how observational data may change. We often refer to such underlying variables as *latent variables*.

Causality and in particular *Structural Causal Models (SCMs)* offer a framework for representing the causal relationships, often referred to as *causal mechanisms* between such latent variables [45]. When a distribution shift happens, most of these mechanisms should remain unchanged with only a sparse subset of them changing, even though this might lead to substantial superficial changes in observed data X [46]. If the lighting conditions change, only the mechanisms responsible for the visual appearance of the objects change, while the physical interactions among objects remain the same. *SCMs* provide the theoretical foundation with assumptions that allow us to ultimately quantify and formalize a distribution shift and the degree of feasible generalization in the first place. By building on some inferred understanding of such learned causal mechanisms, humans can cope much better with such distribution shifts as they have a better model of the causal implication of interventions. We often anticipate how changing one factor would affect others. One of the most frequent distribution shifts is the so-called *covariate shift*. A covariate shift occurs in the distribution of these latent variables, more formally $P_{\text{train}}(C) \neq P_{\text{test}}(C)$, while the conditional distribution remains the same*. See Figure 1.1 (right) for an example. Imagine a child who has learned to build structures

[45]: Pearl (2009), *Causality*

* Note that a covariate shift may also lead to changes in $P(Y)$ which are commonly referred to as label or target shifts.

with blocks under well-lit conditions. Now, the child faces building the same structure in a dimly lit room. The input distribution $P(X)$ of visual information X changes due to the altered lighting with a fundamental change in $P(C)$, but the conditionals, which involve the observational data-generating process and logic of building the structure, remain the same. Similarly, the child can extrapolate and interpolate its skills along various values of such variables, which we often refer to as *zero-shot generalization* in machine learning, e.g., building with cube sizes the child has never seen. These skills extend to effortlessly mastering tasks with unseen combinations, such as green cylinders when only having played with green cubes and red cylinders. See Figure 1.2 for a schematic illustration of out-of-distribution generalization examples.

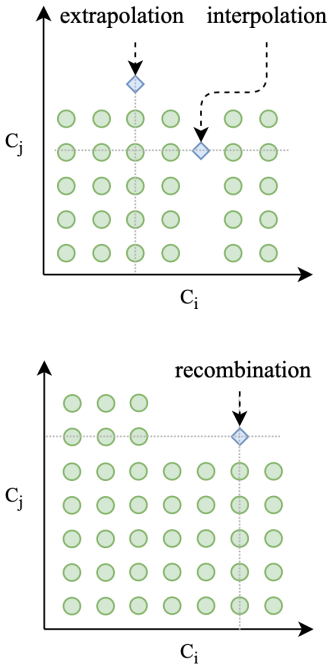


Figure 1.2: Various examples of expected out-of-distribution generalizations in the latent variables such as recombination, extrapolation or interpolation along different variable axes.

Of course, generalization can only be expected as long as there remains a level of shared information and mechanisms between the distributions before and after the distribution shift. If a child masters building a structure with physical blocks, they may also be able to build a similar structure in a virtual environment or with a different set of building materials. This demonstrates an ability to transfer skills learned in one context (physical blocks in a familiar setting) to a different but related context (virtual simulation or other materials). Even if the simulation does not perfectly mirror the real world, we can still apply our knowledge as long as it represents a sufficiently similar environment.

We expect humans to cope with even more severe distribution shifts, i.e. due to interventions. Imagine a child engaged in a virtual building block simulation where they are accustomed to constructing stable structures with gravity as a guiding force. Suddenly, the simulation introduces an intervention where the friction of the blocks' material is significantly reduced, akin to surfaces becoming as slick as ice, i.e., $P(X, Y|\text{do(friction=slippery)})$. While the fundamental dynamics of block interaction change dramatically, an attentive child can quickly adapt their building strategy to this new scenario. This situation is related to experiencing a *concept drift*, where the next actions need to be adjusted, in other words the conditional $P(Y|X)$ changes, challenging the child to rethink their approach to the task.

Similarly, humans are not required to learn with every new training example selected *i.i.d.*, as is the case for modern neural network training. We can learn under changing conditions, i.e. persistent drifts in the training distribution, sometimes referred to as *continual learning*. Let us think about a scenario where a child spends an extended period learning to construct complex structures using a particular set of building blocks. After mastering this, the child switches to building entirely different structures with a new set of blocks or materials, like clay or magnetic tiles. The child could

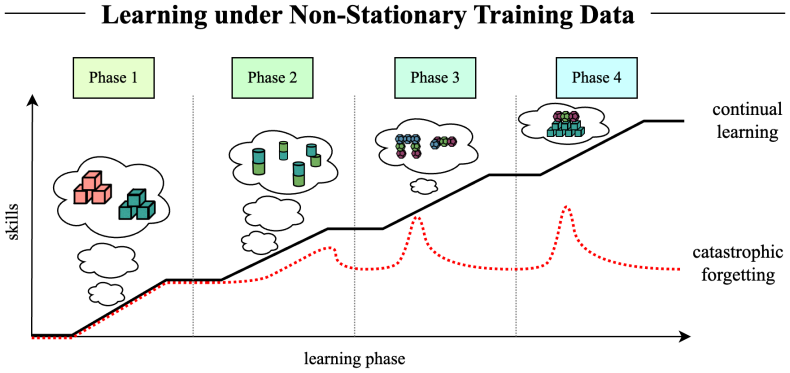


Figure 1.3: An example of continual learning, where different tasks are presented only consecutively during the learning phase. Naive training of standard neural networks causes severe catastrophic forgetting of previously learned skills (red line), while humans can incrementally learn new skills without severe forgetting (black line).

even do entirely different things in-between such as reading a book. Remarkably, the child is able to retain their earlier skills while integrating this new knowledge, effectively updating their mental models of construction without substantial forgetting; see Figure 1.3 for an illustration. This ability to continually update our understanding and skills with new information reflects a hallmark of human learning.

In this thesis, we will primarily focus on better understanding many of such distribution shifts and how we can build ML models that can address a variety of these out-of-distribution scenarios.

1.2 Generalization and Adaptation Strategies

Building ML models that exhibit the generalization and adaptability seen in human cognition is one central open goal in artificial intelligence. This objective necessitates developing models capable of abstracting data complexities into manageable representations. One pivotal path to approach this is the concept of deep representation learning [47]. Just as a child recognizes and operates within a world of distinct objects governed by physical laws, an AI model must learn to abstract and distill data into representations that encapsulate the underlying structure and hierarchy of the environment [48–50]. This involves not only recognizing patterns in data but also understanding the natural grounding and the individual mechanisms that govern their environment [46]. Therefore, a popular goal is to develop models that learn to distill this complex information into useful, generalizable abstract representations through exposure to diverse, labeled, and unlabeled data. One key strategy to achieve this involves the *pre-training* of neural networks on large and often unlabeled but related datasets as a first step. Thereby, the neural networks ought to develop stable, fundamental representations that provide a scaffold for learning new tasks and adapting to new situations. To some degree, this connects with the idea of *sim-to-real* transfer, where initially, ML

[47]: Bengio (2012), 'Deep learning of representations for unsupervised and transfer learning'

[48]: Goodfellow et al. (2016), *Deep learning*

[49]: Bengio et al. (2013), 'Representation learning: A review and new perspectives'

[50]: Lake et al. (2017), 'Building machines that learn and think like people'

[46]: Schölkopf et al. (2021), 'Toward causal representation learning'

models are being trained on vast amounts of data from a simulated environment with a distribution $P_{\text{sim}}(X, Y)$. The pre-trained ML model is then expected to generalize or adapt more quickly to the real-world distribution $P_{\text{real}}(X, Y)$.

While neural networks are proficient at detecting correlations and patterns in large datasets, they often lack the capacity for abstract reasoning and interpretation that come with well-defined causal models and variables, sometimes called symbols. Therefore, one promising approach to make ML models more robust under distribution shifts is to endow them with principles of causality [51]. A child understands that gravity uniformly affects all blocks, regardless of color or shape. Similarly, ML models must detect and adapt to underlying causal relationships within data. ML models operating on causal models should be able to anticipate interventions by better capturing the changing underlying mechanisms. For example, an ML model that understands the causal impact of gravity on block structures can adapt its strategies when faced with a scenario where gravity is altered or absent.

[51]: Schölkopf et al. (2012), 'On causal and anticausal learning'

However, identifying causal factors within real-world data poses a significant challenge, as these variables are rarely known and obvious [52]. Neural networks bring the advantage of learning from data directly, but the learned features may not always reflect causal relationships and underlying generative factors. *Disentanglement learning* is a term that comprises these different but strongly connected goals, one of which is learning representations that reflect causal structures. Another one is separating underlying generative factors in training data, like distinguishing between the shape, size, and color of blocks to understand better their individual functions in tasks [53]. Under a constrained setting, this can formally be described by attempting to infer a representation of factors $\mathbf{c}$ for which their joint distribution factorizes, i.e.,

[52]: Peters et al. (2017), *Elements of causal inference: foundations and learning algorithms*

[53]: Bengio (2013), 'Deep learning of representations: Looking forward'

$$p(\mathbf{c}) = p(c_1, c_2, ..., c_n) = \prod_{i=1}^{n} p(c_i).$$

Popular methods to learn such representations are based on the framework of so-called variational autoencoders (VAE) [54, 55]. Within this dissertation, we will study various generalization aspects of VAE-based models. In a VAE, the high-dimensional observations $\mathbf{x}$ are modeled as being generated from latent features $\mathbf{z}$ according to the probabilistic model $p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. The latent features $\mathbf{z}$ are assumed to take on the role of the unknown hidden variables $\mathbf{c}$ underlying the data generative process. The prior $p(\mathbf{z})$ is typically fixed to be an isotropic Gaussian distribution. At the same time, the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ and the approximate posterior

[54]: Kingma et al. (2014), 'Auto-encoding variational Bayes'
[55]: Rezende et al. (2014), 'Stochastic backpropagation and approximate inference in deep generative models'

$q_\phi(\mathbf{z}|\mathbf{x})$ [1] are parameterized by neural networks with parameters θ and ϕ respectively. Both the decoder (the generative model p_θ) and the encoder (the posterior q_ϕ) are optimized by maximizing the variational lower bound (ELBO) of the log-likelihood $\log p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$ of data $D = \{\mathbf{x}^{(i)}\}_{i=1}^N$:

$$\begin{aligned}
\log p_\theta(D) &= \log p_\theta(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}) \\
&= \log \int_{\mathbf{z}} p_\theta(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)} \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \\
&\geq \mathcal{L}^{\text{ELBO}}_{\theta,\phi}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}) \\
&= \sum_{i=1}^{N} \left(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] - D_{\text{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) \| p(\mathbf{z})\big) \right)
\end{aligned}$$

1: Canonically this is also being modeled by a Gaussian distribution with a diagonal covariance matrix

Here, $D_{\text{KL}}(\cdot\|\cdot)$ represents the Kullback-Leibler divergence between the encoder model and the latent prior. The learned latent dimensions of VAEs often tend to align with meaningful factors of the underlying data at least locally. This phenomenon can be explained by local approximations of the PCA directions of the underlying data [56]. Without additional inductive biases on both the models and the data, however, this VAE objective alone does not explicitly enforce any structure or semantic alignment on the latent space except for similarity in KL divergence to the prior $p(\mathbf{z})$. Therefore, the research community has proposed various further modifications that incorporate suitable disentanglement-imposing regularizers such as β-VAE, AnnealedVAE, FactorVAE, β-TCVAE, and DIP-VAE [57–62]. Recent studies have highlighted some challenges in achieving these goals within idealized experimental setups. [63–65]. In the initial chapters of this dissertation, we will focus on how disentanglement methods and their learned representations work in more complex real-world scenarios beyond idealized benchmark settings. Generally, achieving disentangled representations could lead to models that are not only more interpretable but also adaptable, enabling quicker learning of new tasks by leveraging an improved understanding of existing data structures [66–70]. Beyond the learning of disentangled representations, it remains of high interest to combine the strengths of neural networks and causality, e.g., via causal representation learning to enhance further the adaptability and interpretability of ML models [46, 51]. Such hybrid approaches seek to combine the pattern recognition capabilities of neural networks with the logical precision and symbolic grounding of causality, pursuing a balance that allows for selective and interpretable model adjustments based on new information.

[56]: Rolinek et al. (2019), 'Variational Autoencoders Recover PCA Directions (by Accident)'

[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'

[58]: Burgess et al. (2018), 'Understanding disentangling in beta-VAE'

[59]: Kim et al. (2018), 'Disentangling by factorising'

[60]: Chen et al. (2018), 'Isolating Sources of Disentanglement in Variational Autoencoders'

[61]: Kumar et al. (2018), 'Variational inference of disentangled latent concepts from unlabeled observations'

[62]: Mathieu et al. (2019), 'Disentangling Disentanglement in Variational Auto-Encoders'

[63]: Zietlow et al. (2021), 'Demystifying inductive biases for (beta-) VAE based architectures'

[64]: Klindt et al. (2021), 'Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding'

[65]: Locatello et al. (2019), 'Challenging common assumptions in the unsupervised learning of disentangled representations'

[46]: Schölkopf et al. (2021), 'Toward causal representation learning'

[51]: Schölkopf et al. (2012), 'On causal and anticausal learning'

Finally, an often overlooked aspect of adaptive ML models is the ability to estimate uncertainty. This skill is crucial for models facing new or changing environments, indicating confidence in their predictions. In scenarios similar to a child building in unfamiliar lighting conditions, a ML models must recognize the limits of its knowledge and adapt its strategies accordingly. Effective uncertainty estimation is essential for making reliable decisions, especially in applications with significant consequences, ensuring that models can identify when they are operating outside their knowledge base.

1.3 Outline

The rest of this thesis is structured as follows. Each chapter covers a publication co-authored during my PhD, addressing a subset of the distribution shifts and generalization strategies discussed above. The chapters are self-contained and adopted almost verbatim from their originally published conference versions with only minor changes. Throughout this dissertation, we aim to deepen our understanding of how machine learning can evolve to become genuinely adaptive and resilient in the face of dynamic and realistic data environments.

For the reader's benefit, this dissertation assumes a foundational understanding of key concepts in machine learning and artificial intelligence. Familiarity with deep learning and neural networks, representation learning, variational inference, reinforcement learning, Bayesian statistics and related topics is presumed.

Navigating Correlations in Disentangled Representation Learning In Chapter 2, we investigate the challenges of identifying independent factors of variation in data, especially when the causal variables are not statistically independent at train time. This chapter will explore if prominent disentanglement approaches can infer the underlying factors $\mathbf{C}$ when applied to synthetically correlated data where $P(X) = \int P(X|\mathbf{C})P(\mathbf{C})d\mathbf{C}$ changes because $P(\mathbf{C})$ exhibits a correlation structure. First, we present a large-scale empirical study to understand how these approaches cope with systematic correlations in the dataset. Second, we propose various adaptation strategies using only very few or weak labels for mitigating troublesome shortcomings, e.g., discrimination in downstream fairness applications. This chapter ties into the broader theme of adaptability and generalization in ML by demonstrating the need for flexible models that can effectively manage and adapt to correlated, real-world data, a fundamental aspect of handling distribution shifts.

Chapter 2 is based on **Träuble** et al. [1].

On Disentangled Representations Learned from Correlated Data

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf and Stefan Bauer.

ICML 2021: International Conference on Machine Learning (*selected for a long oral talk*).

Evaluating Generalization in RL with Causal Interventions Chapter 3 introduces *CausalWorld*, a benchmark for studying the transfer of reinforcement learning (RL) agents' learned skills toward new environments. This chapter presents a robotic manipulation environment, facilitating the learning of acting within a real-world, thus causally grounded, dynamical system. The tasks are inspired by how children learn to build complex structures, forming a first understanding of our world's principles. The unique feature of *CausalWorld* is its support for interventions on all causal variables, enabling precise control over task similarities and creating training and evaluation distributions for broad generalization challenges. As an example, one can invoke a broad range of interventional distributions. The chapter will examine the generalization capabilities of RL agents to changes in task parameters, therefore addressing one critical aspect towards adaptability under intervention-induced distribution shifts.

Chapter 3 is based on **Ahmed** et al. [2].

Causalworld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning

Ossama Ahmed*, **Frederik Träuble***, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf and Stefan Bauer.

ICLR 2021 International Conference for Learning Representations.

OA and FT were shared first authors. Asterisk denotes equal contribution.

Bridging Sim-to-Real Gaps with Disentangled Representations Chapter 4 focuses on the scalability and real-world applicability of disentangled representations within the robotics setup introduced in Chapter 3. The chapter introduces a novel high-resolution dataset for disentanglement learning that presents a more complex, correlated, and realistic challenge compared to previous datasets, i.e., a real-world manifestation of a correlation in $P(C)$. It is designed to test the efficacy of disentangled representations both in simulated and real-world environments, particularly under conditions where the encoder is either in-distribution or out-of-distribution (OOD). This exploration is crucial for understanding how well disentanglement techniques can generalize and adapt to real-world scenarios that exhibit complex, non-ideal characteristics. We will discuss new architectures aimed at scaling disentangled representation learning to these high-resolution, realistic settings and present a comprehensive empirical study on this dataset. The chapter ties into the broader narrative of generalization across distribution shifts, emphasizing the importance of disentanglement as a potential predictor for OOD task performance and its role in enhancing the generalization and adaptation capabilities of machine learning models in complex real-world environments.

Chapter 4 is based on **Dittadi** et al. [3].

On the Transfer of Disentangled Representations in Realistic Settings

Andrea Dittadi*, **Frederik Träuble***, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer and Bernhard Schölkopf

ICLR 2021: International Conference on Learning Representations.

AD and FT were shared first authors. Asterisk denotes equal contribution.

Pre-trained Representations for Real-World Distribution Shifts Chapter 5 explores the challenge of building sample-efficient RL agents that can generalize out-of-oistribution in real-world settings, a crucial step towards higher-level cognition. This chapter explores the promising approach of utilizing low-dimensional, pre-trained and disentangled representations as a foundation for efficient downstream learning and generalization in RL environments. Through an extensive study involving hundreds of pre-trained representation models and more than 10,000 derived RL policies on CausalWorld tasks, we investigate how various properties of these representations impact the OOD generalization capabilities of downstream RL agents. We specifically fo-

cus on their robustness to realistic distribution shifts, including the complex sim-to-real transfer. Additionally, we examine the potential of using simple downstream proxy tasks as predictors for the generalization performance of RL agents in diverse OOD scenarios. The chapter aligns with the dissertation's focus on adaptability and generalization, highlighting the potential of pre-trained representations to enhance RL agents' adaptability in facing the unpredictable and varied conditions of the real world.

Chapter 5 is based on **Träuble** et al. [4].

The Role of Pretrained Representations for the OOD Generalization of RL Agents

Frederik Träuble*, Andrea Dittadi*, Manuel Wüthrich, Felix Widmaier, Peter Vincent Gehler, Ole Winther, Francesco Locatello, Olivier Bachem, Bernhard Schölkopf and Stefan Bauer.

ICLR 2022: International Conference on Learning Representations.

AD and FT were shared first authors. Asterisk denotes equal contribution.

Addressing Catastrophic Forgetting in Continual Learning In Chapter 6 of this dissertation, we will leave behind the domain of a known underlying data structure and explore challenges in scenarios characterized by continual learning environments. Specifically, we will address one of the most difficult class incremental learning scenarios with substantial shifts in $P(Y)$ over the course of learning. The chapter will introduce a discrete key-value bottleneck mechanism to allow for sparse, localized, and context-dependent model updates under changing data distributions. The novel model architecture is designed to mitigate the effects of catastrophic forgetting — a key issue when dealing with continual learning and distribution shifts.

Chapter 6 is based on **Träuble** et al. [5].

Discrete Key-Value Bottleneck

Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael Curtis Mozer, Kenji Kawaguchi, Yoshua Bengio and Bernhard Schölkopf.

ICML 2023: International Conference on Machine Learning.

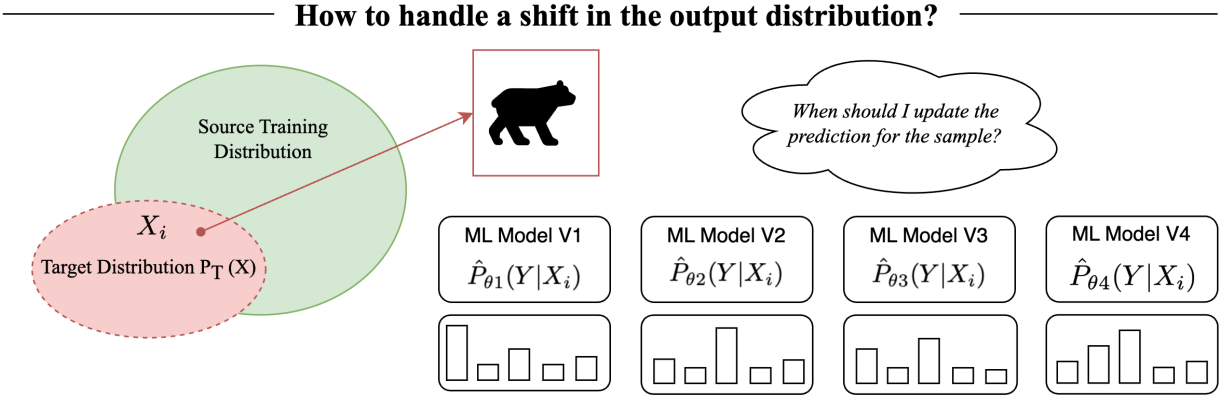


Figure 1.4: Beyond traditional distribution shifts, we will also study the problem of how to deal with a shift in the output distribution. Such shifts could happen when an integrated ML model within a bigger prediction pipeline is being updated with presumably better models.

Prediction Updates under Model Shifts. Finally, compared to previous chapters, focusing on shifts in the input data distribution, Chapter 7 explores a particular challenge with shifts in the output distribution. Specifically, we introduce and formalize the *Prediction Update Problem*, a critical issue arising when updating machine learning systems in real-world applications, i.e., when the learned $\hat{P}_\theta(Y|X)$ changes. See the illustration in Figure 1.4 for an example in the case of classification. This exploration is connected to the broader themes of adaptability under distribution shifts, emphasizing the need for predictive systems to remain up to date without compromising the integrity and reliability of their outputs. The chapter is illustrating how to efficiently navigate the balance between adopting advancements in predictive ML models and ensuring the stability and consistency of backward-compatibility that is required in many practical applications.

Chapter 7 is based on **Träuble** et al. [6].

Backward-Compatible Prediction Updates: A Probabilistic Approach

Frederik Träuble, Julius Von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf and Peter Gehler.

NeurIPS 2021: Advances in Neural Information Processing Systems.

On Disentangled Representations Learned from Correlated Data

2

Abstract

The focus of disentanglement approaches has been on identifying independent factors of variation in data. However, the causal variables underlying real-world observations are often not statistically independent. In this work, we bridge the gap to real-world scenarios by analyzing the behavior of the most prominent disentanglement approaches on correlated data in a large-scale empirical study (including 4260 models). We show and quantify that systematically induced correlations in the dataset are being learned and reflected in the latent representations, which has implications for downstream applications of disentanglement such as fairness. We also demonstrate how to resolve these latent correlations, either using weak supervision during training or by post-hoc correcting a pre-trained model with a small number of labels.

About This Chapter

This chapter has been adapted from **Träuble** et al. [1].

On Disentangled Representations Learned from Correlated Data

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf and Stefan Bauer.

ICML 2021: International Conference on Machine Learning (*selected for a long oral talk*).

2.1 Introduction 15
2.2 Background 17
2.3 The Problem with Correlated Data 18
2.4 Unsupervised Disentanglement under Correlated Factors 21
2.5 Recovering Factorized Latent Codes 26
2.6 Other Related Work . . 29
2.7 Conclusion 30

Author Contributions based on CRediT (Contributor Roles Taxonomy)

Frederik Träuble: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, **Elliot Creager**: Conceptualization, Software, Investigation, Writing - Review & Editing **Niki Kilbertus**: Conceptualization, Investigation, Writing - Review & Editing **Francesco Locatello**: Conceptualization, Software, Methodology, Investigation, Writing - Review & Editing, Supervision **Andrea Dittadi**: Formal analysis Writing - Review & Editing, **Anirudh Goyal**: Writing - Review & Editing **Bernhard Schölkopf**: Conceptualization, Resources, Writing - Review & Editing, Supervision, Funding acquisition **Stefan Bauer**: Conceptualization, Methodology, Validation, Investigation, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration

2.1 Introduction

Disentangled representations promise generalization to unseen scenarios [67, 71], increased interpretability [72, 73], domain generalization [74] or better sim-to-real transfer [3] as well as a connection to robustness and causal inference [75].

While the advantages of disentangled representations have been well established, they generally assume the existence of natural factors that vary independently within the given dataset, which is rarely the case in real-world settings. As an example, consider a dataset containing images of various persons (see Figure 2.1). Higher-level factors of this representation, such as foot length and body height are in fact found to be statistically correlated [76, 77]. Humans can still conceive varying both factors independently across their entire range. However, only given a correlated dataset, learning systems may be tempted to encode both factors simultaneously in a single *size* factor. It is thus argued that what we actually want to infer are independent (causal) mechanisms [46, 78, 79].

A complex generative model can be thought of as the composition of independent mechanisms, which generate high-dimensional observations (such as images or videos). In the causality community, this is often considered a prerequisite to achieve representations that are robust to interventions upon variables determined by such models [80]. The notion of *disentangled* representation is one particular instantiation of this idea [49]. The goal of disentanglement learning is to find a representation of the data which captures all the ground-truth factors of variation (FoV) independently.

Despite the recent growth of the field, state-of-the-art disentanglement learners have mostly been trained and evaluated on datasets in which all FoV indeed vary independently from each other. Their performance remains unknown for more realistic settings where

[72]: Adel et al. (2018), 'Discovering interpretable representations for both deep generative and discriminative models'

[73]: Higgins et al. (2018), 'SCAN: Learning Hierarchical Compositional Visual Concepts'

[74]: Miladinović et al. (2020), 'Disentangled state space representations'

[3]: **Dittadi** et al. (2021), 'On the Transfer of Disentangled Representations in Realistic Settings'

[75]: Suter et al. (2019), 'Interventional Robustness of Deep Latent Variable Models'

[76]: Agnihotri et al. (2007), 'Estimation of stature by foot length'

[77]: Grivas et al. (2008), 'Correlation of foot length with height and weight in school age children'

[46]: Schölkopf et al. (2021), 'Toward causal representation learning'

[78]: Goyal et al. (2021), 'Recurrent independent mechanisms'

[79]: Leeb et al. (2022), 'Structure by architecture: Structured representations without regularization'

[80]: Peters et al. (2017), *Elements of Causal Inference - Foundations and Learning Algorithms*

[49]: Bengio et al. (2013), 'Representation learning: A review and new perspectives'

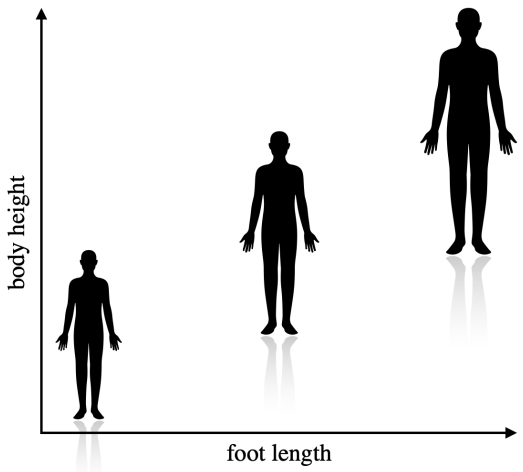


Figure 2.1: While in principle we would like to consider foot length and body height to be independent features of our model, they exhibit strong positive correlation in observed data.

FoV are correlated during training. Given the potential societal impact in the medical domain [81] or fair decision making [70, 82, 83], the evaluation of the usefulness of disentangled representations trained on correlated data is of high importance.

[81]: Chartsias et al. (2018), 'Factorised spatial representation learning: Application in semi-supervised myocardial segmentation'

[70]: Locatello et al. (2019), 'On the fairness of disentangled representations'

[82]: Madras et al. (2018), 'Learning adversarially fair and transferable representations'

[83]: Creager et al. (2019), 'Flexibly fair representation learning by disentanglement'

The idealized settings employed for validating disentanglement learning so far would typically build on a dataset that has as many persons with small feet and small body height as small feet and large body height. Realistic unlabeled datasets are rather unlikely to be of this type. It is thus an open question to what degree existing inductive biases from the encoder/decoder architecture, but most importantly these dataset biases, affect the learned representation. In this work, we introduce dataset correlations in a controlled manner to understand in detail to what degree state-of-the-art approaches can cope with such correlations.

To this end, we report a large-scale empirical study to systematically assess the effect of induced correlations between pairs of factors of variation in training data on the learned representations. To provide a qualitative and quantitative evaluation, we investigate multiple datasets with access to ground-truth labels. Moreover, we study the generalization abilities of the representations learned on correlated data as well as their performance in particular for the downstream task of fair decision making.

Contributions. We summarize our contributions:

- We present the first large-scale empirical study (4260 models)[1] that examines how modern disentanglement learners perform when ground truth factors of the observational data are *correlated*.
- We find that, in this case, factorization-based inductive biases are insufficient to learn disentangled representations. Existing methods fail to disentangle correlated factors of variation as the latent space dimensions become statistically entangled, which has implications for downstream applications of disentanglement such as fairness. We corroborate this theoretically by showing that, in the correlated setting, all generative models that match the marginal likelihood of the ground truth model are not disentangled.
- We propose a simple post-hoc procedure that can align entangled latents with only very few labels and investigate the usefulness of weakly-supervised approaches to resolve latent entanglement already at train time when only correlated data is available.

1: Each model was trained for 300,000 iterations on Tesla V100 GPUs. Reproducing these experiments requires approximately 0.79 GPU years.

2.2 Background

Current state-of-the-art disentanglement approaches use the framework of variational autoencoders (VAEs) [54, 55]. The (high-dimensional) observations $\mathbf{x}$ are modeled as being generated from latent features $\mathbf{z}$ according to the probabilistic model $p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. Typically, the prior $p(\mathbf{z})$ is fixed, while the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ and the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ are parameterized by neural networks with parameters θ and ϕ respectively. These are optimized by maximizing the variational lower bound (ELBO) of the log likelihood $\log p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$ of data $D = \{\mathbf{x}^{(i)}\}_{i=1}^N$:

$$\mathcal{L}_{ELBO} = \sum_{i=1}^{N} \Big(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] - D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) \| p(\mathbf{z})\big) \Big) \tag{2.1}$$

This objective does not enforce any structure on the latent space, except for similarity (in KL divergence) to the prior $p(\mathbf{z})$ (typically chosen as an isotropic Gaussian). Thus, no specific structure and semantic meaning of latent representations is encouraged by this objective. Consequently, various works propose new evaluation metrics to quantify different notions of disentanglement of the learned representations, as well as new disentanglement learning methods, such as β-VAE, AnnealedVAE, FactorVAE, β-TCVAE, and DIP-VAE, that incorporate suitable structure-imposing regularizers [57–62, 84].

Since unsupervised disentanglement by optimizing the marginal likelihood in a generative model is impossible [65], inductive biases like grouping information [85] or access to labels [66] is required. To address this theoretical limitation, methods have been proposed that only require weak label information [67, 86]. Changes in natural environments, which typically correspond to changes of only a few underlying FoV, can provide a weak supervision signal for representation learning algorithms [64, 78, 87–90]. In the absence of correlations it has been shown on common disentanglement datasets that this weak supervision facilitates learning disentangled representations [67, 86].

Most popular datasets in the disentanglement literature exhibit perfect independence in their FoV such as *dSprites* [57], *Cars3D* [91], *SmallNORB* [92], *Shapes3D* [59] or *MPI3D* variants [93]. At some level this is sensible as it reflects the underlying assumption in the inductive bias being studied. However, this assumption is unlikely to hold in practice as is also shown by Li et al. [94].

[54]: Kingma et al. (2014), 'Auto-encoding variational Bayes'
[55]: Rezende et al. (2014), 'Stochastic backpropagation and approximate inference in deep generative models'

[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'
[58]: Burgess et al. (2018), 'Understanding disentangling in beta-VAE'
[59]: Kim et al. (2018), 'Disentangling by factorising'
[60]: Chen et al. (2018), 'Isolating Sources of Disentanglement in Variational Autoencoders'
[61]: Kumar et al. (2018), 'Variational inference of disentangled latent concepts from unlabeled observations'
[62]: Mathieu et al. (2019), 'Disentangling Disentanglement in Variational Auto-Encoders'
[84]: Eastwood et al. (2018), 'A framework for the quantitative evaluation of disentangled representations'

[65]: Locatello et al. (2019), 'Challenging common assumptions in the unsupervised learning of disentangled representations'

[85]: Bouchacourt et al. (2018), 'Multi-level variational autoencoder: Learning disentangled representations from grouped observations'

[66]: Locatello et al. (2020), 'Disentangling Factors of Variations Using Few Labels'

[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'
[86]: Shu et al. (2020), 'Weakly Supervised Disentanglement with Guarantees'

[64]: Klindt et al. (2021), 'Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding'
[78]: Goyal et al. (2021), 'Recurrent independent mechanisms'
[87]: Földiák (1991), 'Learning invariance from transformation sequences'
[88]: Schmidt et al. (2007), 'Learning graphical model structure using L1-regularization paths'
[89]: Bengio et al. (2020), 'A meta-transfer objective for learning to disentangle causal mechanisms'
[90]: Ke et al. (2019), 'Learning neural causal models from unknown interventions'
[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'
[86]: Shu et al. (2020), 'Weakly Supervised Disentanglement with Guarantees'
[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'
[91]: Reed et al. (2015), 'Deep visual analogy-making'
[92]: LeCun et al. (2004), 'Learning methods for generic object recognition with invariance to pose and lighting'
[59]: Kim et al. (2018), 'Disentangling by factorising'
[93]: Gondal et al. (2019), 'On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset'
[43]: Geirhos et al. (2020), 'Shortcut learning in deep neural networks'
[94]: Li et al. (2019), 'Learning Disentangled Representation with Pairwise Independence'

2.3 The Problem with Correlated Data

Most work on learning disentangled representations assumes that there is an underlying set of independent ground truth variables that govern the generative process of observable data. These methods are hence predominantly trained and evaluated on datasets that obey independence in the true factors of variation by design, which we then consider to be the correct factorization. Formally, disentanglement methods typically assume $\mathbf{x} \sim \int_{\mathbf{c}} p^*(\mathbf{x}|\mathbf{c})p^*(\mathbf{c})d\mathbf{c}$ where the prior over ground truth factors $\mathbf{c}$ factorizes as

$$p^*(c_1, c_2, \ldots, c_n) = \prod_{i=1}^{n} p^*(c_i). \tag{2.2}$$

Note that this is the dataset-generating prior and distinct from the latent prior $p(\mathbf{z})$ of the model.

In the real world, however, we generally expect correlations in the collected datasets, i.e., the joint distribution of the ground truth factors $\{c_i\}_{i=1}^n$ does not factorize:

$$p^*(c_1, c_2, \ldots, c_n) \neq \prod_{i=1}^{n} p^*(c_i). \tag{2.3}$$

In this case, we speak of dependence between the random variables, also commonly referred to as correlation. Even though there might be correlation between multiple variables simultaneously, for a principled systematic analysis we will mostly consider pairwise correlations. Correlation between two variables can stem from various sources, e.g., from a direct causal relationship (one variable affects the other), or from unobserved circumstances (confounders) affecting both. Real-world datasets display many of these 'spurious' and often a priori unknown correlations [43, 94]. In the introductory example, various (potentially hidden) confounders such as age or sex may indeed affect both foot length and body height. Even though we expect such a dataset to exhibit strong correlations in these factors, we would like to model them independently of each other.

To establish some intuition for why such correlations pose problems for generative models attempting to learn a disentangled representation, assume two FoV c_1 and c_2 that are correlated in the dataset. In a perfectly disentangled representation, a single latent dimension z_{c_1} would model factor c_1 and z_{c_2} would model factor c_2. However, because our model has an independent latent prior, the resulting generative model would generate all combinations of the two correlated factors, i.e., put probability mass also outside of the training distribution. Hence, perfect disentanglement would lead

to a sub-optimal log likelihood. We can show this more formally as follows (see Appendix B.1 for a proof).

Proposition 2.3.1 *Consider* $p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z}$ *to be the latent variable model and let the true data distribution be* $p^*(\mathbf{x}) = \int_{\mathbf{c}} p^*(\mathbf{x} \mid \mathbf{c}) p^*(\mathbf{c}) d\mathbf{c}$ *where* $\mathbf{z} \in \mathcal{Z}$ *and* $\mathbf{c} \in \mathcal{C}$*. Assume that the fixed prior* $p(\mathbf{z}) = \prod_i p(z_i)$ *factorizes, the true FoV prior* $p^*(\mathbf{c}) \neq \prod_i p^*(c_i)$ *does not, and that there exists a smooth bijection* $f^* : \mathcal{C} \to \mathcal{Z}$ *with smooth inverse (i.e., a diffeomorphism) that transforms* $p^*(\mathbf{c})$ *into* $p(\mathbf{z})$*. Then, the likelihood* $p_\theta(\mathbf{x})$ *can be equal to the optimal likelihood* $p^*(\mathbf{x})$ *if and only if the representations* $\mathbf{z}$ *are entangled w.r.t. the true factors of variation* $\mathbf{c}$*.*

Note that we define a representation to be *disentangled* if the Jacobian of f^* is diagonal up to a permutation, in line with [65, 67]. Under the assumptions of this proposition, we established that a generative model that matches the true marginal likelihood $p^*(\mathbf{x})$ cannot be disentangled. This suggests that, if some FoV are correlated in the dataset and the prior $p(\mathbf{z})$ factorizes, methods that optimize a lower bound to the log likelihood might have a bias *against* disentanglement. While the impossibility result of Locatello et al. [65] states that there may be many generative models achieving the optimal likelihood for a ground-truth model with independent FoVs, Proposition 1 states that if the ground-truth prior is correlated, a disentangled representation will never achieve this optimal likelihood and therefore entangled representations are preferred.

[65]: Locatello et al. (2019), 'Challenging common assumptions in the unsupervised learning of disentangled representations'
[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'

The primary goal of this work is thus to empirically assess to what extent the additional inductive biases of state-of-the-art disentanglement methods still suffice to learn disentangled representations when the training data exhibits correlations.

Why not encode train-time correlations? The prior discussion raises the question why we would like to learn representations that do not encode data correlations. Why not encode foot size and body height in a single latent dimension when we are only shown data in which they are strongly correlated? Maximizing the log likelihood, which amounts to matching the data distribution at train time, would embrace such correlations, and it would later allow us to sample novel examples from this distribution. We now provide three major reasons why this may not always be the goal in real-world applications.

First, if we want to build models that generalize well across multiple tasks and dataset distributions the standard approach is myopic. A key promise of disentanglement is to learn meaningful and compact representations, in which relevant semantic aspects of the data are structurally disentangled and can be analyzed or

controlled independently from each other during inference or generation. From this perspective, resolving instead of exploiting correlations in the train data is desirable at test time, where such representations would remain robust and meaningful under distribution shifts. Returning to our running example, assume training data is constrained to a fixed sex and age bracket, both being confounders for foot length and body height [77]. If we could disentangle these factors despite the correlation, the model would arguably be more robust to distribution shifts, for example when testing on a different sex or different ages.

[77]: Grivas et al. (2008), 'Correlation of foot length with height and weight in school age children'

Second, we would like to be able to sample out-of-distribution (OOD) examples or intervene on individual factors independently of the others. Should a good model not be able to meaningfully reason about and imagine someone's foot length changing independently of their body height?

Lastly, disentangled representations are relevant in fairness settings, where representations are used in downstream tasks for consequential decisions that ought to be fair with respect to sensitive (protected) variables [70, 83]. Undesirable data correlations with such protected variables are a major problem in fairness applications. Therefore, it is crucial to evaluate the validity of factorization-based inductive biases to learn disentangled representations from correlated observational data.

[70]: Locatello et al. (2019), 'On the fairness of disentangled representations'
[83]: Creager et al. (2019), 'Flexibly fair representation learning by disentanglement'

How can we resolve these latent correlations? To resolve potential latent correlations, we will investigate two approaches. First, we explore the scenario in which few labels are given after training with the specific purpose of resolving these correlations. For this, we propose a simple approach in Section 2.5, which we find to be effective. Second, we employ the performance of the weakly-supervised approach of Locatello et al. [67]. They showed (on uncorrelated data) that access to pairs of observations which display differences in a known number of FoV (without knowing which ones specifically) suffices to learn disentangled representations. These additional weak assumptions render the generative model identifiable in contrast to unsupervised disentanglement and may arguably be indeed available in certain practical settings, e.g., in temporally close frames from a video of a moving robot arm where some factors remain unchanged.[2] We test the Ada-GVAE algorithm from Locatello et al. [67], which requires a pair of observations that differ in an unknown number of factors. To impose this structure in the latent space, the latent factors that are shared by the two observations are estimated from the k largest coordinate-wise KL divergences $D_{KL}(q_\phi(z_i|\mathbf{x}^{(1)})\|q_\phi(z_i|\mathbf{x}^{(2)}))$. They then maximize the following modified β-VAE [57] objective for the

2: On the other hand, in applications with fairness concerns it may be impossible to intervene on FoV representing sensitive attributes of individuals (sex, race, etc.); we refer to Madras et al. [69] and Kilbertus et al. [95] for a more complete discussion.

[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'

corr. dataset	data source	1st correlated FoV	2nd correlated FoV
A	Shapes3D	object size	azimuth
B	dSprites	orientation	x-position
C	MPI3D	1st DoF	2nd DoF
D	Shapes3D	object color	object size
E	Shapes3D	object color	azimuth

Table 2.1: Dataset variants with introduced correlations under investigation within this empirical study.

pair of observations

$$\sum_{i\in\{1,2\}} \mathbb{E}_{(\mathbf{x}^{(1)},\mathbf{x}^{(2)})}\Big[\mathbb{E}_{\tilde{q}_\phi^{(i)}(\hat{\mathbf{z}}|\mathbf{x}^{(1)},\mathbf{x}^{(2)})} \log(p_\theta(\mathbf{x}^{(i)}|\hat{\mathbf{z}})) - \beta D_{KL}\left(\tilde{q}_\phi^{(i)}(\hat{\mathbf{z}}|\mathbf{x}^{(1)},\mathbf{x}^{(2)})\|p(\hat{\mathbf{z}})\right)\Big], \tag{2.4}$$

with

$$\tilde{q}_\phi^{(i)}(\hat{z}_j|\mathbf{x}^{(1)},\mathbf{x}^{(2)}) = q_\phi(\hat{z}_j|\mathbf{x}^{(i)}) \tag{2.5}$$

for the latent dimensions z_j that are inferred to be changing and

$$\tilde{q}_\phi^{(i)}(\hat{z}_j|\mathbf{x}^{(1)},\mathbf{x}^{(2)}) = a(q_\phi(\hat{z}_j|\mathbf{x}^{(1)}), q_\phi(\hat{z}_j|\mathbf{x}^{(2)})) \tag{2.6}$$

for those inferred to be shared. The averaging function a forces the approximate posterior of the shared latent variable to be the same in the two observations.

2.4 Unsupervised Disentanglement under Correlated Factors

In this section, we present the key findings from our empirical study of unsupervised disentanglement learning approaches on a particular variant of correlated datasets. We start by outlining the experimental design of our studies in Section 2.4.1. Based on this, in Section 2.4.2 we present a latent space analysis of unsupervised disentanglement learners and find that factorization-based inductive biases are insufficient to learn disentangled representations from correlated observational data. Although these methods fail to disentangle correlated factors, in Section 2.4.3 we qualitatively show that they still generalize to FoV combinations that are out-of-distribution with respect to the training data.

2.4.1 Experimental Design

To begin to systematically understand the unknown behavior of SOTA disentanglement approaches on correlated data, we focus on linear correlations with Gaussian noise between pairs of FoV,

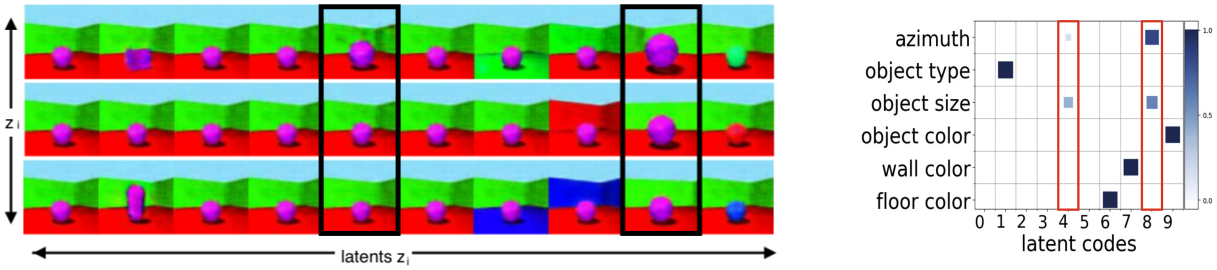


Figure 2.2: Left: Latent traversals for the model with best DCI score (commonly used metric for measuring disentanglement [84]) among all 180 models trained on strongly correlated ($\sigma = 0.2$) Shapes3D data (A). The traversals in latent dimensions 5 and 9 (highlighted in black) encode a mixture of azimuth and object size, reflecting the main correlation line of the joint distribution and a smaller, locally orthogonal one. **Right:** A heat map of the GBT feature importance matrix of this model indicates an entanglement of azimuth and object size encoded into both latent codes.

denoted by c_1 and c_2.[3] For our experiments we introduce five correlated dataset variants comprising dSprites, Shapes3D and MPI3D with correlations between single pairs of FoV. Table 2.1 shows the correlated FoV in each dataset variant and the names we refer to them by.

We then parameterize correlations by sampling the training dataset from the joint distribution

$$p(c_1, c_2) \propto \exp\left(-\frac{(c_1 - \alpha c_2)^2}{2\sigma^2}\right)$$

with $c_1 \in \{0, \ldots, c_1^{\max}\}$ and $c_2 \in \{0, \ldots, c_2^{\max}\}$ and $\alpha = c_2^{\max}/c_1^{\max}$. The correlation strength can be tuned by σ, for which we choose 0.2, 0.4, 0.7 in normalized units with respect to the range of values in $\{c_1, c_2\}$. Lower σ indicates stronger correlation. See Figure 2.4 for an example of $p(c_1, c_2)$ for correlating azimuth and object size in Shapes3D. Additionally, we study the uncorrelated limit ($\sigma = \infty$), which amounts to the case typically considered in the literature. All remaining factors are sampled uniformly at random.

We train the same six VAE methods as Locatello et al. [65], i.e., β-VAE, FactorVAE, AnnealedVAE, DIP-VAE-I, DIP-VAE-II and β-TC-VAE, each with 6 hyperparameter settings and 5 random seeds. These 180 models are trained on datasets A, B, and C with $\sigma \in \{0.2, 0.4, 0.7, \infty\}$ and datasets D and E with $\sigma \in \{0.2, 0.4\}$, totaling 2880 models.

2.4.2 Unsupervised Methods for Correlated Data

Here we assess the applicability of SOTA unsupervised learning approaches to correlated data. Are the factorization-based inductive biases introduced by these methods enough to disentangle correlated FoV although it is sub-optimal from a standard VAE perspective?

3: We emphasize that this seemingly narrow class of correlations already captures most relevant effects of more general correlations between FoV at train time on the learned representation. This is not to say that one can draw general conclusions about highly nonlinear correlated settings. However, rigorously understanding the linear case in a wide range of controlled settings already represents a considerable set of experiments and is a vital step in bridging the gap between highly idealized settings and real-world applications.

Corr. strength		$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.7$	$\sigma = \infty$ (uc)
Shapes3D (A)	object size - azimuth	0.38	0.26	0.13	0.08
	median uncorrelated pairs	0.09	0.09	0.09	0.08
dSprites (B)	orientation - position x	0.17	0.16	0.14	0.11
	median uncorrelated pairs	0.13	0.13	0.13	0.13
MPI3D (C)	First DOF - Second DOF	0.2	0.19	0.17	0.16
	median uncorrelated pairs	0.16	0.16	0.15	0.15

Table 2.2: The means of the pairwise entanglement scores for the correlated pair (red) and the median of the uncorrelated pairs. Stronger correlation leads to statistically more entangled latents compared to the baseline score without correlation (blue), thus uncovering still existent correlations in the latent representation.

Latent structure and pairwise entanglement. We start by visually inspecting latent traversals of some trained models on Shapes3D (A). For strong correlations ($\sigma = 0.2$ and $\sigma = 0.4$), we typically observe trained models with two latent codes encoding the two correlated variables simultaneously. In these cases, one of the latent codes corresponds to data along the major axis of the correlation line whereas the other latent code dimension manifests in an orthogonal change of the two variables along the minor axis. Perhaps unsurprisingly, a full traversal of the code corresponding to the minor axis often seems to cover only observations within the variance of the correlation line, i.e., in the training distribution. Figure 2.2 (left) shows this effect for the latent space of a model trained on Shapes3D (A) with strongest correlation ($\sigma = 0.2$). To quantify this observation, we analyze the importance of individual latent codes in predicting the value of a given ground truth FoV. An importance weight for each pair of {FoV, latent dimension} is computed by training a gradient boosting tree (GBT) classifier to predict the ground truth labels from the latents (10,000 examples). In the right panel of Figure 2.2, we show these importance weights for the model used to generate traversals in the left panel. The corresponding evaluation for a model trained on the same dataset with a much weaker correlation of $\sigma = 0.7$ does not reveal this feature visually (see Figure B.2 in Appendix B.3.1).

To support this claim empirically across the full study with all datasets, we calculate a pairwise entanglement score that allows us to measure how difficult it is to separate two factors of variation from their latent codes. This computation involves grouping FoV into pairs based on an ordering of their pairwise mutual information or GBT feature importance between latents and FoV; we refer to Appendix B.2 for a detailed description of this procedure. Table 2.2 shows that across all datasets the pair of correlated FoV has a substantially higher score than the median of all other pairs, indicating that they are harder to disentangle. This score decreases with weaker correlation, i.e., the pair becomes easier to disentangle for $\sigma \geq 0.7$. We conclude that models still disentangle weakly correlated factors, but inductive biases become insufficient to avoid latent entanglements for stronger correlations. In Appendix B.3.1 we further consolidate our claims by showing that the correlated pair is more entangled in the latent representation across all unsu-

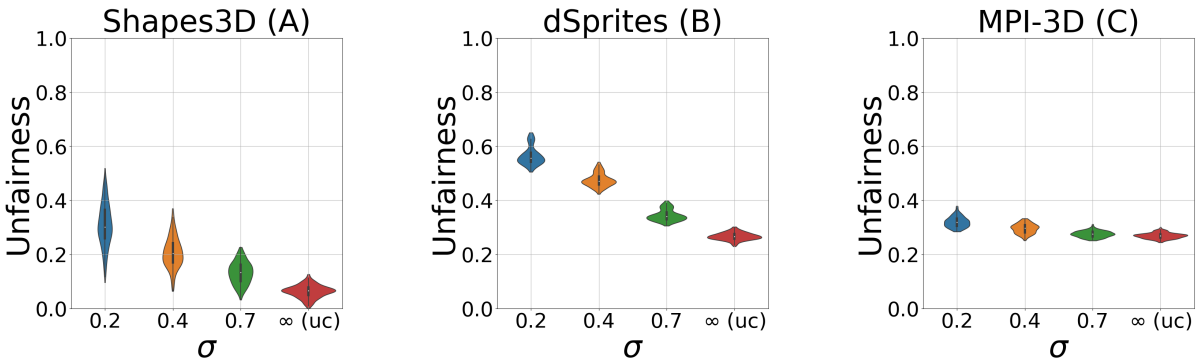


Figure 2.3: Stronger correlations in the train data lead to substantially elevated unfairness scores for the correlated pair of factors across the three correlated dataset variants.

pervised experiments and datasets. We also further corroborate our finding that standard disentanglement metrics when evaluated on all FoV are insufficient to reveal this entanglement.

Does this matter for downstream tasks? Correlations between variables are of crucial importance in fairness applications, motivating an additional investigation on ramifications of these entangled latent spaces. In this setting we are interested in the unfairness of predicting one of the two correlated variables when the other represents a protected or sensitive attribute. In the following, we use a variant of demographic parity [96] that computes pairwise mutual information between latents and FoV [70]. In Figure 2.3 we evaluate this score when correlations are present across all unsupervised experiments and datasets. Unfairness tracks correlation strength in this scenario. Locatello et al. [70] show that representations learned without supervision may exhibit unfairness even without correlations. Our results suggest that the problem is substantially aggravated in the presence of correlations between a sensitive attribute and another ground truth FoV.

[96]: Dwork et al. (2012), 'Fairness through awareness'

[70]: Locatello et al. (2019), 'On the fairness of disentangled representations'

Summary. Factorization-based inductive biases are insufficient to learn disentangled representations from correlated observational data in the unsupervised case. We observed persisting pairwise entanglement in the latent space, both visually and quantitatively using appropriate metrics, which might be particularly problematic for fairness applications.

2.4.3 Generalization Under Changing Correlations

We now aim to understand the robustness and generalization capabilities of these models on data that is sampled not i.i.d. from the correlated train distribution but instead out-of-distribution (OOD) from a low-probability region given the correlation structure, e.g., large feet but short body height. In our Shapes3D experiments, this amounts to large object size and small azimuth which has *zero*

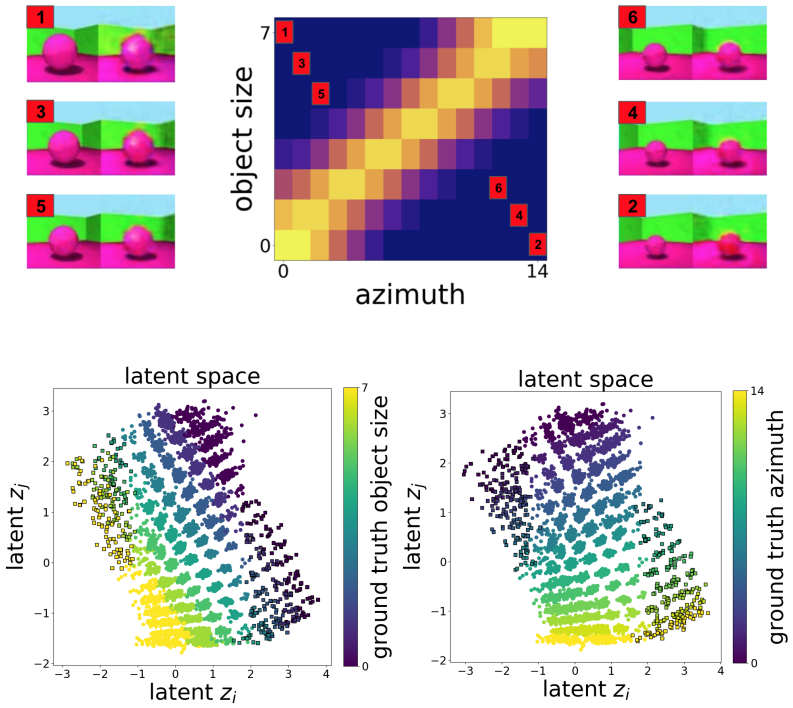


Figure 2.4: Generalization to out-of-distribution (OOD) test data. **Top:** Reconstructions of observations the model has never seen during training. **Bottom:** Latent space distribution of the two entangled dimensions. Circles without edges represent latent encodings of (correlated) training examples. Circles with edges are OOD examples that break the correlation pattern.

probability under the correlated train distribution. We focus on the model from Figure 2.2, which has disentangled all non-correlated factors well, allowing us to only focus on the two latent dimensions encoding the entangled variables.

First, we analyze examples with object size and azimuth values that have numerically zero probability under the correlated data distribution as shown as numbers 1 to 6 in Figure 2.4 (top panel) in the FoV space. The model can reconstruct these examples despite never having seen such configurations during training. This suggests that OOD examples are meaningfully encoded into the existing structure of the latent space *and* that the decoder is equally capable of generating observations from such unseen representations. Note that in contrast to Zhao et al. [97] who test this combinatorial generalization in the interpolation regime from uncorrelated, unbiased, but sparsely populated train data, to the best of our knowledge this extrapolation OOD generalization has not been studied on strongly correlated data so far. To test this hypothesis further, we analyzed latent traversals originating from these OOD points and observe that changes in the remaining factors reliably yield the expected reconstructions. Traversals with respect to the two entangled latent codes continue to encode object size and azimuth.

To fully understand these models' generalization properties we visualize the latent space encoding of the training distribution projected onto the two identified entangled dimensions (i.e., marginal-

izing over all others) in Figure 2.4 (bottom panel). We depict the ground truth value of each correlated variable via a color gradient. The two sets of depicted points are (1) latent codes for samples from the correlated training data (no edges) and (2) latent codes for samples with (object size, azimuth) configurations that have zero probability under the correlated training distribution (edges). As expected, contours of equal color (ground truth) are not aligned with the latent axes, indicating that the two latent dimensions encode both FoV at the same time. The generalization capabilities of this model away from the training data at least for the decoder can be described as follows: If we imagine the contour lines of equal color in both plots for the training data, i.e., only circles without edges, and linearly extend those lines, they form a grid in the latent space beyond where the training data lies. Latent encodings of unseen combinations of ground truth FoV happen to meaningfully extend and complete this grid structure in the latent space. Note that in this example, there is a natural ordering on both correlated FoV (numerical values for azimuth and size). Even though for categorical factors such as object color in datasets D and E we cannot expect the ad-hoc ordering of colors to be preserved in the latent space, the latent encodings still extrapolate meaningfully for examples with unseen combinations of ground truth FoV. In Appendix B.3.2, we show some of these characteristic latent space visualizations with similar extrapolation and generalization capabilities.

Summary. We conclude from these results that even though the models fail to disentangle the correlated FoV, they are still incorporating enough structure in the latent space to generalize towards unseen FoV configurations that are OOD w.r.t. the correlated train distribution.

2.5 Recovering Factorized Latent Codes

We now investigate the usefulness of semi- and weakly-supervised approaches to resolve latent entanglement.

Post-hoc alignment correction with few labels. When a limited number of FoV labels $\{\mathbf{c}_i\}_{i=1}^{M}$ can be accessed, a reasonable option for resolving entangled dimensions of the latent code is by *fast adaptation*. To identify the two entangled dimensions $\mathbf{z}_{fa} := (z_i, z_j)$ we look at the maximum feature importance for a given FoV from a GBT trained using these labels only. We then train a *substitution function* $f_\theta : \mathbb{R}^2 \to \{0, \dots, c_1^{\max}\} \times \{0, \dots, c_2^{\max}\}$ via supervised learning to infer the two ground truth FoV c_1, c_2 from the entangled latent codes z_i, z_j, $f_\theta(\mathbf{z}_{fa}) = (c_1, c_2)$. We then use this prediction to replace these two dimensions of the latent codes. Crucially, both

# Labels		0	100	1000	10000
Shapes3D (A) ($\sigma = 0.2$)	object size - azimuth	0.38	0.17	0.15	0.15
	median uncorr. pairs	0.09	0.08	0.07	0.07
Shapes3D (A) ($\sigma = 0.4$)	object size - azimuth	0.26	0.1	0.1	0.1
	median uncorr. pairs	0.09	0.08	0.08	0.08

Table 2.3: Fast adaption with few labels: Pairwise entanglement scores (averaged over 180 models each) for correlated FoV pair in Shapes3D (A). The correlated pair is highlighted (red). Zero labels reflects the unsupervised baseline. Fast adaption using linear regression can reduce these correlations with as little as 100 labels (blue).

steps of this procedure rely on the same M FoV labels, of which we assume only very few being available in practice.

Table 2.3 shows the pairwise entanglement score of the correlated FoV under this fast adaptation with a linear regression as the substitution function, which succeeds with as few as 100 labels (we only need the correlated factor labels), corresponding to less than 0.02% of all ground truth labels in Shapes3D. However, fast adaptation with linear regression substitution fails in some settings: when no two latent dimensions encode the applied correlation isolated from the other latent codes, or when the correlated variables do not have a unique natural ordering (e.g., categorical variables). Accordingly, a nonlinear substitution function such as an MLP can further reduce pairwise entanglement in these cases (see additional results in Appendix B.4.1).

Weak supervision mitigates learning latent entanglement. We now return to the weakly-supervised method from Section 2.2 and evaluate its applicability when training data is correlated. Specifically, we study the variant where the difference in the observation pair is present in one random ground truth factor. Each time we construct such a pair, all underlying ground truth factors share the same values except for one factor. Whenever this happens to be one of the correlated factors, the values of this factor within each pair are drawn from the marginal probability distribution conditioned on the other correlated factor. In these cases the difference in this factor is typically very small and depends on the correlation strength. Note that this procedure assures that constructed pairs are consistent with the observational data such that the correlation is never broken. In this part of our study, we consider datasets A, D, and E, and train β-VAEs with the same 6 hyperparameters and 5 random seeds as in the unsupervised study, yielding 360 models. Figure 2.5 summarizes the weak supervision results on Shapes3D (A) when imposing correlations in object size and azimuth. We consistently observe much better disentangled models, often achieving perfect DCI score irrespective of correlations in the dataset. The latent spaces tend to strongly align their coordinates with the ground truth label axis. Finally, weak-supervision reduces unfairness relative to the unsupervised

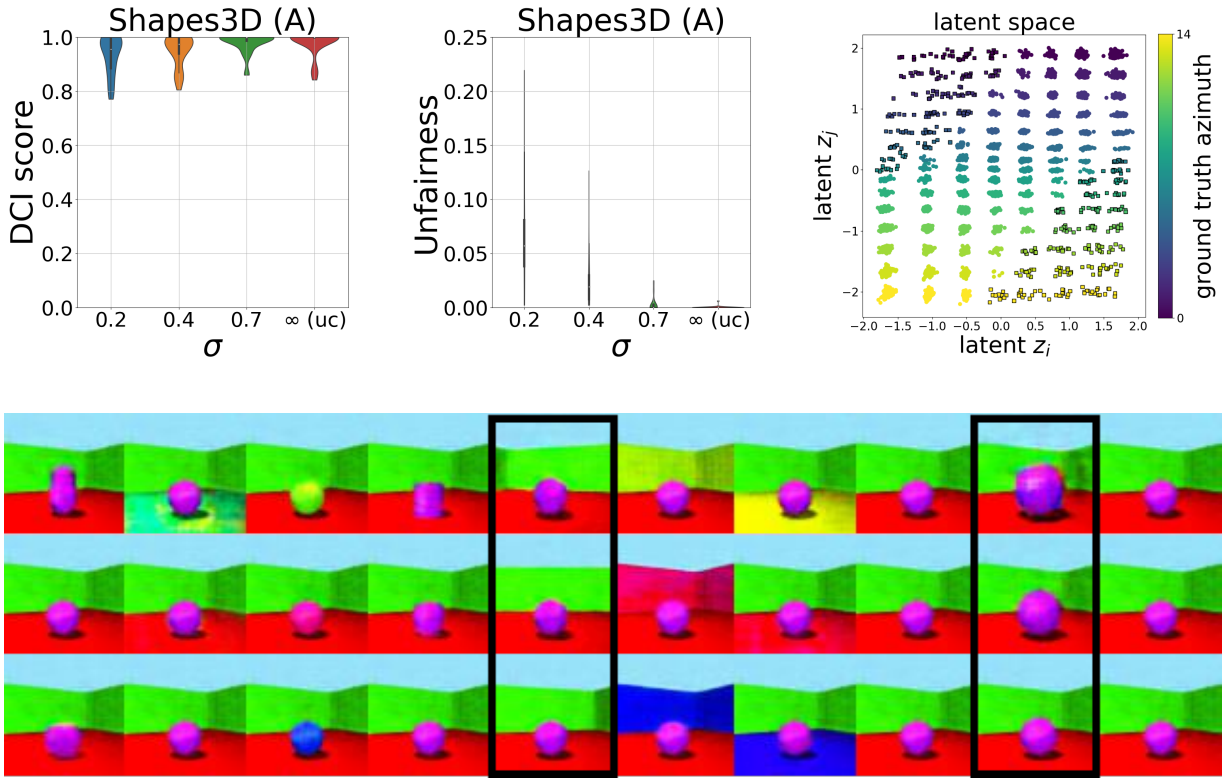


Figure 2.5: Top left: We show the disentanglement (DCI score) for models trained on Shapes3D *with weak supervision* when object size and azimuth are correlated with different strengths. Weak supervision helps to recover improved, often perfect, disentanglement. **Top middle:** Unfairness scores between correlated FoV are drastically reduced when using weak supervision (see scale) across all correlation strengths. **Top right:** Latent dimensions of the best DCI model with strong correlation ($\sigma = 0.2$) when using weak supervision. Representations are axis-aligned with respect to both of the correlated ground truth factors. **Lower:** Latent traversals of this model trained on strong correlation ($\sigma = 0.2$).

baseline from Figure 2.3, and occasionally even achieves zero unfairness score. Additional results on the other datasets can be found in Appendix B.4.2 including two additional training regimes in which the weak-supervision pairs are sampled from an interventional (but still correlated) distribution, representing an additional 1020 models. We consistently observe the same strong trends regarding disentangled correlations in all of the above studies using weak supervision.

These results suggest that weak supervision can provide a strong inductive bias capable of finding the right factorization and resolving spurious correlations for datasets of unknown degree of correlation. As a prominent example, this is an issue in the fairness context where real-world datasets often display unknown discriminatory correlations.

Summary. When ground truth information about the correlated FoV is available, our fast adaptation method can resolve latent correlations using only few labels of correlated FoV. When no labels are available, recently proposed approaches in weakly supervised disentanglement learning applied on correlated data offer an

alternative to overcoming pairwise correlations at train time. We have thus shown how to leverage even weak supervision signals to learn disentangled representations from correlated data.

2.6 Other Related Work

ICA and Disentanglement. Ideas related to disentangling the factors of variation date back to the non-linear ICA literature [98–102]. Recent work combines non-linear ICA with disentanglement [103, 104].

[98]: Comon (1994), 'Independent component analysis, a new concept?'
[99]: Jutten et al. (2003), 'Advances in nonlinear blind source separation'
[100]: Hyvärinen et al. (1999), 'Nonlinear independent component analysis: Existence and uniqueness results'
[101]: Hyvarinen et al. (2019), 'Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning'
[102]: Hyvarinen et al. (2016), 'Unsupervised feature extraction by time-contrastive learning and nonlinear ICA'
[103]: Khemakhem et al. (2020), 'Variational autoencoders and nonlinear ica: A unifying framework'
[104]: Sorrenson et al. (2020), 'Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN)'

Entangled and rotated latent spaces. Zhou et al. [105] measure disentanglement based on topological similarity of latent submanifolds, arguing that this may help uncover latent entanglements. While having orthogonal goals to ours, it complements our empirical findings regarding the problematic structure of entangled latent spaces. Rolinek, Zietlow, and Martius [56] contributed a related theoretical result showing that VAEs promote latent spaces pursued by (locally) orthogonal PCA embeddings due to the role of the diagonal covariance of the latent prior. They confirm this experimentally using the uncorrelated dSprites dataset. Stühmer, Turner, and Nowozin [106] use a family of rotationally asymmetric distributions as the latent prior, which can help learning disentangled subspaces. In contrast to the modeling perspective, we studied the effect of dependencies from the data perspective with strong correlations in the data generating prior $p^*(\mathbf{c})$.

Studies on correlated data. The literature so far is missing a systematic large-scale empirical study how popular inductive biases such as factorized priors behave when actually learning from correlated datasets, although several smaller experiments along these lines can be acknowledged. Chen et al. [60] studied correlated 3DFaces [107] by fixing all except three factors in which the authors conclude that the β-TC-VAE regularizer can help to disentangle imposed correlations based on the MIG metric. However, the latent structure was not studied in detail; our findings suggest that global disentanglement metrics are insufficient to fully diagnose models learned from correlated data. Based on the observation that the assumption of independent factors is unlikely to hold in practice and certain factors might be correlated in data Li et al. [94] propose methods based on a pairwise independence assumption instead. Brekelmans et al. [108] show that Echo noise results in superior disentanglement compared to standard β-VAE in a small experiment on a downsampled dSprites variant. Creager et al. [83] based some of the evaluations of a proposed new autoencoder architecture in the fairness context on a biased dSprites variant and Yang et al. [109] study a linear SCM in a VAE architecture on

[107]: Paysan et al. (2009), 'A 3D face model for pose and illumination invariant face recognition'

datasets with dependent variables. However, their studies focused on representation learners that require strong supervision via FoV labels at train time.

2.7 Conclusion

We have taken first steps to understand the gap between idealized datasets and realistic applications of modern disentanglement learners by presenting the first large-scale empirical study examining how such models cope with correlated observational data. We find that existing methods fail to learn disentangled representations for strongly correlated factors of variation. We discuss and quantify practical implications for downstream tasks like fair decision making. Despite these shortcomings, the learned latent space structure of some models naturally accommodates unseen examples via meaningful extrapolation, leading to out-of-distribution generalization capabilities. Based on these findings, we ultimately demonstrate how to correct for latent entanglement via fast adaptation and other weakly supervised training methods. Future work is needed to address open question surrounding whether these results extend to high resolution inputs [110] and their impact on downstream tasks in the real world [2, 4]. Finally, our findings draw additional attention towards how inductive biases and weak supervision can be combined for successful disentanglement and under what circumstances this leads to strong out-of-distribution generalization.

[110]: Miladinović et al. (2021), 'Spatial Dependency Networks: Neural Layers for Improved Generative Image Modeling'

[2]: **Ahmed** et al. (2021), 'Causalworld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning'

[4]: **Träuble** et al. (2022), 'The Role of Pretrained Representations for the OOD Generalization of RL Agents'

A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning

3

3.1 Introduction 33
3.2 CausalWorld Benchmark 35
3.3 Related Work 40
3.4 Experiments 41
3.5 Conclusion 44

Abstract

Despite recent successes of reinforcement learning (RL), it remains a challenge for agents to transfer learned skills to related environments. To facilitate research addressing this, we propose *CausalWorld*, a benchmark for causal structure and transfer learning in a robotic manipulation environment. The environment is a simulation of an open-source robotic platform, hence offering the possibility of sim-to-real transfer. Tasks consist of constructing 3D shapes from a given set of blocks - inspired by how children learn to build complex structures. The key strength of *CausalWorld* is that it provides a combinatorial family of such tasks with common causal structure and underlying factors (including, e.g., robot and object masses, colors, sizes). The user (or the agent) may intervene on all causal variables, which allows for fine-grained control over how similar different tasks (or task distributions) are. One can thus easily define training and evaluation distributions of a desired difficulty level, targeting a specific form of generalization (e.g., only changes in appearance or object mass). Further, this common parametrization facilitates defining curricula by interpolating between an initial and a target task. While users may define their own task distributions, we present eight meaningful distributions as concrete benchmarks, ranging from simple to very challenging, all of which require long-horizon planning and precise low-level motor control at the same time. Finally, we provide baseline results for a subset of these tasks on distinct training curricula and corresponding evaluation protocols, verifying the feasibility of the tasks in this benchmark.

About This Chapter

This chapter has been adapted from **Ahmed** et al. [2].

CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning

Ossama Ahmed*, **Frederik Träuble***, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf and Stefan Bauer.

ICLR 2021 International Conference for Learning Representations.

OA and FT were shared first authors. Asterisk denotes equal contribution.

Author Contributions based on CRediT (Contributor Roles Taxonomy)

Ossama Ahmed: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization **Frederik Träuble**: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision **Anirudh Goyal**: Conceptualization, Writing - Original Draft, Writing - Review & Editing, Supervision **Alexander Neitz**: Conceptualization, Software, Investigation, Writing - Review & Editing **Manuel Wüthrich**: Conceptualization, Writing - Review & Editing, Supervision **Yoshua Bengio**: Writing - Review & Editing, Supervision **Bernhard Schölkopf**: Conceptualization, Resources, Writing - Review & Editing, Supervision, Funding acquisition **Stefan Bauer**: Conceptualization, Investigation, Writing - Review & Editing, Supervision, Project administration

3.1 Introduction

Benchmarks have played a crucial role in advancing entire research fields, for instance computer vision with the introduction of CIFAR-10 and ImageNet [27, 111]. When it comes to the field of reinforcement learning (RL), similar breakthroughs have been achieved in domains such as game playing [34, 35], learning motor control for high-dimensional simulated robots [112], multi-agent settings [113, 114] and for studying transfer in the context of meta-learning [115]. Nevertheless, trained agents often fail to transfer the knowledge about the learned skills from a training environment to a different but related environment sharing part of the underlying structure. This can be attributed to the fact that it is quite common to evaluate an agent on the training environments themselves, which leads to overfitting on these narrowly defined environments [116], or that algorithms are compared using highly engineered and biased reward functions which may result in learning suboptimal policies with respect to the desired behaviour; this is particularly evident in robotics.

In existing benchmarks [115, 117–120] the amount of shared causal structure between the different environments is mostly unknown. For instance, in the Atari Arcade Learning environments, it is unclear how to quantify the underlying similarities between different Atari games and we generally do not know to which degree an agent can be expected to generalize.

To overcome these limitations, we introduce a novel benchmark in a robotic manipulation environment that we call `CausalWorld`. It features a diverse set of environments that, in contrast to previous designs, share a large set of parameters and parts of the causal structure. Being able to intervene on these parameters (individually or collectively) permits the experimenter to evaluate agents' generalization abilities with respect to different types and magnitudes of changes in the environment. These parameters can be varied gradually, which yields a continuum of similar environments. This allows for fine-grained control of training and test distributions and the design of learning curricula.

A remarkable skill that humans learn to master early on in their life is building complex structures using spatial-reasoning and dexterous manipulation abilities [121–123]. Playing with toy blocks constitutes a natural environment for children to develop important visual-spatial skills, helping them 'generalize' in building complex composition designs from presented or imagined goal structures [124–127]. Inspired by this, `CausalWorld` is designed to aid in learning and investigating these skills in a simulated

[27]: Krizhevsky et al. (2012), 'Imagenet classification with deep convolutional neural networks'
[111]: Krizhevsky et al. (2009), 'Learning multiple layers of features from tiny images'
[34]: Mnih et al. (2020), 'Playing atari with deep reinforcement learning'
[35]: Silver et al. (2017), 'Mastering the game of go without human knowledge'
[112]: Akkaya et al. (2019), 'Solving rubik's cube with a robot hand'
[113]: Baker et al. (2019), 'Emergent tool use from multi-agent autocurricula'
[114]: Berner et al. (2019), 'Dota 2 with large scale deep reinforcement learning'
[115]: Yu et al. (2020), 'Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning'
[116]: Whiteson et al. (2011), 'Protecting against evaluation overfitting in empirical reinforcement learning'
[115]: Yu et al. (2020), 'Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning'
[117]: Goyal et al. (2019), 'Infobot: Transfer and exploration via the information bottleneck'
[118]: Cobbe et al. (2019), 'Quantifying generalization in reinforcement learning'
[119]: Bellemare et al. (2013), 'The arcade learning environment: An evaluation platform for general agents'
[120]: James et al. (2020), 'Rlbench: The robot learning benchmark & learning environment'
[121]: Casey et al. (2008), 'The development of spatial skills through interventions involving block building activities'
[122]: Caldera et al. (1999), 'Children's play preferences, construction play with blocks, and visual-spatial skills: Are they related?'
[123]: Kamii et al. (2004), 'The development of logico-mathematical knowledge in a block-building activity at ages 1–4'

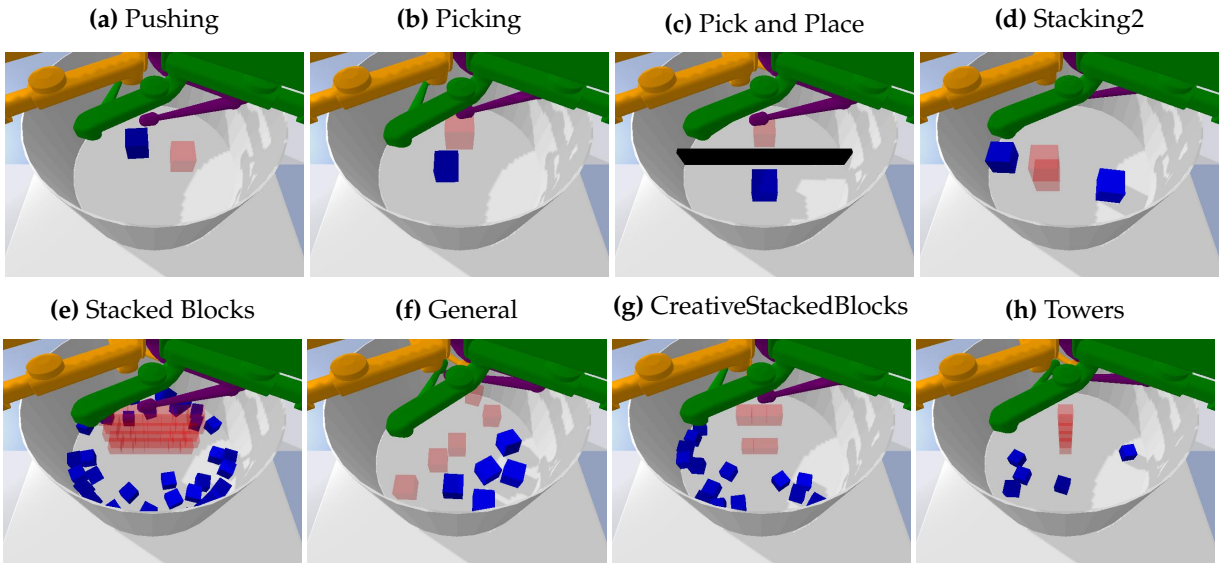


Figure 3.1: Example tasks from the task generators provided in the benchmark. The goal shape is visualized in opaque red and the blocks in blue.

[124]: Verdine et al. (2017), *Links between spatial and mathematical skills across the preschool years*
[125]: Nath et al. (2014), 'Construction play and cognitive skills associated with the development of mathematical abilities in 7-year-old children'
[126]: Dewar (2018), 'The benefits of toy blocks: The science of construction play'
[127]: Richardson et al. (2014), 'Children's construction task performance and spatial ability: Controlling task complexity and predicting mathematics performance'

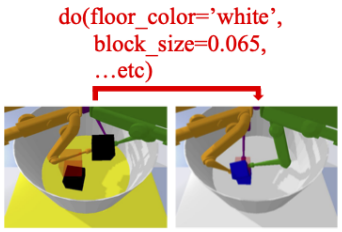


Figure 3.2: Example of do-interventions on exposed variables in `CausalWorld`.

[118]: Cobbe et al. (2019), 'Quantifying generalization in reinforcement learning'
[129]: Chevalier-Boisvert et al. (2018), 'BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning'

robotic manipulation environment corresponding to the open-source TriFinger robot platform from Wüthrich et al. [128], which can be built in the real world. Tasks are formulated as building 3D goal shapes using a set of available blocks by manipulating them - as seen in Figure 3.1 and Figure 3.2. This yields a diverse familiy of tasks, ranging from relatively simple (e.g. pushing a single object) to extremely hard (e.g. building a complex structure from a large number of objects).

`CausalWorld` improves upon previous benchmarks by exposing a large set of parameters in the causal generative model of the environments, such as weight, shape and appearance of the building blocks and the robot itself. The possibility of intervening on any of these properties at any point in time allows one to set up training curricula or to evaluate an agent's generalization capability with respect to different parameters. Furthermore, in contrast to previous benchmarks [118, 129], researchers may build their own real-world platform of this simulator at low cost, as detailed in [128], and transfer their trained policies to the real world.

Finally, by releasing this benchmark we hope to facilitate research in causal structure learning, i.e. learning the causal graph (or certain aspects of it) as we operate in a complex real-world environment whose dynamics follow the laws of physics, which induce causal relations between the variables. Changes to the variables we expose can be considered do-interventions on the underlying structural causal model (SCM). Consequently, we believe that this benchmark offers an exciting opportunity to investigate causality and its connection to RL and robotics.

Table 3.1: Comparison of Causal World with RLBench [120], MetaWorld [115], IKEA [130], BabyAI [129], CoinRun [118], AtariArcade [119] and MuJoBan [131]

Benchmark	RLBench	MetaWorld	IKEA	MuJoBan	BabyAI	CoinRun	AtariArcade	`CausalWorld`
do-interventions interface	✗	✗	✗	✗	✗	✗	✗	✓
procedurally generated environments	✗	✗	✗	✗	✓	✓	✗	✓
online distribution of tasks	✗	✗	✗	✗	✗	✗	✗	✓
setup custom curricula	✗	✗	✗	✓	✗	✗	✗	✓
disentangle generalization ability	✗	✗	✗	✗	✗	✗	✗	✓
real-world similarity	✓	✓	✓	✓	✗	✗	✗	✓
open-source robot	✗	✗	✗	✗	✗	✗	✗	✓
low-level motor control	✓	✓	✓	✓	✗	✗	✗	✓
long-term planning	✗	✗	✓	✓	✓	✗	✓/✗	✓
unified success metric	✗	✗	✓	✓	✓	✓	✓	✓

Our main contributions can be summarized as follows:

- We propose `CausalWorld`, a new benchmark comprising a parametrized family of robotic manipulation environments for advancing out-of-distribution generalization and causal structure learning in RL *.
- We provide a systematic way of defining curricula and disentangling generalization abilities of RL agents with respect to different changes in the environment, since we allow for do-interventions to be performed on different environment variables (parameters and states) individually.
- We establish baseline results for some of the available tasks under different learning algorithms, thus verifying the feasibility of the tasks.
- We show how different learning curricula affect generalization across different axes by reporting some of the in-distribution and out-of-distribution generalization capabilities of the trained agents.

3.2 CausalWorld Benchmark

Here we make the desiderata outlined in the introduction more precise:

1. The set of environments should be sufficiently diverse to allow for the design of challenging transfer tasks.
2. We need to be able to intervene on different properties (e.g. masses, colors) individually, such that we can investigate different types of generalization.
3. It should be possible to convert any environment to any other environment by gradually changing its properties through

* `CausalWorld` can be installed via pip with more details under: Project page

interventions; this requirement is important for evaluating different levels of transfer and for defining curricula.
4. The environments should share some causal structure to allow algorithms to transfer the learned causal knowledge from one environment to another.
5. There should be a unified measure of success, such that an objective comparison can be made between different learning algorithms.
6. The benchmark should make it easy for users to define meaningful distributions of environments for training and evaluation. In particular, it should facilitate evaluation of in-distribution and out-of-distribution performance.
7. The simulated benchmark should have a real-world counterpart to allow for sim-to-real.

In light of these desiderata, we propose a setup in which a robot must build goal shapes using a set of available objects. It is worth noting that similar setups were proposed previously in a less realistic setting, e.g. in [112, 132–137]. Specifically, a task is formulated as follows: *given a set of available objects, the agent needs to build a specific goal structure*, see Figure 3.2 for an example. The vast amount of possible target shapes and environment properties (e.g. mass, shape and appearance of objects and the robot itself) makes this a diverse and challenging setting to evaluate different generalization aspects. `CausalWorld` is a simulated version (using the Bullet physics engine [138]) of the open-source TriFinger robot platform from Wüthrich et al. [128]. Each environment is defined by a set of variables such, as gravity, floor friction, stage color, floor color, joint positions, various block parameters (e.g. size, color, mass, position, orientation), link colors, link masses and the goal shape. See Table C.1 in the Appendix for a more extensive list of these variables.

Desideratum 1 is satisfied since different environment properties and goal shapes give rise to very different tasks, ranging from relatively easy (e.g. re-positioning a single cube) to extremely difficult (e.g. building a complex structure). Desideratum 2 is satisfied because we allow for arbitrary interventions on these properties, hence users or agents may change parameters individually or jointly. Desideratum 3 is satisfied because the parameters can be changed gradually. Desideratum 4 is satisfied because all the environments share the causal structure of the robot, and one may also use subsets of environments which share even more causal structure. We satisfy desideratum 5 by defining the measure of success for all environments as the volumetric overlap of the goal shape with available objects. Further, by splitting the set of parameters into a set A, intended for training and in-distribution evaluation, and a set B, intended for out-of-distribution evaluation, we satisfy

[112]: Akkaya et al. (2019), 'Solving rubik's cube with a robot hand'
[132]: Janner et al. (2019), 'Reasoning About Physical Interactions with Object-Oriented Prediction and Planning'
[133]: Bapst et al. (2019), 'Structured agents for physical construction'
[134]: McCarthy et al. (2020), 'Learning to build physical structures better over time.'
[135]: Fahlman (1974), 'A planning system for robot construction tasks'
[136]: Winston (1972), 'Learning Structual Descriptions from Examples'
[137]: Winograd (1972), 'Understanding natural language'
[138]: Coumans et al. (2013), 'Bullet real-time physics simulation'

desideratum 6. Finally, since the TriFinger robot [128] can be built in the real-world, we satisfy desideratum 7. Desideratum 7 and 2 are in partial conflict since sim-to-real is only possible for the tasks which are constrained to the variables on which the robot can physically act upon.

[128]: Wüthrich et al. (2021), 'TriFinger: An Open-Source Robot for Learning Dexterity'

Task generators: To generate meaningful families of similar goal shapes, `CausalWorld` allows for defining task generators which can generate a variety of different goal shapes in an environment. For instance, one task generator may generate pushing tasks, while another one may generate tower-building tasks (see Figure 3.1). Each task generator is initialized with a default goal shape from its corresponding family and comes with a sampler to sample new goal shapes from the same family. Additionally, upon construction, one can specify the environments' initial state and initial goal shape structure when deviating from the default. The maximum episode time to build a given shape is `number_of_blocks` $\times$ 10 seconds. `CausalWorld` comes with eight pre-defined task generators (see Figure 3.1).

- Three generators create goal shapes with a single block: *Pushing* with the goal shape on the floor, *Picking* having the goal shape defined above the floor and *Pick and Place* where a fixed obstacle is placed between the initial block and goal pose.
- *Stacking2* involves a goal shape of two stacked blocks, which can also be considered one instance of the *Towers* generator.
- The remaining generators use a variable number of blocks to generate much more complex and challenging target shapes, as detailed in the appendix: *Towers*, *Stacked Blocks*, *Creative Stacked Blocks* and *General*.

Given that building new environments using current physics simulators is often tedious, we provide a simple API for users who wish to create task generators for new challenging shape families, which may be added to `CausalWorld's` task generators repository.

Action and Observation Spaces: The robot can be chosen to operate in either joint position control mode, joint torque control mode, end-effector position control mode, or the delta of each. In any of these cases, the action is 9-dimensional (one per joint). We provide two observation modes: *structured* and *pixel*. In the *structured* mode, the observation vector is constructed using a rule for the ordering of the relevant variables, such as joint positions, joint velocities, block positions, etc. Thus, the size of the observation space depends on the number of blocks, which could potentially

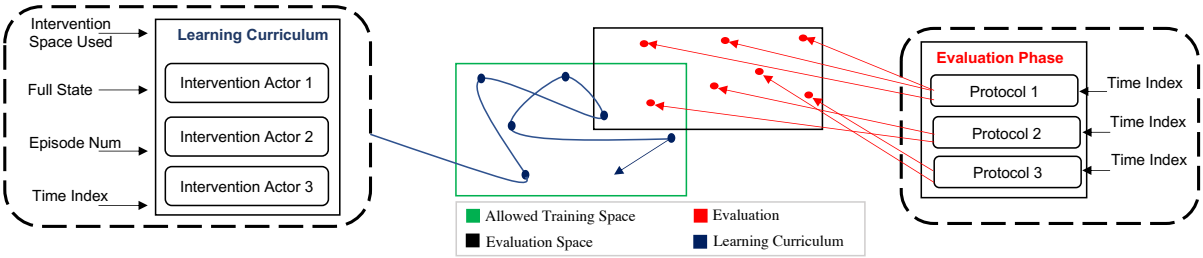


Figure 3.3: Key components for generic training and evaluation of RL agents. Left: A learning curriculum which is composed of various intervention actors that decide which variables to intervene on (for a valid intervention, values need to be in the allowed training space (ATS)). Right: Evaluation protocols are shown which may intervene on variables at episode resets or within episodes (for a valid intervention, values need to be in the evaluation space (ES)). Middle: we represent the ATS and ES, where each intervention results in one point in the spaces. As shown ATS and ES may intersect, eg. if the protocols are meant to evaluate in-distribution generalization.

change with every new goal sampled, e.g. in Towers, (Creative) Stacked Blocks and General. In contrast, in the *pixel* mode, the agent receives six RGB images (hence the dimension of the observation is $6 \times 3 \times 128 \times 128$), the first three images are rendered from the three cameras mounted around the TriFinger robot, and the last three images specify the goal image of the target shape rendered from the same cameras. Additionally, `CausalWorld` allows users to set up a fully customized observation space.

Rewards: The reward function r is defined as the fractional volumetric overlap of the blocks with the goal shape, which ranges between 0 (no overlap) and 1 (complete overlap). Since this reward function is shared across all tasks, an agent that learned r from some training tasks could in principle use it to solve unseen goal structures. There is also the possibility of modifying the reward function to 1) sparsify the reward further by returning a binary reward signal instead, or 2) add a dense reward function in order to introduce inductive biases via domain knowledge and solution guidance. We hope that the considerable complexity and diversity of goal shapes motivate and accelerate the development of algorithms that are not dependent on hand-tuned reward functions.

Training and evaluation spaces: In this benchmark, a learning setting consists of an allowed training space (ATS) and an evaluation space (ES), both of which are subspaces of the full parameter space. During training, in the simplest setting, parameters are sampled i.i.d. from the ATS. However, unlike existing benchmarks, `CausalWorld` allows in addition for curricula within the ATS as well as settings where the agent itself intervenes on the parameters within an episode (see Figure 3.3). Similarly, during evaluation, parameters may be sampled i.i.d. from the evaluation space at each episode reset, or there can be interventions within an episode. Moreover, in order to retrieve the setting considered in most RL

benchmarks, we could set the ATS and the ES to be identical and intervene only on object and robot states (and keep other environment properties constant) at each episode reset. However, to evaluate out-of-distribution generalization, one should set the two spaces (ATS and ES) to be different; possibly even disjoint. Additionally, to evaluate robustness with respect to a specific parameter (e.g. object mass), one may define the training and evaluation spaces which only differ in that particular parameter. In order to facilitate the definition of appropriate training and evaluation settings, we pre-define two disjoint sets, $\mathbf{A}_i$ and $\mathbf{B}_i$, for each parameter i. Through this, one can for instance define the training space to be $\mathbf{A}_1 \times \mathbf{A}_2 \times ...$ and the evaluation space to be $\mathbf{B}_1 \times \mathbf{B}_2 \times ...$ to assess generalization with respect to all parameters simultaneously. Alternatively, the evaluation space could be defined as $\mathbf{A}_1 \times \mathbf{A}_2 \times ... \times \mathbf{B}_i \times \mathbf{A}_{i+1} \times ...$ to assess generalization with respect to parameter i only. Lastly, users may also define their own spaces which could then be integrated into the benchmark to give rise to new learning settings.

Intervention actors: To provide a convenient way of specifying learning curricula, we introduce intervention actors. At each time step, such an actor takes all the exposed variables of the environment as inputs and may intervene on them. To encourage modularity, one may combine multiple actors in a learning curriculum. This actor is defined by the episode number to start intervening, the episode number to stop intervening, the timestep within the episode it should intervene and the episode periodicity of interventions. We provide a set of predefined intervention actors, including an actor which samples parameters randomly at each episode reset, which corresponds to domain-randomization. It is also easy to define custom intervention actors, we hope that this facilitates investigation into optimal learning curricula (see Figure 3.3).

Probing the Causal Structure in RL: The problem setting in RL is usually formulated using the language of Markov Decision Processes (MDPs) or Partially Observable Markov Decision Processes (POMDPs) [139], but can be also represented by Structural Causal Models (SCMs), as shown in [140]. We refer to Appendix C.6 in the Appendix for a detailed explanation. This is achieved by formulating all conditional probability distributions as deterministic functions that take independent noise variables as inputs. These independent noise variables can specify different scenarios while the deterministic functions reflect the causal mechanisms of the system [46]. Changes in the environment can stem from two different sources:

[139]: Sutton et al. (1999), 'Reinforcement learning'

[140]: Buesing et al. (2018), 'Woulda, coulda, shoulda: Counterfactually-guided policy search'

[46]: Schölkopf et al. (2021), 'Toward causal representation learning'

1. The agent may alter the state of the environment (e.g. the position of a block) indirectly, through its actions (e.g. pushing the block by applying appropriate torques at the motors).
2. During the execution of a learning curriculum or an evaluation protocol, we may directly intervene on any variable of the SCM, including all the latent variables of the causal model that are not accessible to the RL agent (such as gravity, object mass or color).

The first source is the default type of admissible interventions in RL benchmarks, whereas CausalWorld allows for interventions of the second type in addition. The idea is that interventions on these latent variables, e.g. during a learning curriculum, will allow the agent to distinguish between spurious correlations that are only present in a particular setting and true causal relations that will hold across all settings (i.e. interventions). If the agent is able to learn such a representation of the underlying SCM structure, we would expect it to perform well even in out-of-distribution scenarios [3, 46] because the causal structure remains the same, even when the functional form of certain relations may vary (e.g. when transferring to the real robot). Moreover, we hope that by having access to a broad range of interventions in CausalWorld it will aid the inference of the underlying SCM structure through different causal discovery methods.

3.3 Related Work

Previous benchmarks for RL mostly focused on the single task learning setting such as OpenAI Gym and DM control suite [141, 142]. In contrast, a recent line of work, e.g. Meta-World and RLBench [115, 120], aims at studying multi-task learning as well as meta-learning. Such benchmarks mostly provide non-parametric, hand-designed task variations, it is hence unclear how much structure is shared between them. For instance, it is not clear how different it is to “open a door" compared to “opening a drawer". To address the ambiguity in the shared structure between the tasks, `CausalWorld` was designed to allow interventions to be performed on many environment variables giving rise to a large space of tasks with well-defined relations between them, which we believe is a missing key component to address generalization in RL. A detailed comparison between CausalWorld and similar benchmarks is shown in Table 3.1.

Similar parametric formulations of different environments were used in the generalization-for-RL literature, which has played an important role in advancing the field [129, 143–148]. In these previ-

[3]: **Dittadi** et al. (2021), ‘On the Transfer of Disentangled Representations in Realistic Settings’
[46]: Schölkopf et al. (2021), ‘Toward causal representation learning’
[141]: Tassa et al. (2018), ‘Deepmind control suite’
[142]: Brockman et al. (2016), ‘Openai gym’
[115]: Yu et al. (2020), ‘Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning’
[120]: James et al. (2020), ‘Rlbench: The robot learning benchmark & learning environment’
[129]: Chevalier-Boisvert et al. (2018), ‘BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning’
[143]: Packer et al. (2018), ‘Assessing generalization in deep reinforcement learning’
[144]: Rajeswaran et al. (2017), ‘Towards generalization and simplicity in continuous control’
[145]: Pinto et al. (2017), ‘Robust adversarial reinforcement learning’
[146]: Yu et al. (2017), ‘Preparing for the unknown: Learning a universal policy with online system identification’
[147]: Henderson et al. (2017), ‘Benchmark environments for multitask learning in continuous domains’
[148]: Dulac-Arnold et al. (2020), ‘An empirical investigation of the challenges of real-world reinforcement learning’

ous works, variables were mostly changed randomly, as opposed to the full control over the variables provided by `CausalWorld`.

Another important open problem for the RL community is the standardization of the reported learning curves and results. RL methods have been shown to be sensitive to a range of different factors [149]. Thus it is crucial to devise a set of metrics that measure reliability of RL algorithms and ensure their reproducibility. Chan et al. [150] distinguishes between several evaluation modes like "evaluation during training" and "evaluation after learning". Osband et al. [151] recently proposed a benchmarking suite that disentangles the ability of an algorithm to deal with different types of challenges. Its main components are: enforcing a specific methodology for an agent's evaluation beyond the environment definition and isolating core capabilities with targeted 'unit tests' rather than integrating the general learning ability.

[149]: Henderson et al. (2018), 'Deep reinforcement learning that matters'

Moreover, causality has been historically studied from the perspective of probabilistic and causal reasoning [45], cognitive psychology [152], and more recently in the context of machine learning [46, 78, 153, 154]. On the contrary, we believe its link to robotics is not yet drawn systematically. To bridge this gap, one of the main motivations of `CausalWorld` was to facilitate research in causal learning for robotics, such as observational discovery of causal effects in physical reality, counterfactual reasoning, and causal structure learning.

[45]: Pearl (2009), *Causality*

[152]: Griffiths et al. (2005), 'Structure and strength in causal induction'

[46]: Schölkopf et al. (2021), 'Toward causal representation learning'

[78]: Goyal et al. (2021), 'Recurrent independent mechanisms'

[153]: Baradel et al. (2019), 'CoPhy: Counterfactual Learning of Physical Dynamics'

[154]: Bakhtin et al. (2019), 'Phyre: A new benchmark for physical reasoning'

3.4 Experiments

To illustrate the usage of this benchmark and to verify the feasibility of some basic tasks, we evaluate current state-of-the-art model-free reinforcment learning algorithms (MF-RL) on a subset of the goal shape families described in Section 3.2 and depicted in Figure 3.1: (a) Pushing, (b) Picking, (c) Pick and Place, and (d) Stacking2. These goal shapes reflect basic skills that are required to solve more complex construction tasks.

Setup: We attempt to investigate how well an agent will perform on different evaluation distributions, depending on the curriculum it has been trained with. We train each method under the following curricula:

- Curriculum 0: no environment changes; each episode is initialized from the default task lying in space **A** - note that here the initial state never changes (i.e. no interventions).

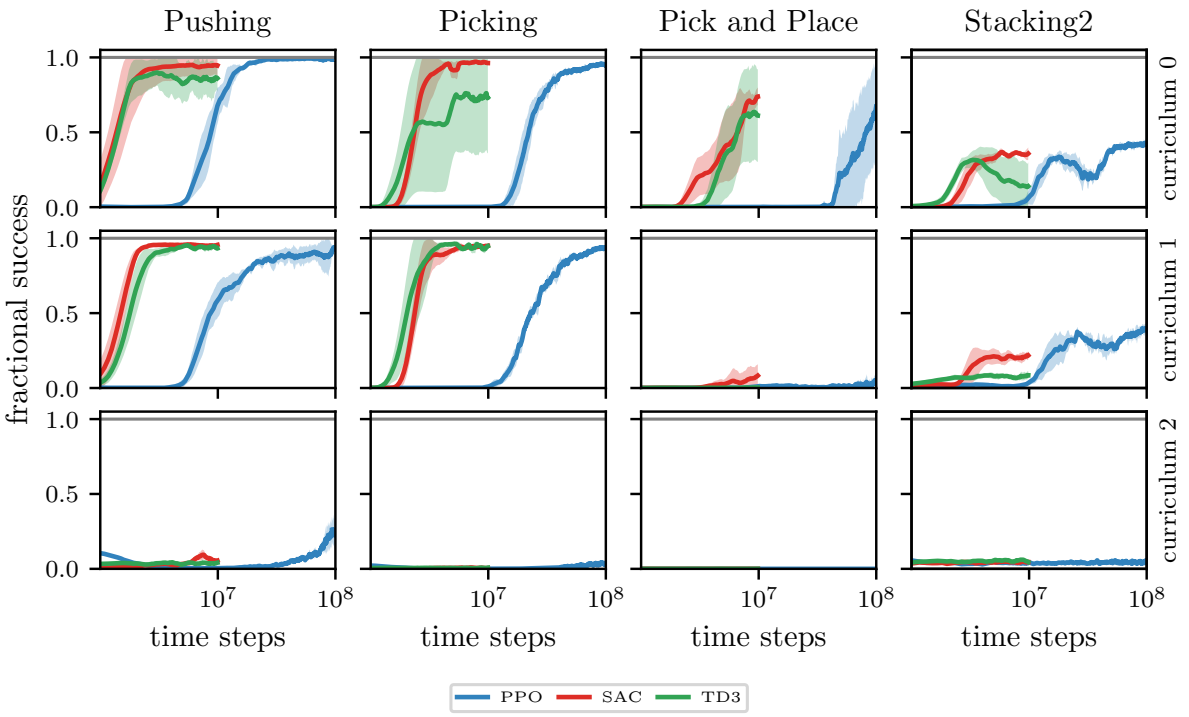
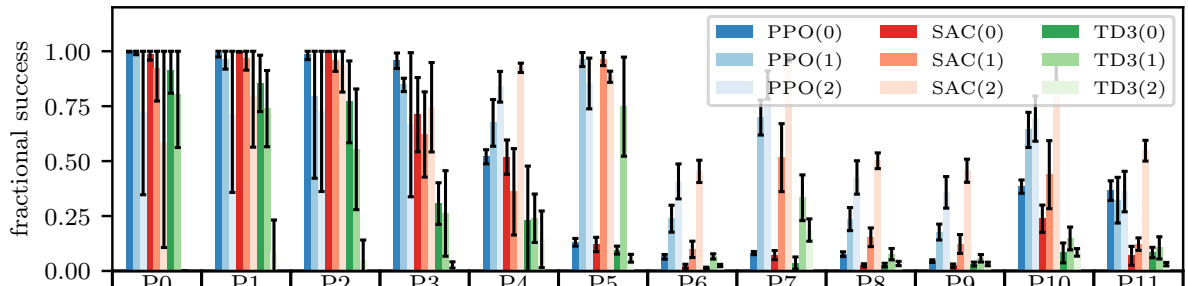


Figure 3.4: Fractional success curves averaged over five random seeds for the tasks and learning algorithms specified above, under three different training curricula: (0) no curriculum, (1) goal position and orientation randomization in space **A** every episode and (2) a curriculum where we intervene on all variables in space **A** simultaneously every episode.

- Curriculum 1: goal shape randomization; at the beginning of each episode a new goal shape is sampled from space **A** (i.e. interventions on goal position and orientation).
- Curriculum 2: full randomization w.r.t. the task variables[1]; every episode a simultaneous intervention on all variables is sampled from space **A** (i.e. can be seen as equivalent to extreme domain randomization in one space).

1: Note that each task generator can suppress interventions that would yield goal shapes outside its family.

The curriculum will, as expected, affect the generalization capabilities of the trained agents. With `CausalWorld`'s formulation, these generalization capabilities can easily be disentangled and benchmarked quantitatively, as explained in Section 3.2. For each of the goal shape families (Figure 3.1 a-d), we train agents under the three described curricula using the following MF-RL algorithms: Proximal Policy Optimization (PPO) from Schulman et al. [155], Soft Actor-Critic (SAC) from Haarnoja et al. [40] and the Twin Delayed DDPG (TD3) from Fujimoto, Hoof, and Meger [156]. We provided these methods with a hand-designed dense reward function as we did not observe any success with the sparse reward only. Each of the mentioned setups is trained for five different random seeds, resulting in 180 trained agents.



	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
space	-	A	B	A	A	A	B	A	B	B	A	B
variables	-	bm	bm	bs	bp	gp	bp, gp	bp, gp, bm	bp, gp, bm	bp, gp, bm, ff	all	all
time step	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.5: Evaluation scores for pushing baselines. Each protocol was evaluated for 200 episodes and each bar is averaged over five models with different random seeds. The variables listed under each protocol are sampled from the specified space at the start of every episode while all other variables remain fixed [bp block pose, bm block mass, bs block size, gp goal pose, ff floor friction].

Training model-free RL methods: We report the training curves averaged over the random seeds in Figure 3.4. As can be seen from these fractional success training curves, MF-RL methods are capable of solving the single block goal shapes (pushing, picking, pick and place) seen during training time given enough experience. However, we observe that none of the methods studied here managed to solve stacking two blocks. The score below 0.5 indicates that it only learns to push the lower cube into the goal shape. This shows that multi-object target shapes can become nontrivial quickly and that there is a need for methods making use of the modular structure of object-based environments. To no surprise, the training curriculum has a major effect on learning. For example, methods rarely manage to pick up any significant success signal under extreme domain randomization as in curriculum 2, even after 100 million timesteps. Note that these curves represent the scores under the conditions of the training environments. Next, we will discuss shared evaluation protocols that allow to benchmark and compare agents trained under different conditions.

Benchmarking generalization capabilities along various axes: For each of the four goal shape families, we define a set of 12 evaluation protocols that we consider meaningful and representative for benchmarking the different algorithms. In the protocols presented here, we sample the values from a protocol-specific set of variables at the start of each episode while keeping all other variables fixed to their default values. We evaluate each agent on 200 episodes by computing the fractional success score at the last time step of each episode and reporting the mean. These evaluation protocols allow to disentangle generalization abilities, as they show robustness with respect to different types of interventions, see Figure 3.5. The following are some of the observations we made for pushing:

- Agents that were trained on the default pushing task environment (curriculum 0) do well (as expected) on the default task (P0). Interestingly, we likewise see a generalization capability to initial poses from variable space A (P4). This can be explained by a substantial exploration of the block positions via manipulation during training. In contrast, we see that the agents exhibit weaknesses regarding goal poses (P5), they seem to overfit to their training settings in this case.
- For agents trained with goal pose randomization (curriculum 1) we see similar results as with curriculum 0, with the difference that agents under this curriculum generalize robustly to different goal poses (P5), as one would expect.
- Finally, agents that experience extreme domain randomization (curriculum 2) at training time, fail to learn any relevant skill as shown by the flat training curve in Figure 3.4. An explanation for this behavior could be that the agent might need more data and optimization steps to handle this much more challenging setting. Another possibility is that it may simply not be possible to find a strategy which simultaneously works for all parameters (note that the agent does not have access to the randomized parameters and hence must be robust to them). This poses an interesting question for future work.

As expected, we observe that an agent's generalization capabilities are related to the experience gathered under its training curriculum. `CausalWorld` allows us to explore this relationship in a differentiated manner, assessing which curricula lead to which generalization abilities. This will not only help uncover an agent's shortcomings but may also aid in investigating novel learning curricula and approaches for robustness in RL. Corresponding evaluations regarding the tasks Picking, Pick and Place and Stacking2 can be found in Appendix C.5. Lastly, we note that this benchmark comprises extremely challenging tasks that appear to be out of reach of current model-free methods without any additional inductive bias.

3.5 Conclusion

We have introduced a new benchmark - `CausalWorld` - to facilitate research in causal structure and transfer learning using a simulated environment of an open-source robot, where learned skills could potentially be transferred to the real world. We showed how allowing for interventions on the environment's properties yields a diverse familiy of tasks with a natural way of defining learning curricula and evaluation protocols that can disentangle different

generalization capabilities. A natural extension of our work is to develop new RL algorithms that focuses on out-of-distribution generalization (whether its a different subspace of the state space or a completely different task).We hope that the flexibility and modularity of `CausalWorld` will allow researchers to easily define appropriate benchmarks of increasing difficulty as the field progresses, thereby coordinating research efforts towards ever new goals.

On the Transfer of Disentangled Representations in Realistic Settings

4

Abstract

Learning meaningful representations that disentangle the underlying structure of the data generating process is considered to be of key importance in machine learning. While disentangled representations were found to be useful for diverse tasks such as abstract reasoning and fair classification, their scalability and real-world impact remain questionable. We introduce a new high-resolution dataset with 1M simulated images and over 1,800 annotated real-world images of the same setup. In contrast to previous work, this new dataset exhibits correlations, a complex underlying structure, and allows to evaluate transfer to unseen simulated and real-world settings where the encoder i) remains in distribution or ii) is out of distribution. We propose new architectures in order to scale disentangled representation learning to realistic high-resolution settings and conduct a large-scale empirical study of disentangled representations on this dataset. We observe that disentanglement is a good predictor for out-of-distribution (OOD) task performance.

4.1 Introduction 49
4.2 Related Work 50
4.3 Scaling Disentangled Representations to Complex Scenarios . . . 52
4.4 Framework for the Evaluation of OOD Generalization 56
4.5 Benefits and Transfer of Structured Representations 57
4.6 Conclusion 60

About This Chapter

This chapter has been adapted from **Dittadi** et al. [3].

On the Transfer of Disentangled Representations in Realistic Settings

Andrea Dittadi*, **Frederik Träuble***, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer and Bernhard Schölkopf

ICLR 2021: International Conference on Learning Representations.

AD and FT were shared first authors. Asterisk denotes equal contribution.

Author Contributions based on CRediT (Contributor Roles Taxonomy)

Andrea Dittadi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Frederik Träuble**: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Francesco Locatello**: Writing - Review & Editing **Manuel Wüthrich**: Writing - Review & Editing **Vaibhav Agrawal**: Software, Data Curation, Writing - Review & Editing, Visualization **Ole Winther**: Writing - Review & Editing **Stefan Bauer**: Conceptualization, Validation, Investigation, Writing - Review & Editing, Supervision, Project administration **Bernhard Schölkopf**: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Resources, Funding acquisition

4.1 Introduction

Disentangled representations hold the promise of generalization to unseen scenarios [71], increased interpretability [72, 73] and faster learning on downstream tasks [70, 157]. However, most of the focus in learning disentangled representations has been on small synthetic datasets whose ground truth factors exhibit perfect independence by design. More realistic settings remain largely unexplored. We hypothesize that this is because real-world scenarios present several challenges that have not been extensively studied to date. Important challenges are scaling (much higher resolution in observations and factors), occlusions, and correlation between factors. Consider, for instance, a robotic arm moving a cube: Here, the robot arm can occlude parts of the cube, and its end-effector position exhibits correlations with the cube's position and orientation, which might be problematic for common disentanglement learners [1]. Another difficulty is that we typically have only limited access to ground truth labels in the real world, which requires robust frameworks for model selection when no or only weak labels are available.

[71]: Higgins et al. (2017), 'DARLA: Improving Zero-Shot Transfer in Reinforcement Learning'

[72]: Adel et al. (2018), 'Discovering interpretable representations for both deep generative and discriminative models'

[73]: Higgins et al. (2018), 'SCAN: Learning Hierarchical Compositional Visual Concepts'

[70]: Locatello et al. (2019), 'On the fairness of disentangled representations'

[157]: Steenkiste et al. (2019), 'Are Disentangled Representations Helpful for Abstract Visual Reasoning?'

[1]: **Träuble** et al. (2021), 'On Disentangled Representations Learned from Correlated Data'

The goal of this work is to provide a path towards disentangled representation learning in realistic settings. First, we argue that this requires a new dataset that captures the challenges mentioned above. We propose a dataset consisting of simulated observations from a scene where a robotic arm interacts with a cube in a stage (see Figure 4.1). This setting exhibits correlations and occlusions that are typical in real-world robotics. Second, we show how to scale the architecture of disentanglement methods to perform well on this dataset. Third, we extensively analyze the usefulness of disentangled representations in terms of out-of-distribution downstream generalization, both in terms of held-out factors of variation and sim-to-real transfer. In fact, our dataset is based on the TriFinger robot from Wüthrich et al. [128], which can be built to test the deployment of models in the real world. While the analysis in this paper focuses on the transfer and generalization of predictive models, we hope that our dataset may serve as a benchmark to explore the usefulness of disentangled representations in real-world control tasks.

The contributions of this paper can be summarized as follows:

- We propose a new dataset for disentangled representation learning, containing 1M simulated high-resolution images from a robotic setup, with seven partly correlated factors of variation. Additionally, we provide a dataset of over 1,800 annotated images from the corresponding real-world setup

that can be used for challenging sim-to-real transfer tasks. These datasets are made publicly available.
- We propose a new neural architecture to successfully scale VAE-based disentanglement learning approaches to complex datasets.
- We conduct a large-scale empirical study on generalization to various transfer scenarios on this challenging dataset. We train 1,080 models using state-of-the-art disentanglement methods and discover that disentanglement is a good predictor for out-of-distribution (OOD) performance of downstream tasks.

4.2 Related Work

Disentanglement methods. Most state-of-the-art disentangled representation learning approaches are based on the framework of variational autoencoders (VAEs) [54, 55]. A (high-dimensional) observation $\boldsymbol{x}$ is assumed to be generated according to the latent variable model $p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$ where the latent variables $\boldsymbol{z}$ have a fixed prior $p(\boldsymbol{z})$. The generative model $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and the approximate posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ are typically parameterized by neural networks, which are optimized by maximizing the evidence lower bound (ELBO):

$$\begin{aligned}\mathcal{L}_{VAE} &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \\ &\leq \log p(\boldsymbol{x})\end{aligned} \tag{4.1}$$

As the above objective does not enforce any structure on the latent space except for some similarity to $p(\boldsymbol{z})$, different regularization strategies have been proposed, along with evaluation metrics to gauge the disentanglement of the learned representations [57–61, 84]. Recently, [65] (Theorem 1) showed that the purely unsupervised learning of disentangled representations is impossible. This limitation can be overcome without the need for explicitly labeled data by introducing weak labels [67, 86]. Ideas related to disentangling the factors of variation date back to the non-linear ICA literature [98–102, 158, 159]. Recent work combines non-linear ICA with disentanglement [64, 103, 104].

Evaluating disentangled representations. The *BetaVAE* [57] and *FactorVAE* [59] scores measure disentanglement by performing an intervention on the factors of variation and predicting which factor was intervened on. The *Mutual Information Gap (MIG)* [60], *Modularity* [160], *DCI Disentanglement* [84] and *SAP* scores [61] are based on matrices relating factors of variation and codes (e.g. pairwise mutual information, feature importance and predictability).

[54]: Kingma et al. (2014), 'Auto-encoding variational Bayes'
[55]: Rezende et al. (2014), 'Stochastic backpropagation and approximate inference in deep generative models'
[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'
[58]: Burgess et al. (2018), 'Understanding disentangling in beta-VAE'
[59]: Kim et al. (2018), 'Disentangling by factorising'
[60]: Chen et al. (2018), 'Isolating Sources of Disentanglement in Variational Autoencoders'
[61]: Kumar et al. (2018), 'Variational inference of disentangled latent concepts from unlabeled observations'
[84]: Eastwood et al. (2018), 'A framework for the quantitative evaluation of disentangled representations'
[65]: Locatello et al. (2019), 'Challenging common assumptions in the unsupervised learning of disentangled representations'
[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'
[86]: Shu et al. (2020), 'Weakly Supervised Disentanglement with Guarantees'
[98]: Comon (1994), 'Independent component analysis, a new concept?'
[99]: Jutten et al. (2003), 'Advances in nonlinear blind source separation'
[100]: Hyvärinen et al. (1999), 'Nonlinear independent component analysis: Existence and uniqueness results'
[101]: Hyvarinen et al. (2019), 'Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning'
[102]: Hyvarinen et al. (2016), 'Unsupervised feature extraction by time-contrastive learning and nonlinear ICA'
[158]: Bach et al. (2002), 'Kernel independent component analysis'
[159]: Gresele et al. (2019), 'The Incomplete Rosetta Stone Problem: Identifiability Results for Multi-View Nonlinear ICA'

Datasets for disentanglement learning. *dSprites* [57], which consists of binary low-resolution 2D images of basic shapes, is one of the most commonly used synthetic datasets for disentanglement learning. *Color-dSprites*, *Noisy-dSprites*, and *Scream-dSprites* are slightly more challenging variants of dSprites. The *SmallNORB* dataset contains toy images rendered under different lighting conditions, elevations and azimuths [92]. *Cars3D* [91] exhibits different car models from Fidler, Dickinson, and Urtasun [161] under different camera viewpoints. *3dshapes* (referred to as Shapes3D in the previous chapter) is a popular dataset of simple shapes in a 3D scene [59]. Finally, Gondal et al. [93] proposed *MPI3D*, containing images of physical 3D objects with seven factors of variation, such as object color, shape, size and position available in a simulated, simulated and highly realistic rendered simulated variant. Except MPI3D which has over 1M images, the size of the other datasets is limited with only $17,568$ to $737,280$ images. All of the above datasets exhibit perfect independence of all factors, the number of possible states is on the order of 1M or less, and due to their static setting they do not allow for dynamic downstream tasks such as reinforcement learning. In addition, except for SmallNORB, the image resolution is limited to 64x64 and there are no occlusions.

Other related work. Locatello et al. [67] probed the out-of-distribution generalization of downstream tasks trained on disentangled representations. However, these representations are trained on the entire dataset. Generalization and transfer performance especially for representation learning has likewise been studied in Dayan [162], Muandet, Balduzzi, and Schölkopf [163], Heinze-Deml and Meinshausen [164], Rojas-Carulla et al. [165], Suter et al. [166], Li et al. [167], Arjovsky et al. [168], Krueger et al. [169], and Gowal et al. [170]. For the role of disentanglement in causal representation learning we refer to the recent overview by Schölkopf et al. [46]. **Träuble** et al. [1] systematically investigated the effects of correlations between factors of variation on disentangled representation learners. Transfer of learned disentangled representations from simulation to the real world has been recently investigated by Gondal et al. [93] on the MPI3D dataset, and previously by Higgins et al. [71] in the context of reinforcement learning. Sim-to-real transfer is of major interest in the robotic learning community, because of limited data and supervision in the real world [41, 171–175].

[64]: Klindt et al. (2021), 'Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding'

[103]: Khemakhem et al. (2020), 'Variational autoencoders and nonlinear ica: A unifying framework'

[104]: Sorrenson et al. (2020), 'Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN)'

[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'

[59]: Kim et al. (2018), 'Disentangling by factorising'

[60]: Chen et al. (2018), 'Isolating Sources of Disentanglement in Variational Autoencoders'

[160]: Ridgeway et al. (2018), 'Learning Deep Disentangled Embeddings with the F-Statistic Loss'

[84]: Eastwood et al. (2018), 'A framework for the quantitative evaluation of disentangled representations'

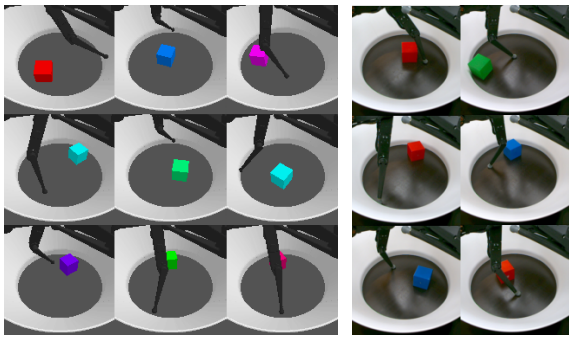
[61]: Kumar et al. (2018), 'Variational inference of disentangled latent concepts from unlabeled observations'

[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'

[92]: LeCun et al. (2004), 'Learning methods for generic object recognition with invariance to pose and lighting'

[91]: Reed et al. (2015), 'Deep visual analogy-making'

[59]: Kim et al. (2018), 'Disentangling by factorising'



FoV	Values
Upper joint	30 values in $[-0.65, +0.65]$
Middle joint	30 values in $[-0.5, +0.5]$
Lower joint	30 values in $[-0.8, +0.8]$
Cube position x	30 values in $[-0.11, +0.11]$
Cube position y	30 values in $[-0.11, +0.11]$
Cube rotation	10 values in $[0°, 81°]$
Cube color hue	12 values in $[0°, 330°]$

Figure 4.1: Left: Images from the simulated dataset and from the real-world setup. **Right**: Factors of variation in the proposed dataset. Values are linearly spaced in the specified intervals. Joint angles are in radians, cube positions in meters.

[41]: Andrychowicz et al. (2020), 'Learning dexterous in-hand manipulation'
[171]: Tobin et al. (2017), 'Domain randomization for transferring deep neural networks from simulation to the real world'
[172]: Rusu et al. (2017), 'Sim-to-real robot learning from pixels with progressive nets'
[173]: Peng et al. (2018), 'Sim-to-real transfer of robotic control with dynamics randomization'
[174]: James et al. (2019), 'Sim-to-real via sim-to-sim: Data-efficient robotic grasping via random-ized-to-canonical adaptation networks'
[175]: Yan et al. (2020), 'How to Close Sim-Real Gap? Transfer with Segmentation!'

4.3 Scaling Disentangled Representations to Complex Scenarios

A new challenging dataset. Simulated images in our dataset are derived from the TriFinger robot platform introduced by Wüthrich et al. [128]. The motivation for choosing this setting is that (1) it is challenging due to occlusions, correlations, and other difficulties encountered in robotic settings, (2) it requires modeling of fine details such as tip links at high resolutions, and (3) it corresponds to a robotic setup, so that learned representations can be used for control and reinforcement learning in simulation and in the real world. The scene comprises a robot finger with three joints that can be controlled to manipulate a cube in a bowl-shaped stage. Figure 4.1 shows examples of scenes from our dataset. The data is generated from 7 different factors of variation (FoV) listed in Figure 4.1. Unlike in previous datasets, not all FoVs are independent: The end-effector (the tip of the finger) can collide with the floor or the cube, resulting in infeasible combinations of the factors (see Appendix D.2.1). We argue that such correlations are a key feature in real-world data that is not present in existing datasets. The high FoV resolution results in approximately 1.52 billion feasible states, but the dataset itself only contains one million of them (approximately 0.065% of all possible FoV combinations), realistically rendered into 128×128 images. Additionally, we recorded an annotated dataset under the same conditions in the real-world setup: we acquired 1,809 camera images from the same viewpoint and recorded the labels of the 7 underlying factors of variation. This dataset can be used for out-of-distribution evaluations, few-shot learning, and testing other sim-to-real aspects.

Model architecture. When scaling disentangled representation learning to more complex datasets, such as the one proposed here, one of the main bottlenecks in current VAE-based approaches is the flexibility of the encoder and decoder networks. In particular,

using the architecture from Locatello et al. [65], none of the models we trained correctly captured all factors of variation or yielded high-quality reconstructions. While the increased image resolution already presents a challenge, the main practical issue in our new dataset is the level of detail that needs to be modeled. In particular, we identified the cube rotation and the lower joint position to be the factors of variation that were the hardest to capture. This is likely because these factors only produce relatively small changes in the image and hence the reconstruction error.

To overcome these issues, we propose a deeper and wider neural architecture than those commonly used in the disentangled representation learning literature, where the encoder and decoder typically have 4 convolutional and 2 fully-connected layers. Our encoder consists of a convolutional layer, 10 residual blocks, and 2 fully-connected layers. Some residual blocks are followed by 1x1 convolutions that change the number of channels, or by average pooling that downsamples the tensors by a factor of 2 along the spatial dimensions. Each residual block consists of two 3x3 convolutions with a leaky ReLU nonlinearity, and a learnable scalar gating mechanism [176]. Overall, the encoder has 23 convolutional layers and 2 fully connected layers. The decoder mirrors this architecture, with average pooling replaced by bilinear interpolation for upsampling. The total number of parameters is approximately 16.3M. See Appendix D.1 for further implementation details.

[176]: Bachlechner et al. (2021), 'Rezero is all you need: Fast convergence at large depth'

Experimental setup. We perform a large-scale empirical study on the simulated dataset introduced above by training 1,080 β-VAE models.* For further experimental details we refer the reader to Appendix D.1. The hyperparameter sweep is defined as follows:

- We train the models using either unsupervised learning or weakly supervised learning [67]. In the weakly supervised case, a model is trained with pairs of images that differ in k factors of variation. Here we fix $k = 1$ as it was shown to lead to higher disentanglement by Locatello et al. [67]. The dataset therefore consists of 500k pairs of images that differ in only one FoV.
- We vary the parameter β in $\{1, 2, 4\}$, and use linear deterministic warm-up [177, 178] over the first $\{0, 10000, 50000\}$ training steps.
- The latent space dimensionality is in $\{10, 25, 50\}$.
- Half of the models are trained with additive noise in the input image. This choice is motivated by the fact that adding noise to the input of neural networks has been shown to be beneficial for out-of-distribution generalization [179, 180].

[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'

[177]: Bowman et al. (2015), 'Generating sentences from a continuous space'
[178]: Sønderby et al. (2016), 'Ladder variational autoencoders'

[179]: Sietsma et al. (1991), 'Creating artificial neural networks that generalize'
[180]: Bishop (1995), 'Training with noise is equivalent to Tikhonov regularization'

* Training these models requires approximately 2.8 GPU years on NVIDIA Tesla V100 PCIe.

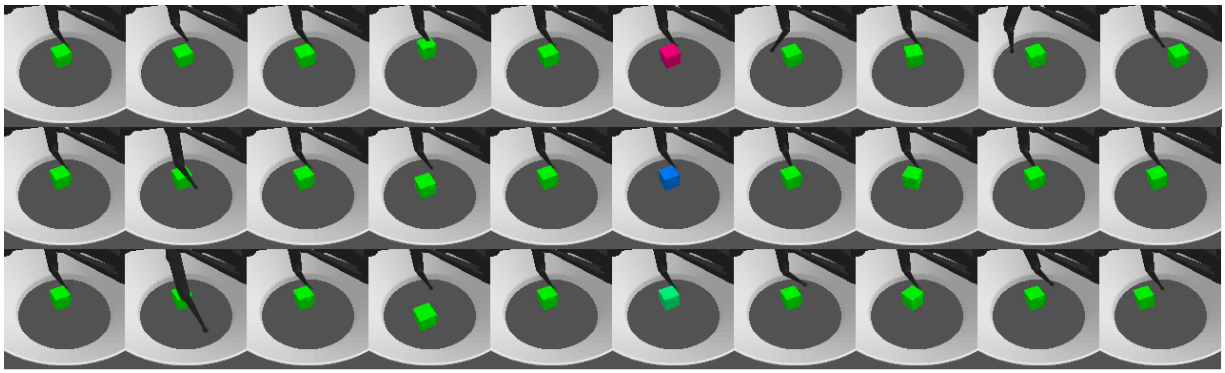


Figure 4.2: Latent traversals of a trained model that perfectly disentangles the dataset's FoVs. In each column, all latent variables but one are fixed.

- Each of the 108 resulting configurations is trained with 10 random seeds.

Can we scale up disentanglement learning? Most of the trained VAEs in our empirical study fully capture all the elements of a scene, correctly model heavy occlusions, and generate detailed, high-quality samples and reconstructions (see Appendix D.2.2). From visual inspections such as the latent traversals in Figure 4.2, we observe that many trained models fully disentangle the ground-truth factors of variation. This, however, appears to only be possible in the weakly supervised scenario. The fact that models trained without supervision learn entangled representations is in line with the impossibility result for the unsupervised learning of disentangled representations from [1, 65]. Latent traversals from a selection of models with different degrees of disentanglement are presented in Appendix D.2.3. Interestingly, the high-disentanglement models seem to correct for correlations and interpolate infeasible states, i.e. the fingertip traverses through the cube or the floor.

[1]: **Träuble** et al. (2021), 'On Disentangled Representations Learned from Correlated Data'
[65]: Locatello et al. (2019), 'Challenging common assumptions in the unsupervised learning of disentangled representations'

Summary: The proposed architecture can scale disentanglement learning to more realistic settings, but a form of weak supervision is necessary to achieve high disentanglement.

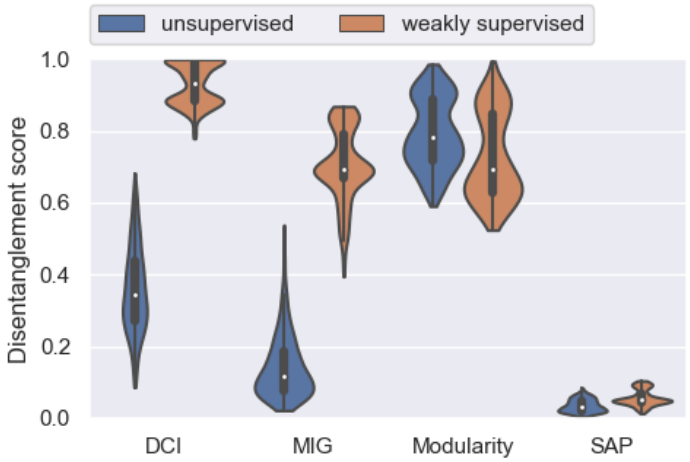


Figure 4.3: Disentanglement metrics aggregating all hyperparameters except for unsupervised and weakly-supervised learning.

How useful are common disentanglement metrics in realistic scenarios? The violin plot in Figure 4.3 shows that DCI and MIG measure high disentanglement under weak supervision and lower disentanglement in the unsupervised setting. This is consistent with our qualitative conclusion from visual inspection of the models (Appendix D.2.3) and with the aforementioned impossibility result. Many of the models trained with weak supervision exhibit a very high DCI score (29% of them have >99% DCI, some of them up to 99.89%). SAP and Modularity appear to be ineffective at capturing disentanglement in this setting, as also observed by Locatello et al. [65]. Finally, note that the BetaVAE and FactorVAE metrics are not straightforward to be evaluated on datasets that do not contain all possible combinations of factor values. According to Figure 4.4, DCI and MIG strongly correlate with test accuracy of GBT classifiers predicting the FoVs. In the weakly supervised setting, these metrics are strongly correlated with the ELBO (positively) and with the reconstruction loss (negatively). We illustrate these relationships in more detail in Appendix D.2.4. Such correlations were also observed by Locatello et al. [67] on significantly less complex datasets, and can be exploited for unsupervised model selection: these unsupervised metrics can be used as proxies for disentanglement metrics, which would require fully labeled data.

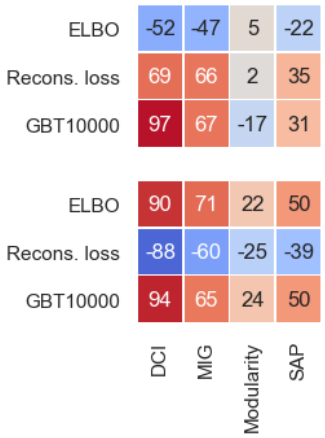


Figure 4.4: Rank correlations (Spearman) of ELBO, reconstruction loss, and the test error of a GBT classifier trained on 10,000 labelled data points with disentanglement metrics. The upper rank correlations correspond to the unsupervised models and the lower to the weakly supervised models.

Summary: DCI and MIG appear to be useful disentanglement metrics in realistic scenarios, whereas other metrics seem to fall short of capturing disentanglement or can be difficult to compute. When using weak supervision, we can select disentangled models with unsupervised metrics.

4.4 Framework for the Evaluation of OOD Generalization

Previous work has focused on evaluating the usefulness of disentangled representations for various downstream tasks, such as predicting ground truth factors of variation, fair classification, and abstract reasoning. Here we propose a new framework for evaluating the out-of-distribution (OOD) generalization properties of representations. More specifically, we consider a downstream task – in our case, regression of ground truth factors – trained on a learned representation of the data, and evaluate the performance on a held-out test set. While the test set typically follows the same distribution as the training set (in-distribution generalization), we also consider test sets that follow a different distribution (out-of-distribution generalization). Our goal is to investigate to what extent, if at all, downstream tasks trained on disentangled representations exhibit a higher degree of OOD generalization than those trained on entangled representations.

Let D denote the training set for disentangled representation learning. To investigate OOD generalization, we train downstream regression models on a subset $D_1 \subset D$ to predict ground truth factor values from the learned representation computed by the encoder. We independently train one predictor per factor. We then test the regression models on a set D_2 that differs distributionally from the training set D_1, as it either contains images corresponding to held-out values of a chosen FoV (e.g. unseen object colors), or it consists of real-world images. We now differentiate between two scenarios: (1) $D_2 \subset D$, i.e. the OOD test set is a subset of the dataset for representation learning; (2) D and D_2 are disjoint and distributionally different. These two scenarios will be denoted by *OOD1* and *OOD2*, respectively. For example, consider the case in which distributional shifts are based on one FoV: the color of the object. Then, we could define these datasets such that images in D always contain a red or blue object, and those in $D_1 \subset D$ always contain a red object. In the OOD1 scenario, images in D_2 would always contain a blue object, whereas in the OOD2 case they would always contain an object that is neither red nor blue.

The regression models considered here are Gradient Boosted Trees (GBT), random forests, and MLPs with $\{1, 2, 3\}$ hidden layers. Since random forests exhibit a similar behavior to GBTs, and all MLPs yield similar results to each other, we choose GBTs and the 2-layer MLP as representative models and only report results for those. To quantify prediction quality, we normalize the ground truth factor values to the range $[0, 1]$, and compute the mean absolute error (MAE). Since the values are normalized, we can define our transfer

metric as the average of the MAE over all factors (except for the FoV that is OOD).

4.5 Benefits and Transfer of Structured Representations

Experimental setup. We evaluate the transfer metric introduced in Section 4.4 across all 1,080 trained models. To compute this metric, we train regression models to predict the ground truth factors of variation, and test them under distributional shift. We consider distributional shifts in terms of cube color or sim-to-real, and we do not evaluate downstream prediction of cube color. We report scores for two different regression models: a Gradient Boosted Tree (GBT) and an MLP with 2 hidden layers of size 256. In Appendix D.1 we provide details on the datasets used in this section.

In the OOD1 setting, we have $D_2 \subset D$, hence the encoder is in-distribution: we are testing the predictor on representations of images that were in the training set of the representation learning algorithm. Therefore, we expect the representations to be meaningful. We consider three scenarios:

- OOD1-A: The regression models are trained on 1 cube color (red) and evaluated on the remaining 7 colors.
- OOD1-B: The regression models are trained on 4 cube colors with high hue in the HSV space, and evaluated on 4 cube colors with low hue (extrapolation).
- OOD1-C: The regression models are again trained and evaluated on 4 cube colors, but the training and evaluation colors are alternating along the hue dimension (interpolation).

In the more challenging setting where even the encoder goes out-of-distribution (OOD2, with $D_2 \cap D = \varnothing$), we train the regression models on a subset of the training set D that includes all 8 cube colors, and we consider the two following scenarios:

- OOD2-A: The regression models are evaluated on simulated data, on 4 cube colors that are out of the encoder's training distribution.
- OOD2-B: The regression models are evaluated on real-world images of the robotic setup, without any adaptation or fine-tuning.

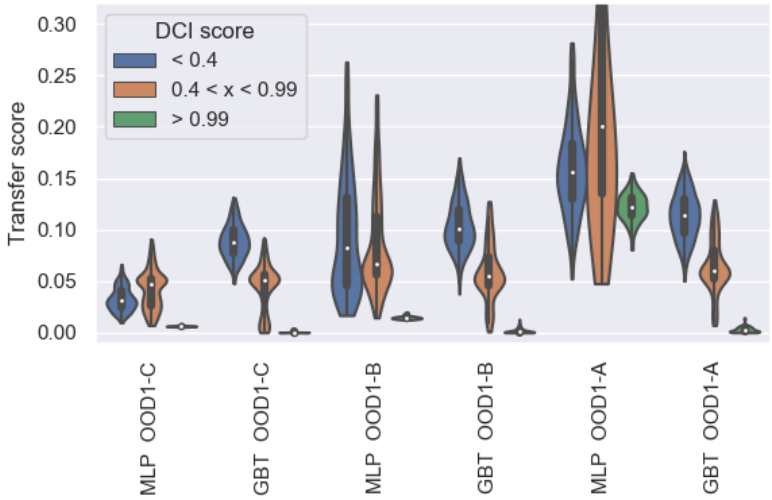


Figure 4.5: Higher disentanglement corresponds to better generalization across all OOD1 scenarios, as seen from the transfer scores. The transfer score is computed as the mean absolute prediction error of ground truth factor values (lower is better). This correlation is particularly evident in the GBT case, whereas MLPs appear to exhibit better OOD1 transfer with very high disentanglement only.

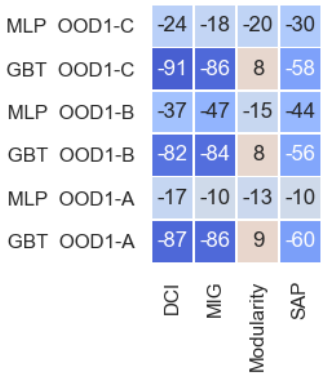


Figure 4.6: Spearman rank correlations between transfer scores and disentanglement metrics.

Is disentanglement correlated with OOD1 generalization? In Figure 4.5 and Figure 4.6 we consistently observe a negative correlation between disentanglement and transfer error across all OOD1 settings. The correlation is mild when using MLPs, strong when using GBTs. This difference is expected, as GBTs have an axis-alignment bias whereas MLPs can – given enough data and capacity – disentangle an entangled representation more easily. Our results therefore suggest that **highly disentangled representations are useful for generalizing out-of-distribution as long as the encoder remains in-distribution**. This is in line with the correlation found by Locatello et al. [65] between disentanglement and the GBT10000 metric. There, however, GBTs are tested on the same distribution as the training distribution, while here we test them under distributional shift. Given that the computation of disentanglement scores requires labels, this is of little benefit in the unsupervised setting. However, it can be exploited in the weakly supervised setting, where disentanglement was shown to correlate with ELBO and reconstruction loss (Section 4.3). Therefore, model selection for representations that transfer well in these scenarios is feasible based on the ELBO or reconstruction loss, when weak supervision is available. Note that, in absolute terms, the OOD generalization error with encoder in-distribution (OOD1) is very low in the high-disentanglement case (the only exception being the MLP in the OOD1-C case, with the 1-7 color split, which seems to overfit). This suggests that disentangled representations can be useful in downstream tasks even when transferring out of the training distribution.

Summary: Disentanglement seems to be positively correlated with OOD generalization of downstream tasks, provided that the encoder remains in-distribution (OOD1). Since in the weakly supervised case disentanglement correlates with the ELBO and

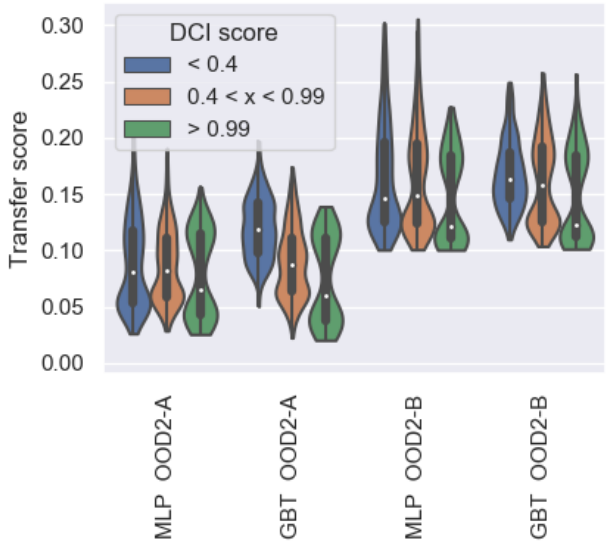


Figure 4.7: As seen from transfer scores, disentanglement (DCI) affects generalization across the OOD2 scenarios only minimally.

the reconstruction loss, model selection can be performed using these metrics as proxies for disentanglement. These metrics have the advantage that they can be computed without labels, unlike disentanglement metrics.



Figure 4.8: Rank correlations between transfer scores on OOD2 scenarios and various disentanglement metrics.

Is disentanglement correlated with OOD2 generalization? As seen in Figure 4.7 and Figure 4.8, the negative correlation between disentanglement and GBT transfer error is weaker when the encoder is out-of-distribution (OOD2). Nonetheless, we observe a non-negligible correlation for GBTs in the OOD2-A case, where we investigate out-of-distribution generalization along one FoV, with observations in D_2 still generated from the same simulator. In the OOD2-B setting, where the observations are taken from cameras in the corresponding real-world setting, the correlation between disentanglement and transfer performance appears to be minor at best. This scenario can be considered a variant of zero-shot sim-to-real generalization.

Summary: Disentanglement has a minor effect regarding out-of-distribution generalization outside of the training distribution of the encoder (OOD2).

What else matters for OOD2 generalization? Results in Figure 4.9 suggest that adding Gaussian noise to the input during training as described in Section 4.3 leads to significantly better OOD2 generalization, and has no effect on OOD1 generalization. Adding noise to the input of neural networks is known to lead to better generalization [179, 180]. This is in agreement with our results, since OOD1 generalization does not require generalization of the encoder, while OOD2 does.

[179]: Sietsma et al. (1991), 'Creating artificial neural networks that generalize'
[180]: Bishop (1995), 'Training with noise is equivalent to Tikhonov regularization'

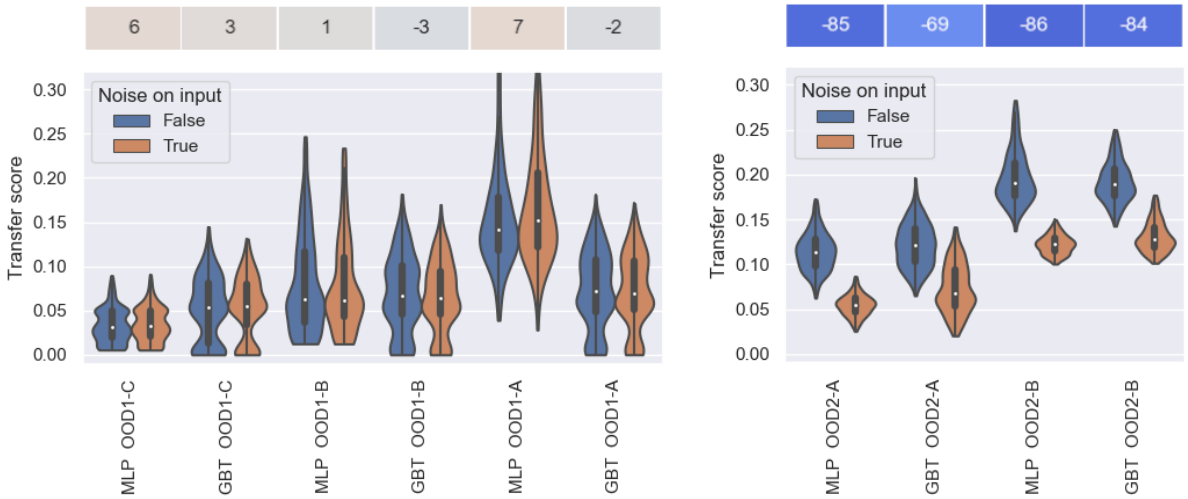


Figure 4.9: Noise improves generalization across the OOD2 scenarios and less so for the OOD1 scenarios as seen from the transfer scores. Top row: Spearman rank correlation coefficients between transfer metrics and presence of noise in the input.

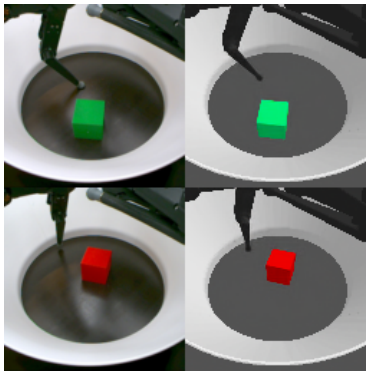


Figure 4.10: Zero-shot transfer to real-world observations of our models trained in simulation. Left: input; right: reconstruction.

Interestingly, closer inspection reveals that the contribution of different factors of variation to the generalization error can vary widely. See Appendix D.2.5 for further details. In particular, with noisy input, the position of the cube is predicted accurately even in real-world images (<5% mean absolute error on each axis). This is promising for robotics applications, where the true state of the joints is observable but inference of the cube position relies on object tracking methods. Figure 4.10 shows an example of real-world inputs and reconstructions of their simulated equivalents.

Summary: Adding input noise during training appears to be significantly beneficial for OOD2 generalization, while having no effect when the encoder is kept in its training distribution (OOD1).

4.6 Conclusion

Despite the growing importance of the field and the potential societal impact in the medical domain [81] and fair decision making [70], state-of-the-art approaches for learning disentangled representations have so far only been systematically evaluated on synthetic toy datasets. Here we introduced a new high-resolution dataset with 1M simulated images and over 1,800 annotated real-world images of the same setup. This dataset exhibits a number of challenges and features which are not present in previous datasets: it contains correlations between factors, occlusions, a complex underlying structure, and it allows for evaluation of transfer to unseen simulated and real-world settings. We proposed a new VAE architecture to scale disentangled representation learning to

[81]: Chartsias et al. (2018), 'Factorised spatial representation learning: Application in semi-supervised myocardial segmentation'

[70]: Locatello et al. (2019), 'On the fairness of disentangled representations'

this realistic setting and conducted a large-scale empirical study of disentangled representations on this dataset. We discovered that disentanglement is a good predictor of OOD generalization of downstream tasks and showed that, in the context of weak supervision, model selection for good OOD performance can be based on the ELBO or the reconstruction loss, which are accessible without explicit labels. Our setting allows for studying a wide variety of interesting downstream tasks in the future, such as reinforcement learning or learning a dynamics model of the environment. Finally, we believe that in the future it will be important to take further steps in the direction of this paper by considering settings with even more complex structures and stronger correlations between factors.

The Role of Pretrained Representations for the OOD Generalization of RL Agents

5

5.1 Introduction 65
5.2 Background 67
5.3 Study Design 69
5.4 Results 72
5.5 Other Related Work . . 79
5.6 Conclusion 79

Abstract

Building sample-efficient agents that generalize out-of-distribution (OOD) in real-world settings remains a fundamental unsolved problem on the path towards achieving higher-level cognition. One particularly promising approach is to begin with low-dimensional, pre-trained representations of our world, which should facilitate efficient downstream learning and generalization. By training 240 representations and over 10,000 reinforcement learning (RL) policies on a simulated robotic setup, we evaluate to what extent different properties of pre-trained VAE-based representations affect the OOD generalization of downstream agents. We observe that many agents are surprisingly robust to realistic distribution shifts, including the challenging sim-to-real case. In addition, we find that the generalization performance of a simple downstream proxy task reliably predicts the generalization performance of our RL agents under a wide range of OOD settings. Such proxy tasks can thus be used to select pre-trained representations that will lead to agents that generalize.

About This Chapter

This chapter has been adapted from **Träuble** et al. [4].

The Role of Pretrained Representations for the OOD Generalization of RL Agents

Frederik Träuble*, Andrea Dittadi*, Manuel Wuthrich, Felix Widmaier, Peter Vincent Gehler, Ole Winther, Francesco Locatello, Olivier Bachem, Bernhard Schölkopf and Stefan Bauer.

ICLR 2022: International Conference on Learning Representations.

AD and FT were shared first authors. Asterisk denotes equal contribution.

Author Contributions based on CRediT (Contributor Roles Taxonomy)

Frederik Träuble: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Andrea Dittadi**: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Manuel Wüthrich**: Writing - Review & Editing **Felix Widmaier**: Software **Peter Gehler**: Writing - Review & Editing, Resources **Ole Winther**: Writing - Review & Editing **Francesco Locatello**: Writing - Review & Editing **Olivier Bachem**: Writing - Review & Editing **Bernhard Schölkopf**: Conceptualization, Resources, Writing - Review & Editing, Supervision, Funding acquisition **Stefan Bauer**: Conceptualization, Validation, Investigation, Writing - Review & Editing, Supervision, Project administration

5.1 Introduction

Robust out-of-distribution (OOD) generalization is one of the key open challenges in machine learning. This is particularly relevant for the deployment of ML models to the real world, where we need systems that generalize beyond the i.i.d. (independent and identically distributed) data setting [46, 181–189]. One instance of such models are agents that learn by interacting with a training environment and we would like them to generalize to other environments with different statistics [2, 118, 190–192]. Consider the example of a robot with the task of moving a cube to a target position: Such an agent can easily fail as soon as some aspects of the environment differ from the training setup, e.g. the shape, color, and other object properties, or when transferring from simulation to real world.

Humans do not suffer from these pitfalls when transferring learned skills beyond a narrow training domain, presumably because they represent visual sensory data in a concise and useful manner [50, 193–196]. Therefore, a particularly promising path is to base predictions and decisions on similar low-dimensional representations of our world [36, 49, 197–201]. The learned representation should facilitate efficient downstream learning [68, 195, 201, 202] and exhibit better generalization [203, 204]. Learning such a representation from scratch for every downstream task and every new variation would be inefficient. If we learned to juggle three balls, we should be able to generalize to oranges or apples without learning again from scratch. We could even do it with cherimoyas, a fruit that we might have never seen before. We can effectively reuse our generic representation of the world.

We thus consider deep learning agents trained from pre-trained representations and ask the following questions: To what extent do they generalize under distribution shifts similar to those mentioned above? Do they generalize in different ways or to different degrees depending on the type of distribution shift, including sim-to-real? Can we predict the OOD generalization of downstream agents from properties of the pre-trained representations?

To answer the questions above, we need our experimental setting to be realistic, diverse, and challenging, but also controlled enough for the conclusions to be sound. We therefore base our study on the robot platform introduced by Wüthrich et al. [128]. The scene comprises a robot finger with three joints that can be controlled to manipulate a cube in a bowl-shaped stage. **Dittadi** et al. [3] conveniently introduced a dataset of simulated and real-world images of this setup with ground-truth labels, which can be used to pre-train and evaluate representations. To train downstream agents,

[46]: Schölkopf et al. (2021), 'Toward causal representation learning'
[181]: Djolonga et al. (2021), 'On robustness and transferability of convolutional neural networks'
[182]: Koh et al. (2021), 'Wilds: A benchmark of in-the-wild distribution shifts'
[183]: Barbu et al. (2019), 'ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models'
[184]: Azulay et al. (2019), 'Why do deep convolutional networks generalize so poorly to small image transformations?'
[185]: Roy et al. (2018), 'Effects of degradations on deep neural network architectures'
[186]: Gulrajani et al. (2020), 'In Search of Lost Domain Generalization'
[187]: Hendrycks et al. (2019), 'Benchmarking Neural Network Robustness to Common Corruptions and Perturbations'
[188]: Michaelis et al. (2019), 'Benchmarking robustness in object detection: Autonomous driving when winter is coming'
[189]: Funk et al. (2021), 'Benchmarking Structured Policies and Policy Optimization for Real-World Dexterous Object Manipulation'

we adapted the simulated reinforcement learning benchmark CausalWorld from **Ahmed** et al. [2] that was developed for this platform. Building upon these works, we design our experimental study as follows (see Figure 5.1): First, we pre-train representations from static simulated images of the setup and evaluate a collection of representation metrics. Following prior work [202, 205–209], we focus on autoencoder-based representations. Then, we train downstream agents from this fixed representation on a set of environments. Finally, we investigate the zero-shot generalization of these agents to new environments that are out of the training distribution, including the real robot.

The goal of this work is to provide the first systematic and extensive account of the OOD generalization of downstream RL agents in a robotic setup, and how this is affected by characteristics of the upstream pre-trained representations. We summarize our contributions as follows:

- We train 240 representations and 11,520 downstream policies,[1] and systematically investigate their performance under a diverse range of distribution shifts.
- We extensively analyze the relationship between the generalization of our RL agents and a substantial set of representation metrics.
- Notably, we find that a specific representation metric that measures the generalization of a simple downstream proxy task reliably predicts the generalization of downstream RL agents under the broad spectrum of OOD settings considered here. This metric can thus be used to select pre-trained representations that will lead to more robust downstream policies.
- In the most challenging of our OOD scenarios, we deploy a subset of the trained policies to the corresponding real-world robotic platform, and observe surprising zero-shot sim-to-real generalization without any fine-tuning or domain randomization.

[202]: Eslami et al. (2018), 'Neural scene representation and rendering'
[205]: Watter et al. (2015), 'Embed to control: A locally linear latent dynamics model for control from raw images'
[206]: Van Hoof et al. (2016), 'Stable reinforcement learning with autoencoders for tactile and visual data'
[207]: Ghadirzadeh et al. (2017), 'Deep predictive policy training using reinforcement learning'
[208]: Nair et al. (2018), 'Visual reinforcement learning with imagined goals'
[209]: Ha et al. (2018), 'Recurrent world models facilitate policy evolution'

1: Training the representations required approximately 0.62 GPU years on NVIDIA Tesla V100. Training and evaluating the downstream policies required about 86.8 CPU years on Intel Platinum 8175M.

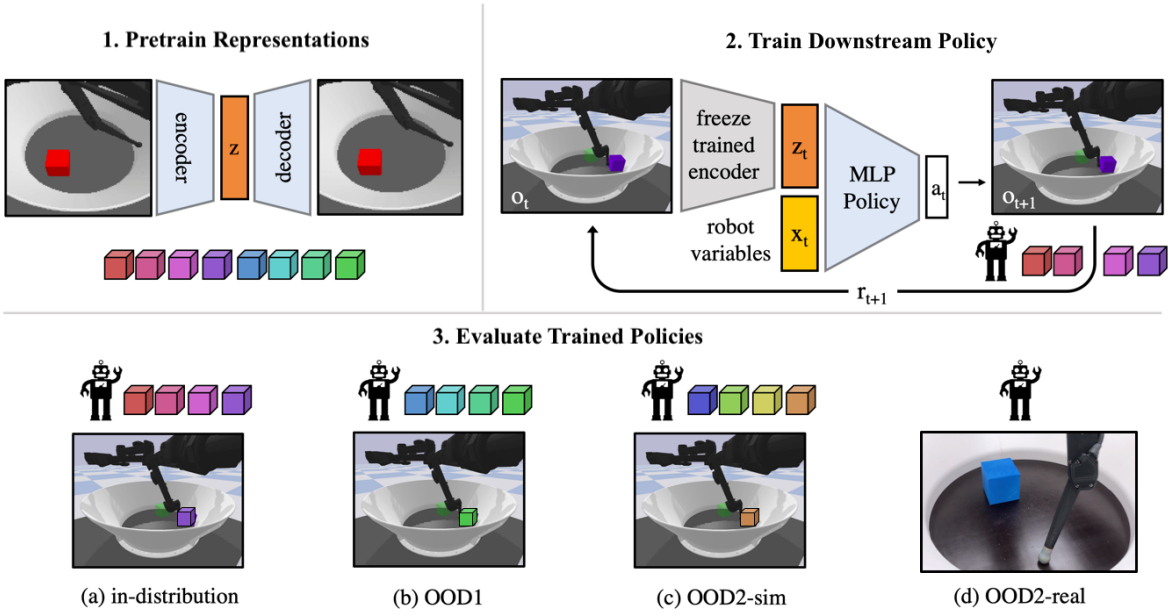


Figure 5.1: Overview of our experimental setup for investigating out-of-distribution generalization in downstream tasks. (1) We train 240 β-VAEs on the robotic dataset from [3]. (2) We then train downstream policies to solve *object reaching* or *pushing*, using multiple random RL seeds per VAE. The input to a policy consists of the output of a pre-trained encoder and additional task-related observable variables. Crucially, the policy is only trained on a subset of the cube colors from the pre-training dataset. (3) Finally, we evaluate these policies on their respective tasks in four different scenarios: (a) in-distribution, i.e. with cube colors used in policy training; (b) OOD1, i.e. with cube colours previously seen by the encoder but OOD for the policy; (c) OOD2-sim, having cube colours also OOD to the encoder; (d) sim-to-real zero-shot on the real-world setup.

5.2 Background

In this section, we provide relevant background on the methods for representation learning and reinforcement learning, and on the robotic setup to evaluate out-of-distribution generalization.

Variational autoencoders. VAEs [54, 55] are a framework for optimizing a latent variable model $p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})d\mathbf{z}$ with parameters θ, typically with a fixed prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, using amortized stochastic variational inference. A variational distribution $q_\phi(\mathbf{z}\,|\,\mathbf{x})$ with parameters ϕ approximates the intractable posterior $p_\theta(\mathbf{z}\,|\,\mathbf{x})$. The approximate posterior and generative model, typically called encoder and decoder and parameterized by neural networks, are jointly optimized by maximizing a lower bound to the log likelihood (the ELBO):

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \mathcal{L}^{ELBO}_{\theta,\phi}(\mathbf{x}) \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_\theta(\mathbf{x}\,|\,\mathbf{z})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}\,|\,\mathbf{x})\|p(\mathbf{z})\right) .\end{aligned} \tag{5.1}$$

In β-VAEs, the KL term is modulated by a factor β to enforce a more structured latent space [57, 58]. While VAEs are typically trained without supervision, we also employ a form of weak supervision

[54]: Kingma et al. (2014), 'Auto-encoding variational Bayes'
[55]: Rezende et al. (2014), 'Stochastic backpropagation and approximate inference in deep generative models'

[57]: Higgins et al. (2017), 'beta-VAE: Learning basic visual concepts with a constrained variational framework'
[58]: Burgess et al. (2018), 'Understanding disentangling in beta-VAE'

that encourages disentanglement [67].

[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'

Reinforcement learning. A Reinforcement Learning (RL) problem is typically modeled as a Partially Observable Markov Decision Process (POMDP) defined as a tuple $(S, A, T, R, \Omega, O, \gamma, \rho_0, H)$ with states $s \in S$, actions $a \in A$ and observations $o \in \Omega$ determined by the state and action of the environment $O(o|s, a)$. $T(s_{t+1}|s_t, a_t)$ is the transition probability distribution function, $R(s_t, a_t)$ is the reward function, γ is the discount factor, $\rho_0(s)$ is the initial state distribution at the beginning of each episode, and H is the time horizon per episode. The objective in RL is to learn a policy $\pi : S \times A \to [0, 1]$, typically parameterized by a neural network, that maximizes the total discounted expected reward $J(\pi) = \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t)\right]$. There is a broad range of model-free learning algorithms to find π^* by policy gradient optimization or by learning value functions while trading off exploration and exploitation [40, 155, 156, 210–213]. Here, we optimize the objective above with *Soft Actor Critic* (SAC), an off-policy method that simultaneously maximizes the expected reward and the entropy $H(\pi(\cdot|s_t))$, and is widely used in control tasks due to its sample efficiency [40].

[40]: Haarnoja et al. (2018), 'Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor'

A robotic setup to evaluate out-of-distribution generalization. Our study is based on a real robot platform where a robotic finger with three joints manipulates a cube in a bowl-shaped stage [128]. We pre-train representations on a labeled dataset introduced by **Dittadi** et al. [3] which consists of simulated and real-world images of this setup. This dataset has 7 underlying factors of variation (FoV): angles of the three joints, and position (x and y), orientation, and color of the cube. Some of these factors are correlated [3], which may be problematic for representation learners, especially in the context of disentanglement [1, 7]. After training the representations, we train downstream agents and evaluate their generalization on an adapted version of the simulated CausalWorld benchmark [2] that was developed for the same setup. Finally, we test sim-to-real generalization on the real robot.

[128]: Wüthrich et al. (2021), 'TriFinger: An Open-Source Robot for Learning Dexterity'

[1]: **Träuble** et al. (2021), 'On Disentangled Representations Learned from Correlated Data'
[7]: **Chen** et al. (2021), 'Boxhead: A Dataset for Learning Hierarchical Representations'

Our experimental setup, illustrated in Figure 5.1, allows us to systematically investigate a broad range of out-of-distribution scenarios in a controlled way. We pre-train our representations from this simulated dataset that covers 8 distinct cube colors. We then train an agent from this fixed representation on a subset of the cube colors, and evaluate it (1) on the same colors (this is the typical scenario in RL), (2) on the held-out cube colors that are still known to the encoder, or (3) OOD w.r.t. the encoder's training distribution, e.g. on novel colors and shapes or on the real world.

We closely follow the framework for measuring OOD generalization proposed by **Dittadi** et al. [3]. In this framework, a representation is initially learned on a training set $\mathcal{D}$, and a simple downstream model is trained on a subset $\mathcal{D}_1 \subset \mathcal{D}$ to predict the ground-truth factors from the learned representation. Generalization is then evaluated by testing the downstream model on a set $\mathcal{D}_2$ that differs distributionally from $\mathcal{D}_1$, e.g. containing images corresponding to held-out values of a chosen factor of variation (FoV). **Dittadi** et al. [3] consider two flavors of OOD generalization depending on the choice of $\mathcal{D}_2$: First, the case when $\mathcal{D}_2 \subset \mathcal{D}$, i.e. the OOD test set is a subset of the dataset for representation learning. This is denoted by **OOD1** and corresponds to the scenario (2) from the previous paragraph. In the other scenario, referred to as **OOD2**, $\mathcal{D}$ and $\mathcal{D}_2$ are disjoint and distributionally different. This even stronger OOD shift corresponds to case (3) above. The generalization score for $\mathcal{D}_2$ is then measured by the (normalized) mean absolute prediction error across all FoVs except for the one that is OOD. Following **Dittadi** et al. [3], we use a simple 2-layer Multi-Layer Perceptron (MLP) for downstream factor prediction, we train one MLP for each FoV, and report the *negative* error. This simple and cheap generalization metric could serve as a convenient proxy for the generalization of more expensive downstream tasks. We refer to these generalization scores as GS-OOD1, GS-OOD2-sim, and GS-OOD2-real depending on the scenario.

The focus of **Dittadi** et al. [3] was to scale VAE-based approaches to more realistic scenarios and study the generalization of these simple downstream tasks, with a particular emphasis on disentanglement. Building upon their contributions, we can leverage the broader potential of this robotic setup with many more OOD2 scenarios to study our research questions: To what extent can agents generalize under distribution shift? Do they generalize in different ways depending on the type of shift (including sim-to-real)? Can we predict the OOD generalization of downstream agents from properties of the pre-trained representations such as the GS metrics from **Dittadi** et al. [3]?

5.3 Study Design

Robotic setup. Our setup is based on TriFinger [128] and consists of a robotic finger with three joints that can be controlled to manipulate an object (e.g. a cube) in a bowl-shaped stage. The agent receives a camera observation consistent with the images in [3] and outputs a three-dimensional action. During training, which always happens in simulation, the agent only observes a

[128]: Wüthrich et al. (2021), 'TriFinger: An Open-Source Robot for Learning Dexterity'

cube of four possible colors, randomly sampled at every episode (see Figure 5.1, step 2).

Distribution shifts. After training, we evaluate these agents in 7 environments: (1) the training environment, which is the typical setting in RL, (2) the OOD1 setting with cube colors that are OOD for the agent but still in-distribution for the encoder, (3) the more challenging OOD2-sim setting where the colors are also OOD for the encoder, (4-6) the OOD2 settings where the object colors are as in the 3 previous settings but the cube is replaced by a sphere (a previously unseen shape), (7) the OOD2-real setting, where we evaluate zero-shot sim-to-real transfer on the real robotic platform.

Tasks. We begin our study with the *object reaching* downstream control task, where the agent has to reach an object placed at an *arbitrary* random position in the arena. This is significantly more challenging than directly predicting the ground-truth factors, as the agent has to learn to reach the cube by acting on the joints, with a scalar reward as the only learning signal. Consequently, the compute required to learn this task is about 1,000 times greater than in the simple factor prediction case. We additionally include in our study a *pushing* task which consists of pushing an object to a goal position that is sampled at each episode. Learning this task takes one order of magnitude more compute than *object reaching*, likely due to the complex rigid-body dynamics and object interactions. To the best of our knowledge, this is the most challenging manipulation task that is currently feasible on our setup. **Ahmed** et al. [2] report solving a similar pushing task, but require the full ground-truth state to be observable.

Training the RL agents. The inputs at time t are the camera observation o_t and a vector of observable variables x_t containing the joint angles and velocities, as well as the target object position in *pushing*. We then feed the camera observation o_t into an encoder e that was pre-trained on the dataset in [3]. The result is concatenated with x_t, yielding a state vector $s_t = [x_t, e(o_t)]$. We then use SAC to train the policy with s_t as input. The policy, value, and Q networks are implemented as MLPs with 2 hidden layers of size 256. When training the policies, we keep the encoder frozen.

Model sweep. To shed light on the research questions outlined in the previous sections, we perform a large-scale study in which we train 240 representation models and 11,520 downstream policies,

as described below. See Appendix E.1 for further implementation details.

- We train 120 β-VAEs [57] and 120 Ada-GVAEs [67] with a subset of the hyperparameter configurations and neural architecture from **Dittadi** et al. [3]. Specifically, we consider $\beta \in \{1, 2, 4\}$, β annealing over $\{0, 50000\}$ steps, with and without input noise, and 10 random seeds per configuration. The latent space size is fixed to 10 following prior work [1, 59, 60, 67].
- For *object reaching*, we train 20 downstream policies (varying random seed) for each of the 240 VAEs. The resulting 4,800 policies are trained for 400k steps (approximately 2,400 episodes).
- Since *pushing* takes substantially longer to train, we limit the number of policies trained on this task: We choose a subset of 96 VAEs corresponding to only 4 seeds, and then use 10 seeds per representation. The resulting 960 policies are trained for 3M steps (about 9,000 episodes).
- Finally, for both tasks we also investigate the role of regularization on the policy. More specifically, we repeat the two training sweeps from above (5,760 policies), with the difference that now the policies are trained with L1 regularization on the first layer.

[67]: Locatello et al. (2020), 'Weakly-Supervised Disentanglement Without Compromises'

Limitations of our study. Although we aim to provide a sound and extensive empirical study, such studies are inevitably computationally demanding. Thus, we found it necessary to make certain design choices. For each of these choices, we attempted to follow common practice, in order to maintain our study as relevant, general, and useful as possible. One such decision is that of focusing on autoencoder-based representations. To answer our questions on the effect of upstream representations on the generalization of downstream policies, we need a diverse range of representations. How these representations are obtained is not directly relevant to answer our research question. Following **Dittadi** et al. [3], we chose to focus on β-VAE and Ada-GVAE models, as they were shown to provide a broad set of representations, including fully disentangled ones. Although we conjecture that other classes of representation learning algorithms should generally reveal similar trends as those found in our study, this is undoubtedly an interesting extension. As for the RL algorithm used in this work, SAC is known to be a particularly sample-efficient model-free RL method that is a popular choice in robotics [214–216]. Extensive results on pushing from ground-truth features on the same setup in [2] indicate that methods like TD3 [156] or PPO [155] perform very similarly to SAC under the same reward structure and observation

[214]: Haarnoja et al. (2018), 'Learning to walk via deep reinforcement learning'

[215]: Kiran et al. (2021), 'Deep reinforcement learning for autonomous driving: A survey'

[216]: Singh et al. (2019), 'End-to-end robotic reinforcement learning without reward engineering'

[156]: Fujimoto et al. (2018), 'Addressing function approximation error in actor-critic methods'

[155]: Schulman et al. (2017), 'Proximal policy optimization algorithms'

space. Thus, we expect the results of our study to hold beyond SAC. Another interesting direction is the study of additional regularization schemes on the policy network, an aspect that is often overlooked in RL. We expect the potential insights from extending the study along these axes to not justify the additional compute costs and corresponding carbon footprint. However, with improving efficiency and decreasing costs, we believe these could become worthwhile validation experiments in the future.

5.4 Results

We discuss our results in three parts: In Section 5.4.1, we present the training results of our large-scale sweep, and how policy regularization and different properties of the pre-trained representations affect in-distribution reward. Section 5.4.2 gives an extensive account of which metrics of the pre-trained representations predict OOD generalization of the agents in simulated environments. Finally, in Section 5.4.3 we perform a similar evaluation on the real robot, in a zero-shot sim-to-real scenario.

5.4.1 Results in the training environment

Figure 5.2 shows the training curves of all policies for *object reaching* and *pushing* in terms of the task-specific success metric. Here we use success metrics for interpretability, as their range is always $[0, 1]$. In *object reaching*, the success metric indicates progress from the initial end effector position to the optimal distance from the center of the cube. It is 0 if the final distance is not smaller than the initial distance, and 1 if the end effector is touching the center of a face of the cube. In *pushing*, the success metric is defined as the volumetric overlap of the cube with the goal cube, and the task can be visually considered solved with a score around 80%.

From the training curves we can conclude that both tasks can be consistently solved from pixels using pre-trained representations. In particular, all policies on *object reaching* attain almost perfect scores. Unsurprisingly, the more complex *pushing* task requires significantly more training, and the variance across policies is larger. Nonetheless, almost all policies learn to solve the task satisfactorily.

To investigate the effect of representations on the training reward, we now compute its Spearman rank correlations with various supervised and unsupervised metrics of the representations (Figure 5.2 bottom). By training reward, we mean the average reward of a fully trained policy over 200 episodes in the training environment (see

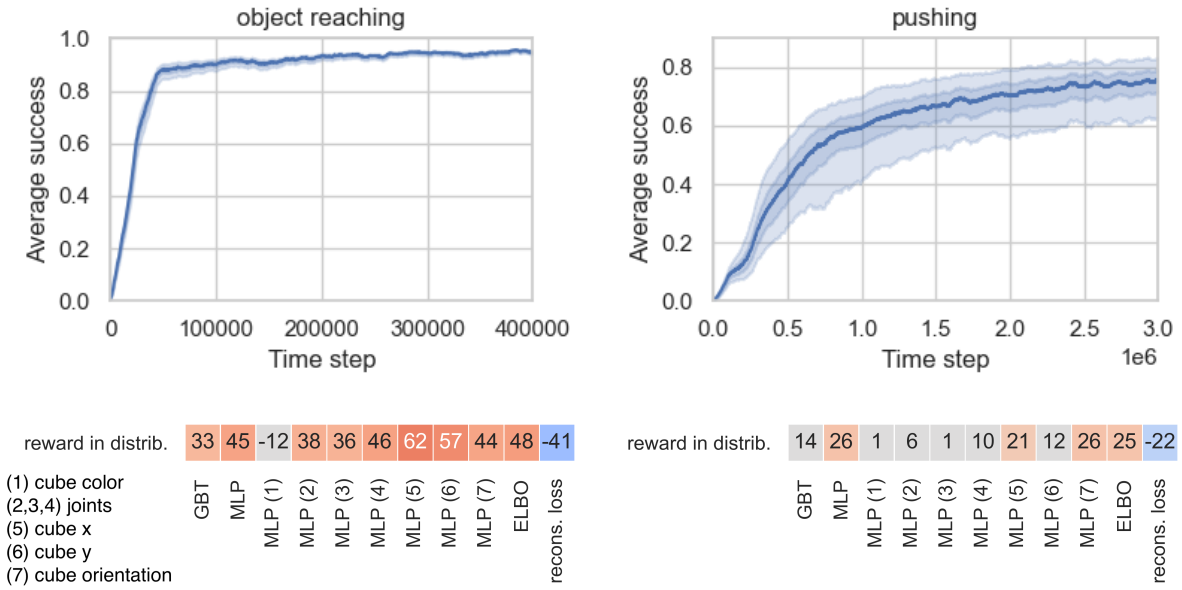


Figure 5.2: Top: Average training success, aggregated over *all* policies from the sweep (median, quartiles, 5th/95th percentiles). Bottom: Rank correlations between representation metrics and in-distribution reward (evaluated when the policies are fully trained), in the case without regularization. Correlations are color-coded in red (positive) or blue (negative) when statistically significant ($p < 0.05$), otherwise they are gray.

Appendix E.1). On *object reaching*, the final reward correlates with the ELBO and the reconstruction loss. A simple supervised metric to evaluate a representation is how well a small downstream model can predict the ground-truth factors of variation. Following [3], we use the MLP10000 and GBT10000 metrics (simply MLP and GBT in the following), where MLPs and Gradient Boosted Trees (GBTs) are trained to predict the FoVs from 10,000 samples. The training reward correlates with these metrics as well, especially with the MLP accuracy. This is not entirely surprising: if an MLP can predict the FoVs from the representations, our policies using the same architecture could in principle retrieve the FoVs relevant for the task. Interestingly, the correlation with the overall MLP metric mostly stems from the cube pose FoVs, i.e. those that are not included in the ground-truth state x_t. These results suggest that these metrics can be used to select good representations for downstream RL. On the more challenging task of *pushing*, the correlations are milder but most of them are still statistically significant.

[3]: **Dittadi** et al. (2021), 'On the Transfer of Disentangled Representations in Realistic Settings'

Summary. Both tasks can be consistently solved from pixels using pre-trained representations. Unsupervised (ELBO, reconstruction loss) and supervised (ground-truth factor prediction) in-distribution metrics of the representations are correlated with reward in the training environment.

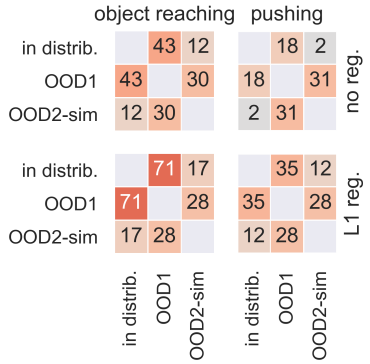


Figure 5.3: Correlations between training (in distrib.) and OOD rewards ($p<0.05$).

5.4.2 Out-of-distribution generalization in simulation

In-distribution and out-of-distribution rewards. After training, the in-distribution reward correlates with OOD1 performance on both tasks (especially with regularization), but not with OOD2 performance (see Figure 5.3). Moreover, rewards in OOD1 and OOD2 environments are moderately correlated across tasks and regularization settings.

Unsupervised metrics and informativeness. In Figure 5.4, we assess the relation between OOD reward and in-distribution metrics (ELBO, reconstruction loss, MLP, and GBT). Both ELBO and reconstruction loss exhibit a correlation with OOD1 reward, but not with OOD2 reward. These unsupervised metrics can thus be useful for selecting representations that will lead to more robust downstream RL tasks, as long as the encoder is in-distribution. While the GBT score is not correlated with reward under distribution shift, we observe a significant correlation between OOD1 reward and the MLP score, which measures downstream factor prediction accuracy of an MLP with the same architecture as the one parameterizing the policies. As in Section 5.4.1, we further investigate the source of this correlation, and find it in the pose parameters of the cube. Correlations in the OOD2 setting are much weaker, thus we conclude that these metrics do not appear helpful for model selection in this case. Our results on *pushing* confirm these conclusions although correlations are generally weaker, presumably due to the more complicated nature of this task. An extensive discussion is provided in Appendix E.2.2.

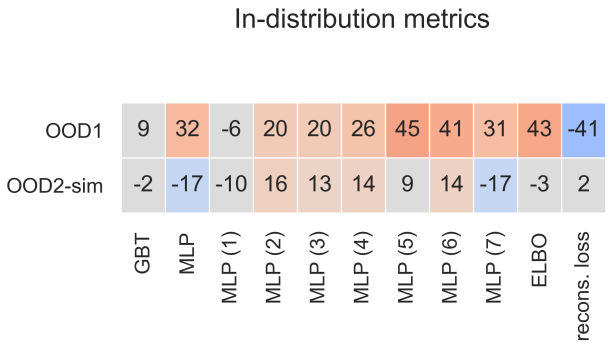


Figure 5.4: Rank correlations of representation properties with OOD1 and OOD2 reward on *object reaching* without regularization. Numbering when splitting metrics by FoV: (1) cube color; (2–4) joint angles; (5–7) cube position and rotation. Correlations are color-coded as described in Figure 5.2.

Correlations with generalization scores. Next, we analyze the link between generalization in RL and the generalization scores (GS) discussed in Section 5.2, which measure the generalization of downstream FoV predictors *out of distribution*, as opposed to the MLP and GBT metrics considered above. For both OOD scenarios, the distribution shifts underlying these GS scores are the same as the ones in the RL tasks in simulation. We summarize our

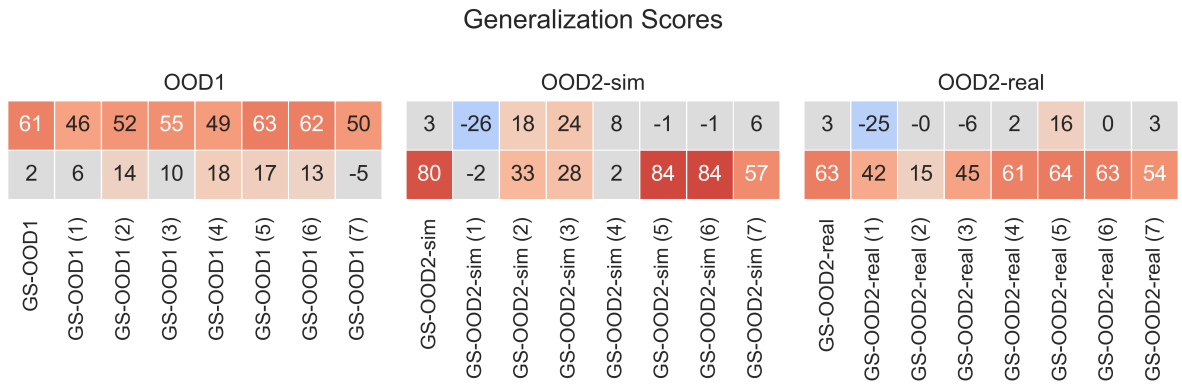


Figure 5.5: Rank correlations of generalization scores with OOD1 and OOD2 reward on *object reaching* without regularization. Numbering when splitting metrics by FoV: (1) cube color; (2–4) joint angles; (5–7) cube position and rotation. Correlations are color-coded as described in Figure 5.2.

findings in Figure 5.5 on the *object reaching* task. Reward in the OOD1 setting is significantly correlated with the GS-OOD1 metric of the pre-trained representation. We observe an even stronger correlation between the reward in the simulated OOD2 setting and the corresponding GS-OOD2-sim and GS-OOD2-real scores. On a per-factor level, we see that the source of the observed correlations primarily stems from the generalization scores w.r.t. the pose parameters of the cube. The OOD generalization metrics can therefore be used as proxies for the corresponding form of generalization in downstream RL tasks. This has practical implications for the training of RL downstream policies which are generally known to be brittle to distribution shifts, as we can measure a representation's generalization score from a few labeled images. This allows for selecting representations that yield more robust downstream policies.

Disentangled representations. Disentanglement has been shown to be helpful for downstream performance and OOD1 generalization even with MLPs [3]. However, in *object reaching*, we only observe a weak correlation with some disentanglement metrics (Figure 5.6). In agreement with **Dittadi** et al. [3], disentanglement does not correlate with OOD2 generalization. The same study observed that disentanglement correlates with the informativeness of a representation. To understand if these weak correlations originate from this common confounder, we investigate whether they persist after adjusting for MLP FoV prediction accuracy. Given two representations with similar MLP accuracy, does the more disentangled one exhibit better OOD1 generalization? To measure this we predict success from the MLP accuracy using kNN (k=5) [70] and compute the residual reward by subtracting the amount of reward explained by the MLP metric. Figure 5.6 shows that this resolves the remaining correlations with disentanglement. Thus,

[70]: Locatello et al. (2019), 'On the fairness of disentangled representations'

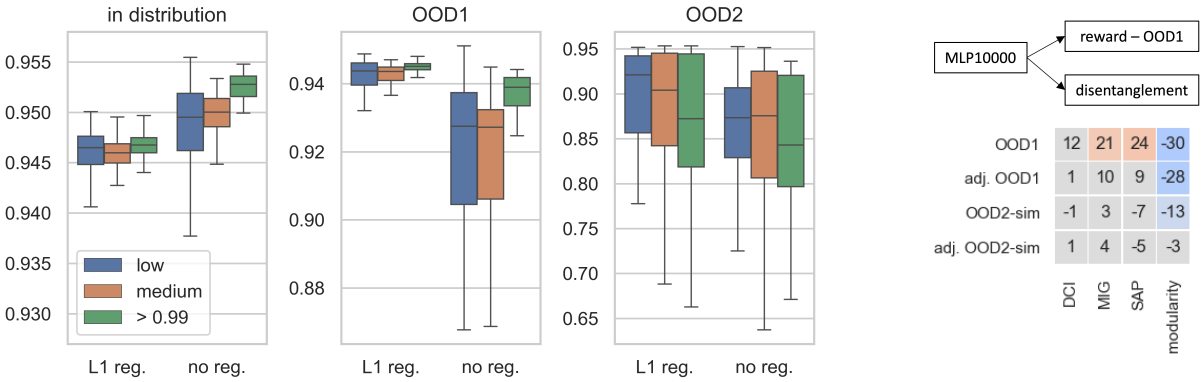


Figure 5.6: Box plots: Fractional success on *object reaching* split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. L1 regularization in the first layer of the MLP policy has a positive effect on OOD1 and OOD2 generalization with minimal sacrifice in terms of training reward (see scale). **Correlation matrix**: Although we observe a mild correlation between some disentanglement metrics and OOD1 (but not OOD2) generalization, this does not hold when adjusting for representation informativeness. Correlations are color-coded as described in Figure 5.2. We use disentanglement metrics from [60, 61, 84, 160].

for the RL downstream tasks considered here, disentanglement per se does not seem to be useful for OOD generalization. We present similar results on *pushing* in Appendix E.2.2.

Policy regularization and observation noise. It might seem unsurprising that disentanglement is not useful for generalization in RL, as MLP policies do not have any explicit inductive bias to exploit it. Thus, we attempt to introduce such inductive bias by repeating all experiments with L1 regularization on the first layer of the policy. Although regularization improves OOD1 and OOD2 generalization in general (see box plots in Figure 5.6), we observe no clear link with disentanglement. Furthermore, in accordance with **Dittadi** et al. [3], we find that observation noise when training representations is beneficial for OOD2 generalization. See Appendix E.2.2 for a detailed discussion.

Stronger OOD shifts: evaluating on a novel shape. On *object reaching*, we also test generalization w.r.t. a novel shape by replacing the cube with a sphere. This corresponds to a strong OOD2-type shift, since shape was never varied when training the representations. Surprisingly, the policies appear to be robust to the novel shape. In fact, when the sphere has the same colors that the cube had during policy training, *all* policies get closer than 5 cm to the sphere on average, with a mean success metric of 95%. On sphere colors from the OOD1 split, more than 98.5% move the finger closer than this threshold, and on the strongest distribution shift (OOD2-sim colors, and cube replaced by sphere) almost 70% surpass that threshold with an average success metric above 80%.

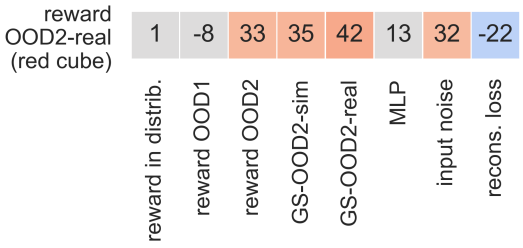


Figure 5.7: Rank-correlations of various representation metrics with the zero-shot sim-to-real OOD2 reward for the task of *object reaching* on the real platform with a red cube (color-coded as described in Figure 5.2).

Summary. (1) In- and out-of-distribution rewards are correlated, as long as the representation remains in its training distribution (OOD1). (2) Similarly, in-distribution representation metrics (both unsupervised and supervised) predict OOD1 reward, but are not reliable when the representation is OOD (OOD2). (3) Disentanglement does not correlate with generalization in our experiments, while (4) input noise when training representations is beneficial for OOD2 generalization. (5) Most notably, the GS metrics, which measure generalization under distribution shifts, are significantly correlated with RL performance under similar distribution shifts. We thus recommend using these convenient proxy metrics for selecting representations that will yield robust downstream policies.

5.4.3 Deploying policies to the real world

We now evaluate a large subset of the agents on the real robot without fine-tuning, quantify their zero-shot sim-to-real generalization, and find metrics that correlate with real-world performance.

Reaching. We choose 960 policies trained in simulation, based on 96 representations and 10 random seeds, and evaluate them on two (randomly chosen, but far apart) goal positions using a red cube. While a red cube was in the training distribution, we consider this to be OOD2 because real-world images represent a strong distribution shift for the encoder [3, 181]. Although sim-to-real in robotics is considered to be very challenging without domain randomization or fine-tuning [171, 172, 217], many of our policies obtain a high fractional success without resorting to these methods. In addition, in Figure 5.7 we observe significant correlations between zero-shot real-world performance and some of the previously discussed metrics. First, there is a positive correlation with the OOD2-sim reward: Policies that generalize to unseen cube colors in simulation also generalize to the real world. Second, representations with high GS-OOD2-sim and (especially) GS-OOD2-real scores are promising candidates for sim-to-real transfer.

[171]: Tobin et al. (2017), 'Domain randomization for transferring deep neural networks from simulation to the real world'
[172]: Rusu et al. (2017), 'Sim-to-real robot learning from pixels with progressive nets'
[217]: Finn et al. (2017), 'Generalizing skills with semi-supervised reinforcement learning'

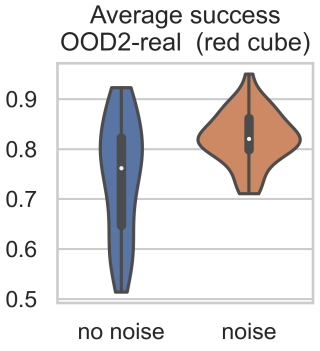


Figure 5.8: Training encoders with additive noise improves zero-shot sim-to-real generalization on *object reaching*

.

Third, if no labels are available, the weaker correlation with the reconstruction loss on the simulated images can be exploited for representation selection. Finally, as observed by **Dittadi** et al. [3] for simple downstream tasks, input noise while learning representations is beneficial for sim-to-real generalization (Figure 5.8).

Based on these findings, we select 50 policies with a high GS-OOD2-real score, and evaluate them on the real world with a green and a blue cube, which is an even stronger OOD2 distribution shift.

In Figure 5.9, where metrics are averaged over 4 cube positions per policy, we observe that most policies can still solve the task: approximately 80% of them position the finger less than 5 cm from the cube. Lastly, we repeat the evaluations on the green sphere that we previously performed in simulation, and observe that many policies successfully reach this completely novel object. See Appendix E.2.3 and the project website for additional results and videos of deployed policies.

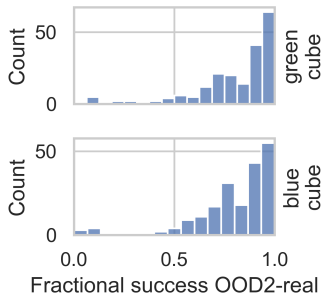


Figure 5.9: Zero-shot sim-to-real on *object reaching* with histograms of fractional success in the more challenging OOD2-real-{green,blue} scenario from 50 policies across 4 different goal positions.

Pushing. We now test whether our real-world findings on *object reaching* also hold for *pushing*. We again select policies with a high GS-OOD2-real score and encoders trained with input noise. We record episodes on diverse goal positions and cube colors to support our finding that pushing policies in simulation can generalize to the real robot. In Figure 5.10, we show three representative episodes with successful task completions.

Summary. Policies trained in simulation can solve the task on the real robot without domain randomization or fine-tuning. Reconstruction loss, encoder robustness, and OOD2 reward in simulation are all good predictors of real-world performance. For real-world applications, we recommend using GS-OOD2-sim or GS-OOD2-real for model selection, and training the encoder with noise.

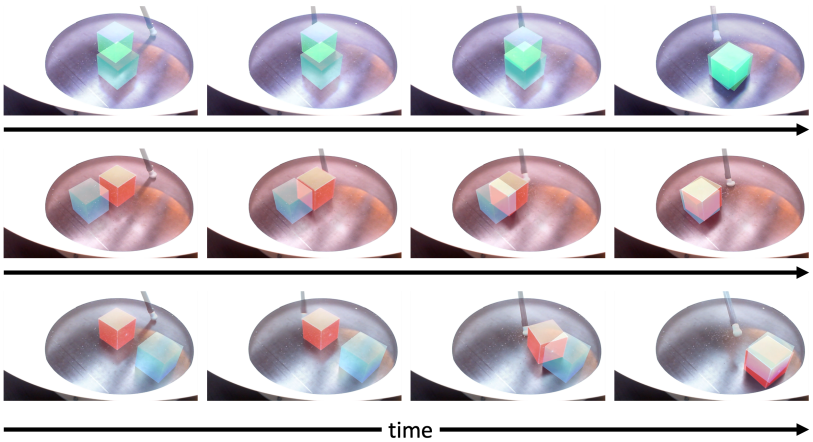


Figure 5.10: We select pushing policies with high GS-OOD2-real score. When deployed on the real robot without fine-tuning, they succeed in pushing the cube to a specified goal position (transparent blue cube).

5.5 Other Related Work

A key unsolved challenge in RL is the brittleness of agents to distribution shifts in the environment, even if the underlying structure is largely unchanged [2, 118]. This is related to studies on representation learning and generalization in downstream tasks [93, 218–221], as well as domain generalization (see [222] for an overview). More specifically for RL, Higgins et al. [71] focus on domain adaptation and zero-shot transfer in DeepMind Lab and MuJoCo environments, and claim disentanglement improves robustness. To obtain better transfer capabilities, Asadi, Abel, and Littman [223] argue for discretizing the state space in continuous control domains by clustering states where the optimal policy is similar. Kulkarni et al. [224] propose geometric object representations by means of keypoints or image-space coordinates and Wulfmeier et al. [225] investigate the effect of different representations on the learning and exploration of different robotics tasks. Transfer becomes especially challenging from the simulation to the real world, a phenomenon often referred to as the sim-to-real gap. This is particularly crucial in RL, as real-world training is expensive, requires sample-efficient methods, and is sometimes unfeasible if the reward structure requires accurate ground truth labels [226, 227]. This issue is typically tackled with large-scale domain randomization in simulation [112, 174].

[2]: **Ahmed** et al. (2021), 'Causal-world: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning'

[118]: Cobbe et al. (2019), 'Quantifying generalization in reinforcement learning'

[93]: Gondal et al. (2019), 'On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset'

[218]: Steenbrugge et al. (2018), 'Improving Generalization for Abstract Reasoning Tasks Using Disentangled Feature Representations'

[219]: Dittadi et al. (2022), 'Generalization and Robustness Implications in Object-Centric Learning'

[220]: Esmaeili et al. (2019), 'Structured disentangled representations'

[221]: Chaabouni et al. (2020), 'Compositionality and Generalization in Emergent Languages'

[222]: Wang et al. (2022), 'Generalizing to unseen domains: A survey on domain generalization'

[226]: Dulac-Arnold et al. (2019), 'Challenges of real-world reinforcement learning'

[227]: Kormushev et al. (2013), 'Reinforcement learning in robotics: Applications and real-world challenges'

[112]: Akkaya et al. (2019), 'Solving rubik's cube with a robot hand'

[174]: James et al. (2019), 'Sim-to-real via sim-to-sim: Data-efficient robotic grasping via random-ized-to-canonical adaptation networks'

5.6 Conclusion

Robust out-of-distribution (OOD) generalization is still one of the key open challenges in machine learning. We attempted to answer central questions on the generalization of reinforcement learning agents in a robotics context, and how this is affected by pre-trained representations. We presented a large-scale empirical study in which we trained over 10,000 downstream agents given pre-trained representations, and extensively tested them under a variety of distribution shifts, including sim-to-real. We observed agents that generalize OOD, and found that some properties of the pre-trained representations can be useful to predict which agents will generalize better. We believe this work brings us one step closer to understanding the generalization abilities of learning systems, and we hope that it encourages many further important studies in this direction.

Discrete Key-Value Bottleneck | 6

Abstract

Deep neural networks perform well on classification tasks where data streams are i.i.d. and labeled data is abundant. Challenges emerge with non-stationary training data streams such as continual learning. One powerful approach that has addressed this challenge involves pre-training of large encoders on volumes of readily available data, followed by task-specific tuning. Given a new task, however, updating the weights of these encoders is challenging as a large number of weights needs to be fine-tuned, and as a result, they forget information about the previous tasks. In the present work, we propose a model architecture to address this issue, building upon a discrete bottleneck containing pairs of separate and learnable key-value codes. Our paradigm will be to *encode; process the representation via a discrete bottleneck; and decode*. Here, the input is fed to the pre-trained encoder, the output of the encoder is used to select the nearest keys, and the corresponding values are fed to the decoder to solve the current task. The model can only fetch and re-use a sparse number of these key-value pairs during inference, enabling *localized and context-dependent model updates*. We theoretically investigate the ability of the discrete key-value bottleneck to minimize the effect of learning under distribution shifts and show that it reduces the complexity of the hypothesis class. We empirically verify the proposed method under challenging class-incremental learning scenarios and show that the proposed model — without any task boundaries — reduces catastrophic forgetting across a wide variety of pre-trained models, outperforming relevant baselines on this task.

6.1 Introduction 83
6.2 Encode, Processing via Discrete Key-Value Bottleneck, Decode . . . 85
6.3 Theoretical Analysis . . 86
6.4 Related Work 89
6.5 Experiments 90
6.6 Limitations and Extensions 96
6.7 Conclusion 97

About This Chapter

This chapter has been adapted from **Träuble** et al. [5].

Discrete Key-Value Bottleneck

Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael Curtis Mozer, Kenji Kawaguchi, Yoshua Bengio and Bernhard Schölkopf.

ICML 2023: International Conference on Machine Learning.

Author Contributions based on CRediT (Contributor Roles Taxonomy)

Frederik Träuble: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration **Anirudh Goyal**: Conceptualization, Methodology, Investigation, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration **Nasim Rahaman**: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization **Michael Mozer**: Writing - Review & Editing, Visualization **Kenji Kawaguchi**: Formal analysis, Writing - Review & Editing **Yoshua Bengio**: Methodology, Writing - Review & Editing, Supervision, Resources, Funding acquisition **Bernhard Schölkopf**: Resources, Validation, Formal analysis, Writing - Review & Editing, Supervision, Funding acquisition

6.1 Introduction

Current neural networks achieve state-of-the-art results across a range of challenging problem domains. Most of these advances are limited to learning scenarios where the training data is sampled i.i.d. from some assumed joint distribution $P(X, Y) = P(Y|X)P(X)$ in the supervised case. In reality, learning scenarios are often far from i.i.d. and exhibit various distribution shifts; for instance during continual learning, when the training distribution changes over time.

Despite all successes, the training of deep neural networks on non-stationary training data streams remains challenging. For instance, training neural networks without additional modifications on some small dataset or input distribution may quickly lead to over-fitting and catastrophic forgetting of previous knowledge [228–230]. One paradigm that has dominated in recent work is the use of models pre-trained on large amounts of data e.g. in a self-supervised manner [32, 231–233]. These models are then re-purposed for smaller datasets via either (a) fine-tuning, which may involve changing large numbers of parameters, or (b) introducing a small number of new parameters to adapt the model [234–236].

[228]: Chen et al. (2018), 'Lifelong machine learning'
[229]: Thrun (1995), 'Is learning the n-th thing any easier than learning the first?'
[230]: Van de Ven et al. (2019), 'Three scenarios for continual learning'
[32]: Brown et al. (2020), 'Language models are few-shot learners'
[231]: Chen et al. (2020), 'Big self-supervised models are strong semi-supervised learners'
[232]: Azizi et al. (2021), 'Big self-supervised models advance medical image classification'
[233]: Caron et al. (2021), 'Emerging Properties in Self-Supervised Vision Transformers'
[234]: Houlsby et al. (2019), 'Parameter-efficient transfer learning for NLP'
[235]: Zhou et al. (2022), 'Learning to prompt for vision-language models'
[236]: Wang et al. (2022), 'Learning to prompt for continual learning'

Building upon this pre-training paradigm, we introduce an approach (see Figure 6.1 for an overview of the model architecture) to distil information into a discrete set of code pairs, each consisting of a coupled key and value code. In a nutshell, we follow a three-step process: *encode, process via a discrete bottleneck, and decode*.

In the first step, an input is fed to the encoder to obtain a continuous-valued representation that is projected into C lower-dimensional heads. In the second step, these heads are used to search for the closest key within head-specific key-value codebooks, and *corresponding* continuously learnable value codes taken from a discrete set are fetched. In the third step, these value codes are passed through a downstream decoder to produce the final output. By freezing all components except for the value codes, the model can continue to improve its performance *locally* on input-dependent value codes without affecting the prediction and key-value pairs of prior training samples.

As we will show theoretically and empirically, this architecture offers an appealing ability to improve generalization under input distribution shifts (also called *covariate shifts*) without common vulnerabilities to non-i.i.d. training distributions. We will show that we can mitigate forgetting via localization because only fetched value codes are being updated. Theoretically (see Section 6.3), this is enabled via the following two mechanisms: first, having these intermediate discrete codes minimizes the nonstationarity of the

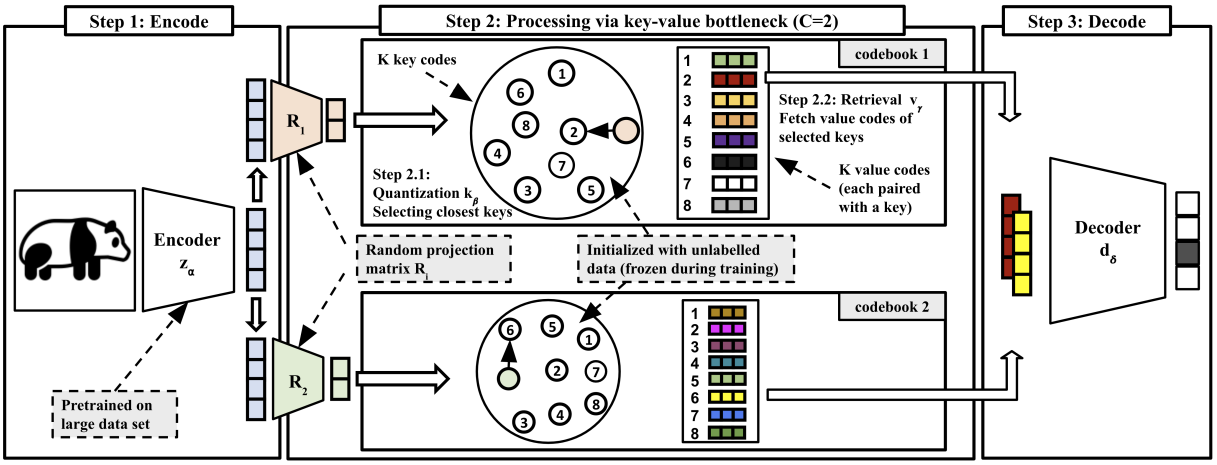


Figure 6.1: Overview of discrete key-value bottleneck. First, a pre-trained encoder maps the high-dimensional input to a lower-dimensional representation. Second, this representation is projected into C lower-dimensional heads. $C = 2$ is chosen for illustrative purposes. Each head is processed as input by one out of C separate key-value codebooks: In the codebook, the head is discretized by snapping to the closest key code, and the corresponding value code for that key is fetched. Third, the values from different key-value codebooks are combined and fed as input to the decoder for the final prediction. Keys and values can be of different dimensions in the proposed model.

input distribution shift by keeping the joint features fixed; and second, the discrete bottleneck will reduce the complexity of the hypothesis class through its separate key-value codes.

The fundamental difference to prior work on discrete bottlenecks [237–239] is the introduction of the discrete pairs with two different codes, the first (key codes) being optimized for the encoder and the second (value codes) being optimized for the decoder. This means that the gradients from a particular downstream task will not directly affect the key codes, which induces a rather different learning behaviour compared to standard neural networks.

[237]: Van Den Oord et al. (2017), 'Neural discrete representation learning'
[238]: Razavi et al. (2019), 'Generating diverse high-fidelity images with vq-vae-2'
[239]: Liu et al. (2021), 'Discrete-valued neural communication'

Contributions: We introduce a model architecture designed for learning under input distribution changes during training. We corroborate this theoretically by proving that under input distribution shifts, the proposed model achieves an architecture-dependent generalization bound that is better than that of models without the discrete bottleneck. We empirically verified the method under a challenging class-incremental learning scenario on real-world data and show that the proposed model — without any task boundaries — reduces the common vulnerability to catastrophic forgetting across a wide variety of publicly available pre-trained models and outperforms relevant baselines on this task.

6.2 Encode, Processing via Discrete Key-Value Bottleneck, Decode

This section introduces the proposed model architecture. An overview of the workings of the model is given in Figure 6.1. Our goal is to learn a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ from training data $S = ((x_i, y_i))_{i=1}^n$ that is robust to strong input distribution changes. We consider a model that can be decomposed as

$$f_\theta(x) = (d_\delta \circ v_\gamma \circ k_\beta \circ z_\alpha)(x) \tag{6.1}$$

with $\theta = (\alpha, \beta, \gamma, \delta)$. In the first step, the encoder $z_\alpha : \mathcal{X} \to \mathcal{Z} \in \mathbb{R}^{m_z}$ extracts a representation from the high-dimensional observation x. We further project this representation into C lower-dimensional feature heads, each of them being passed as the input into a separate head-specific learnable key-value codebook. As the name suggests, a key-value codebook is a bijection that maps each key vector to its value (where the latter is learned). Within each codebook, a quantization process k_β selects the closest key to its input head (think of it as a "feature anchor") from a head-specific key-value codebook. Next, a corresponding value vector is being fetched and returned (one per codebook), which we denote via v_γ. The returned values across all codebooks finally serve as input to a decoder d_δ for prediction. In the following, each component is motivated in detail.

Step 1: Encoding. The encoder z_α projects the input x into a lower-dimensional vector $z \in \mathbb{R}^{m_z}$, further down-projected into C separate heads of dimension d_{key} using C fixed Gaussian random projection matrices. For tasks where we judge x to be sufficiently low-dimensional, we skip encoding and partition x directly. For the purpose of this work we take a broad spectrum of differently pre-trained encoders [25, 29, 233, 240–242], and show that they are all capable of embedding features that are relevant for the studied task.

[25]: He et al. (2016), 'Deep residual learning for image recognition'
[29]: Dosovitskiy et al. (2020), 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'
[233]: Caron et al. (2021), 'Emerging Properties in Self-Supervised Vision Transformers'
[240]: Caron et al. (2020), 'Unsupervised learning of visual features by contrasting cluster assignments'
[241]: Trockman et al. (2023), 'Patches Are All You Need?'
[242]: Radford et al. (2021), 'Learning transferable visual models from natural language supervision'

Step 2: Processing via Discrete Key-Value Bottleneck.

> **Step 2.1: Quantization.** A quantization process snaps each head to the closest key from the corresponding head's codebook, based on the head's content. Closeness is defined by the L2 distance between the head and key vectors.
> **Step 2.2: Retrieval.** For each head, a simple lookup in the head-specific codebook is performed to retrieve the value corresponding to the key selected in the previous step.

Overall, the bottleneck will retrieve a set of C value codes.

Step 3: Decoding. Finally, we predict the target variable from this set of fetched values using any suitable decoder function d_δ. For

the purpose of our experiments on classification, we can apply a simple non-parametric decoder function which uses average pooling to calculate the element-wise average of all the fetched value codes and then applies a softmax function to the output.

Initializing the Model. During training, we first initialize the key codes. The non-parametric 1-to-1 mapping between key and value codes means no gradient back-propagates from the values to the keys. Instead, they are solely determined from the encoding of the input observation, implying that no supervised data is required to learn them. A simple and effective way to initialize the keys is via exponential moving average (EMA) updates as proposed for VQ-VAE in [237] and used for VQ-VAE2 in [238]. This procedure allows us to obtain keys that are broadly distributed in the feature space of the encoder, without introducing additional parameters. We refer to Appendices F.3 and F.5.3 and Section 6.5.3 for more details on the implementation and robustness of this procedure. Once initialized, the keys are frozen and thus not influenced by the task on the target domain, implying that any given input x is mapped to keys from a fixed set.

[237]: Van Den Oord et al. (2017), 'Neural discrete representation learning'

[238]: Razavi et al. (2019), 'Generating diverse high-fidelity images with vq-vae-2'

Training under Dataset Shifts. Assume we are now given training data $S = ((x_i, y_i))_{i=1}^n$, but the data generation process might undergo various covariate shifts of $P(X)$ without knowing when they occur. A particularly challenging example of that would be the task of continual learning class-incrementally without having task boundaries, where the support of $P(X)$ changes dramatically. Having the encoder and decoder solely connected through the key-value codes, we can now train the model on this data stream by only updating the value codes under such an input distribution shift. Since we only locally update the actual retrieved values under the input distribution changes, the values from other data domains remain unchanged, thereby enabling *localized, context-dependent model updates*. In this way, we can integrate new data and thus gradually improve the models' performance. Moreover, as we will show theoretically, the proposed architecture benefits from the fact that the decoder works with a discrete set of value codes, as opposed to directly predicting from the encoder representation.

6.3 Theoretical Analysis

In this section, we theoretically investigate the behavior of the proposed method. In contrast to standard models, the proposed model is shown to have the ability to reduce the generalization error of transfer learning under input distribution shifts via two mechanisms: (1) the discretization with key-value pairs, mitigating the effect of the input distribution shift, and (2) the discrete

bottleneck with key-value pairs, reducing the complexity of the hypothesis class.

To formally state this, we first introduce the notation. We consider the distribution shift of x via an arbitrary function $(x, \epsilon) \mapsto g_\epsilon(x)$ with random vectors ϵ. In the target domain, we are given a training dataset $S = ((x_i, y_i))_{i=1}^n$ of n samples where $x_i \in \mathcal{X} \subseteq \mathbb{R}^{m_x}$ and $y_i \in \mathcal{Y} \subseteq \mathbb{R}^{m_y}$ are the i-th input and the i-th target respectively. The training loss per sample is defined by $\ell(f_\theta(x_i), y_i)$, where $\ell : \mathbb{R}^{m_y} \times \mathcal{Y} \to [0, M]$ is a bounded loss function. We compare the proposed model against an arbitrary model $\hat{f} : \mathcal{X} \to \mathbb{R}^{m_y}$ without the key-value discretization with a hypothesis class $\hat{\mathcal{F}} \ni \hat{f}$. Let f_θ^S (for the proposed model) and $\hat{f}^S$ (for an arbitrary model) be the corresponding hypotheses learned through the training set S. Let $\mathcal{K} = \{(k_\beta \circ z_\alpha)(x) : x \in \mathcal{X}\}$ be the set of keys. By ordering elements of $\mathcal{K}$, we use $\mathcal{K}_i$ to denote i-th element of $\mathcal{K}$. Let $\hat{d}$ be a distance function. We define $[n] = \{1, \ldots, n\}$, $\mathcal{C}_k = \{x \in \mathcal{X} : k = \operatorname{argmin}_{i \in [|\mathcal{K}|]} \hat{d}((k_\beta \circ z_\alpha)(x), \mathcal{K}_i)\}$ (the input region corresponding to the k-th key), and $\mathcal{I}_k^y = \{i \in [n] : x_i \in \mathcal{C}_k, y_i = y\}$ (the data indices corresponding to the k-th key and the label y). We define $\mathbb{1}\{a = b\} = 1$ if $a = b$ and $\mathbb{1}\{a = b\} = 0$ if $a \neq b$. We denote by $\mathcal{D}_{x|y}$ the conditional distribution of the input x given a label y, and by $\bar{S} = ((\bar{x}_i, \bar{y}_i))_{i=1}^n$ the random variable having the same distribution as S.

The following theorem (proof in Appendix F.1) shows that under input distribution shifts, a model with the proposed architecture has the ability to reduce the generalization error of the base model:

Theorem 6.3.1 *There exists a constant c (independent of $n, f, \hat{\mathcal{F}}, \delta, \epsilon$ and S) such that for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for any $f \in \{f_\theta^S, \hat{f}^S\}$:*

$$\begin{aligned}\mathbb{E}_{(x,y),\epsilon}[\ell(f(g_\epsilon(x)), y)] \leq \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i) \\ + c\sqrt{\frac{2\ln(2e/\delta)}{n}} \\ + \mathcal{B}_{g_\epsilon}(f) \\ + \mathbb{1}\{f = \hat{f}^S\}\mathcal{G}_{\hat{f}^S},\end{aligned}$$

where

$$\begin{aligned}\mathcal{B}_{g_\epsilon}(f) = \frac{1}{n}\sum_{y,k} |\mathcal{I}_k^y| \cdot \mathbb{E}_{x \sim \mathcal{D}_{x|y}, \epsilon}\left[\ell(f(g_\epsilon(x)), y)\right. \\ \left. - \ell(f(x), y) | x \in \mathcal{C}_k\right],\end{aligned}$$

and

$$\mathcal{G}_{\hat{f}^S} = \sum_{y,k} \frac{2\mathcal{R}_{y,k}(\ell \circ \hat{\mathcal{F}})}{n} + c\sqrt{\frac{\ln(2e/\delta)}{2n}}.$$

Here, with independent uniform random variables $\xi_1, \ldots, \xi_n$ taking values in $\{-1, 1\}$,

$$\mathcal{R}_{y,k}(\ell \circ \hat{\mathcal{F}}) = \mathbb{E}_{\bar{S},\xi}[\sup_{\hat{f} \in \hat{\mathcal{F}}} \sum_{i=1}^{|\mathcal{I}_k^y|} \xi_i \ell(\hat{f}(\bar{x}_i), \bar{y}_i) | \bar{x}_i \in \mathcal{C}_k, \bar{y}_i = y].$$

Moreover, if $x \in \mathcal{C}_k \Rightarrow g_\epsilon(x) \in \mathcal{C}_k$ with probability one, then

$$\mathcal{B}_{g_\epsilon}(f_\theta^S) = 0.$$

Implications for Empirical Results: The first benefit of the use of key-value pairs appears in the training loss $\frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i)$ in the right-hand side (RHS) of the generalization error bound in Theorem 6.3.1. Since the bound is independent of the complexity of v_γ, we can set the hypothesis class of the local values v_γ to be highly expressive to reduce the training loss for each local region of the input space (in order to minimize the RHS of the bound in Theorem 6.3.1). Indeed, Figure 6.2 demonstrates this (although the main purpose is to show another benefit in terms of localization and forgetting). Moreover, the benefit of the proposed architecture is captured by two further mechanisms: The first mechanism, the term $\mathcal{B}_{g_\epsilon}(f)$ in Theorem 6.3.1 measures the effect of the input distribution shift g_ϵ. Theorem 6.3.1 shows that if the key of x is still the key of $g_\epsilon(x)$ (i.e., $x \in \mathcal{C}_k \Rightarrow g_\epsilon(x) \in \mathcal{C}_k$) with probability one, then we can further reduce the error as $\mathcal{B}_{g_\epsilon}(f_\theta^S) = 0$ while $\mathcal{B}_{g_\epsilon}(\hat{f}^S) > 0$ in general. This captures the mechanism, which is the minimization of the effect of the input distribution shifts via the discretization with key-value pairs. The second mechanism is shown by the fact that a model with the proposed architecture reduces the bound by making the last term $\mathbb{1}\{f = \hat{f}^S\}\mathcal{G}_{\hat{f}^S}$ vanish, because $\mathbb{1}\{f = \hat{f}^S\} = 0$ for $f = f_\theta^S$. This removed term consists of the complexity of its hypothesis class $\mathcal{R}_{y,k}(\ell \circ \hat{\mathcal{F}})$. Thus, the proposed model avoids the complexity cost as compared to other architectures. This is because the *bottleneck* with key-value pairs prevents over-fitting. Finally, the benefit from random projections in terms of Theorem 6.3.1 is that this projection can reduce the term of $\mathcal{B}_{g_\epsilon}(f)$ in addition to the flexibility in reducing the training loss term when having multiple codebooks as further discussed in Appendix F.2.

6.4 Related Work

The hallmark of machine learning is to be able to develop models that can quickly adapt to new tasks once trained on sufficiently diverse tasks [243, 244]. Multiple ways to transfer information from one task to another exist: (1) transfer information via the transfer of the neural network weights (when trained on source tasks); (2) reuse raw data as in retrieval-based methods [245–251]; or (3), via knowledge distillation [252]. Each approach implies inevitable trade-offs: When directly transferring neural network weights, previous information about the data may be lost in the fine-tuning process, while transfer via raw data may be prohibitively expensive as there can be hundreds of thousands of past experiences. At the same time, we want models that can continually accumulate information without training from scratch and forgetting previous information [228, 229, 253, 254]. To prevent models from forgetting previous information, continual learning approaches make use of replay-buffers to replay old information [255–258], regularization during optimization [259, 260], incrementally adding task-specific model components to increase capacity [261], using parameter isolation-based approaches [262–265], or meta-learning fast-adapting models to promote quick adaptation [266–268]. The proposed method can be mostly linked with parameter isolation-based approaches by isolating the key-value pair for different tasks and classes. This bears some similarity with the theory of Sparse Distributed Memory (SDM), which is a model of associative memory inspired by human long-term memory and connections to neural circuits found in various organisms [269, 270]. The key-value bottleneck method has some similarities with the concurrent work on SDMLPs' by Bricken et al. [271] which builds upon SDM ideas, but there are several crucial differences. While this model also writes and reads from a fixed set of "neurons", the proposed method does not require the implementation and careful tuning of additional neurobiology-inspired model components, such as "GABA switch implementations" to avoid "dead-neuron problems". Furthermore, key-value bottlenecks benefit from handling high-dimensional embedding spaces through random down-projection matrices and multiple distributed codebooks, while maintaining theoretical guarantees on catastrophic forgetting, thereby performing better on the presented task. The benefit of random down-projections is further supported by recent work exploring the effect of orthogonal projection-based methods as weight regularization to avoid forgetting [272, 273]. Other models such as Shen, Dasgupta, and Navlakha [274] and Pourcel, Vu, and French [275] focus on the online-learning setting without being exposed to the same classes for thousands of epochs. Most approaches for continual learning require the provision of task boundaries, but the proposed model

[243]: Baxter (2000), 'A model of inductive bias learning'
[244]: Thrun et al. (2012), *Learning to learn*
[245]: Borgeaud et al. (2022), 'Improving language models by retrieving from trillions of tokens'
[246]: Nakano et al. (2021), 'WebGPT: Browser-assisted question-answering with human feedback'
[247]: Lee et al. (2019), 'Latent retrieval for weakly supervised open domain question answering'
[248]: Lewis et al. (2020), 'Retrieval-augmented generation for knowledge-intensive nlp tasks'
[249]: Guu et al. (2020), 'Retrieval augmented language model pre-training'
[250]: Sun et al. (2021), 'Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation'
[251]: Goyal et al. (2022), 'Retrieval-augmented reinforcement learning'
[252]: Hinton et al. (2014), 'Distilling the knowledge in a neural network'
[228]: Chen et al. (2018), 'Lifelong machine learning'
[229]: Thrun (1995), 'Is learning the n-th thing any easier than learning the first?'
[253]: Davidson et al. (2020), 'Sequential mastery of multiple visual tasks: Networks naturally learn to learn and forget to forget'
[254]: van der Vaart et al. (1996), *Weak Convergence and Empirical Processes*
[255]: Rebuffi et al. (2017), 'icarl: Incremental classifier and representation learning'
[256]: Li et al. (2017), 'Learning without forgetting'
[257]: Shin et al. (2017), 'Continual learning with deep generative replay'
[258]: Tang et al. (2022), 'Learning to imagine: Diversify memory for incremental learning using unlabeled data'
[259]: Kirkpatrick et al. (2017), 'Overcoming catastrophic forgetting in neural networks'
[260]: Zenke et al. (2017), 'Continual learning through synaptic intelligence'

architecture using discrete key-value codes avoids this constraint. The information is distilled into discrete key-value codes and a sparse set of value codes are locally adapted when faced with new tasks, which integrates information organically as discussed in Section 6.3 and has some parallels with local learning ideas [276]. A central part of the proposed method is the use of pre-trained models, a prerequisite that is also harnessed for continual learning tasks recently [277, 278]. The proposed method builds upon discrete representations, most notably the VQ-VAE [237, 239] and various successor methods that have been shown to improve robustness under data corruptions [238, 239, 279–282]. While these works can similarly reduce the representation space and enhance robustness, such bottlenecks have a single set of codes, which does not allow tuning them while keeping the feature-extracting backbone frozen. Finally, the key-value pairs can be interpreted as a memory bottleneck from which the model can selectively retrieve stored information. This interpretation bears similarity to memory networks, such as those proposed by [283–287], which also write and read information from a set of symbol-like memory cells. However, key-value codes act as an information bottleneck, whereas memory networks condition predictions on retrieved memory.

6.5 Experiments

The insight we aim to convey with the presented experiments is to validate that having a key-value bottleneck allows for *localized, context-dependent model updates* that are robust to training under input distribution changes. We will show that this allows us to *reduce catastrophic forgetting*; *integrate new information*; and thereby *mitigate vulnerabilities* in challenging class-incremental learning settings. We first showcase a simplified yet challenging class-incremental learning setting to give readers an understanding of why standard approaches fail and how the proposed method succeeds. We then explore the method's benefits under a challenging class-incremental learning scenario on CIFAR10 and analyze the role of different components and design choices of a key-value bottleneck. The code for reproducing all experiments can be found in the supplementary material.

6.5.1 A simplified learning setting motivating the method

We consider a 2D input feature classification problem for 8 classes, where the training data is not i.i.d. but changes over four stages (see Figure 6.2). In each stage, we sample 100 examples of two

[261]: Xiao et al. (2014), 'Error-driven incremental learning in deep convolutional neural network for large-scale image classification'

[262]: Verma et al. (2021), 'Efficient feature transformations for discriminative and generative continual learning'

[263]: Gao et al. (2022), 'Efficient architecture search for continual learning'

[264]: Serra et al. (2018), 'Overcoming catastrophic forgetting with hard attention to the task'

[265]: Ke et al. (2021), 'Achieving Forgetting Prevention and Knowledge Transfer in Continual Learning'

[266]: Harrison et al. (2020), 'Continuous meta-learning without tasks'

[267]: He et al. (2019), 'Task agnostic continual learning via meta learning'

[268]: Finn et al. (2019), 'Online meta-learning'

[269]: Kanerva (1988), *Sparse distributed memory*

[270]: Kanerva (1992), *Sparse distributed memory and related models*

[272]: Zeng et al. (2019), 'Continual learning of context-dependent processing in neural networks'

[273]: Saha et al. (2021), 'Gradient Projection Memory for Continual Learning'

[276]: Bottou et al. (1992), 'Local learning algorithms'

[277]: Banayeeanzade et al. (2021), 'Generative vs. Discriminative: Rethinking The Meta-Continual Learning'

[278]: Ostapenko et al. (2022), 'Continual learning with foundation models: An empirical study of latent replay'

[237]: Van Den Oord et al. (2017), 'Neural discrete representation learning'

[239]: Liu et al. (2021), 'Discrete-valued neural communication'

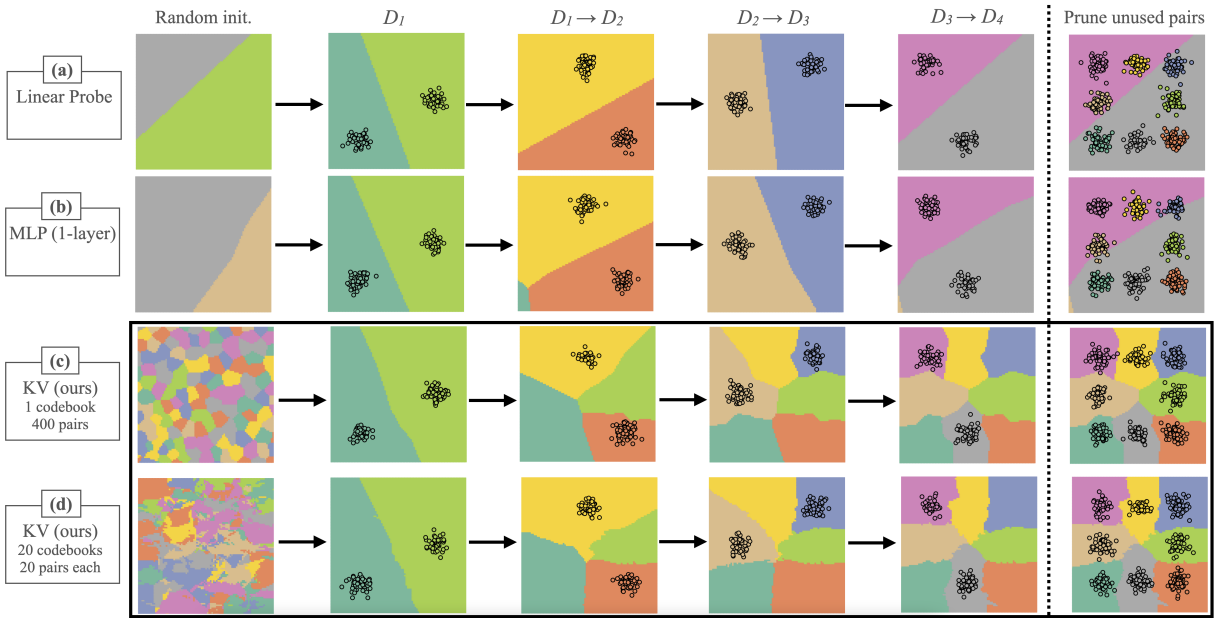


Figure 6.2: Incremental training on non-overlapping datasets $\mathcal{D}_1 \rightarrow \mathcal{D}_2 \rightarrow \mathcal{D}_3 \rightarrow \mathcal{D}_4$ (dots) using (a) Linear Probe (LP), (b) 1-layer MLP (MLP), (c) key-value bottleneck with C = 1 and ground truth feature heads, (d) key-value bottleneck with C = 20 and randomly projected feature heads. Using a discrete bottleneck of 400 key-value pairs with fixed key codes, and locally updated value codes, alleviates catastrophic forgetting. Color domains indicate one out of the 8 possible classes predicted by the model.

classes for 1000 training steps, using gradient descent to update the weights, then move on to two new classes for the next 1000 steps. The input features of each class follow spatially separated normal distributions. We first discuss the behaviour on this task using a vanilla linear probe or 1-layer MLP with 32 hidden dimensions trained by minimizing the cross-entropy along the training data stream. As can be seen from the decision boundaries in Figure 6.2 (a, b), these naive approaches — unsurprisingly — over-fit on the most recent training data. This behaviour is because the MLP is optimizing *all* weights on the most recent training data, thereby overwriting prior predictions. This problem can be easily alleviated by *sparse, local* updates using a discrete key-value bottleneck. Here we present two different configurations. In the first configuration, we use 400 pairs of 2-dim keys and 8-dim values without random projection and we initialize these keys from random points in the domain $x \in [0, 1]^2$ (not the target dataset). Input x_i will now snap to the closest key and fetch its corresponding value. After keys are initialized, they are frozen, and we train the model on the non-stationary data stream using the same loss and optimizer as before but optimizing the values only. As can be seen from Figure 6.2 (c), the proposed bottleneck mechanism will now solely update the input-dependent value codes. This enables the model to incrementally update its predictions on unseen input domains and minimize interference. Finally, we can prune all key-value pairs that were never selected in order to remove any dormant key-value codes (rightmost column). In the above example, it is

[238]: Razavi et al. (2019), 'Generating diverse high-fidelity images with vq-vae-2'
[239]: Liu et al. (2021), 'Discrete-valued neural communication'
[279]: Zeghidour et al. (2021), 'Soundstream: An end-to-end neural audio codec'
[280]: Yu et al. (2022), 'Vector-quantized Image Modeling with Improved VQGAN'
[281]: Shin et al. (2021), 'Translation-equivariant Image Quantizer for Bi-directional Image-Text Generation'
[282]: Mama et al. (2021), 'NWT: Towards natural audio-to-video generation with representation learning'
[283]: Sukhbaatar et al. (2015), 'End-to-end memory networks'
[284]: Chandar et al. (2016), 'Hierarchical memory networks'
[285]: Webb et al. (2021), 'Emergent Symbols through Binding in External Memory'
[286]: Lample et al. (2019), 'Large memory layers with product keys'
[287]: Panigrahy et al. (2021), 'Sketch based Memory for Neural Networks'

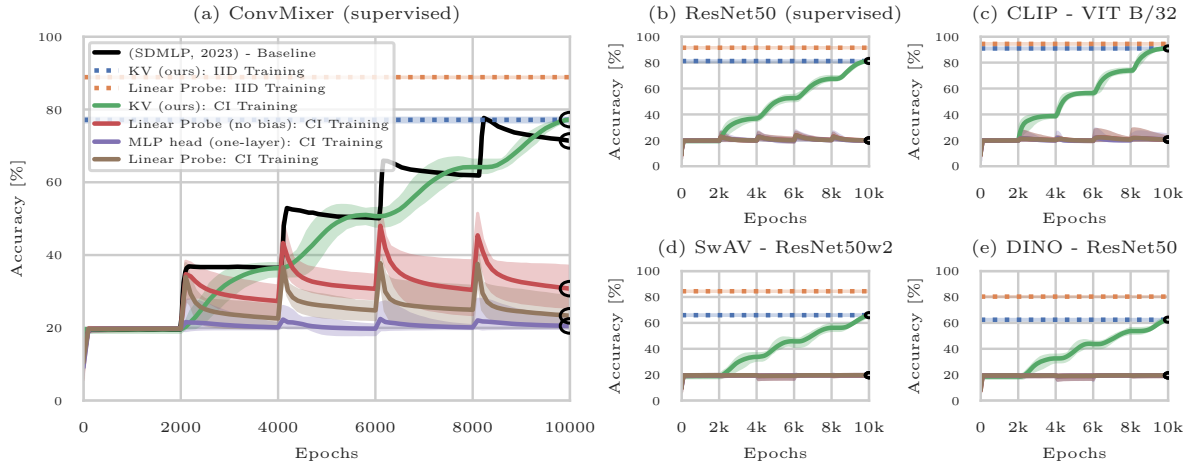


Figure 6.3: The key-value bottleneck architecture successfully learns and integrates new information in a challenging 5-split CIFAR10 class-incremental (CI) learning setting. Our method (green line) reduces catastrophic forgetting and yields strong final performance across various pre-trained frozen backbone architectures including (a) ConvMixer; (b) ResNet50 ; (c) ViT B/32 with CLIP, (d) ResNet50w2 with SwAV; (e) ResNet50 with DINO. At the end of the training, the proposed method outperforms the reported performance of the SDM method using the same pre-trained ConvMixer. Standard probe tuning exhibits major forgetting in this scenario (red, brown, purple curve).

essential to have two separate sets of codes compared to a single set of discrete codes. Finally, we also study a configuration with 20 codebooks each with 20 key-value pairs and use randomly initialized projection matrices. This configuration, which uses multiple codebooks but the same total number of key-value pairs, works just as well as seen in Figure 6.2 (d).

6.5.2 Continual learning on real-world data

Having established some first intuition on the localized model updates, we now attempt to present the strong learning capabilities of the introduced bottleneck on a challenging and realistic continual class-incremental learning task.

Experimental setup. We use a class-incremental CIFAR10 task with pre-training on another dataset without any memory replay or provision of task boundaries, which is a very challenging task setting in continual learning. Here, five disjoint sets with two classes each are incrementally presented for many epochs each and the goal is to learn new classes and not forget previous classes at the very end of the training. We present each set for 2000 epochs, which in an ordinary architecture would cause catastrophic forgetting of the previous sets. We perform five replications of each model with different random seeds, with the seed also re-sampling the class splits to avoid selecting any favourable class split. Importantly, we study one of the hardest yet most realistic settings in continual learning by *not allowing any memory replay or provision of task boundaries* which is required by the vast majority of existing

continual learning methods. To the best of our knowledge, the only method that can deal with this learning scenario is Bricken et al. [271]. We report experiments on five publicly available backbones including:

1. ResNet50 pre-trained on ImageNet
2. ViT-B/32 pre-trained with CLIP
3. ResNet50w2 pre-trained with SwAV as SSL method
4. ResNet50 pre-trained with DINO as SSL method
5. ConvMixer pre-trained on 32x32 downsampled Imagenet. We select this backbone to directly compare against the performance reported in [271].

We initialize keys on the unlabelled non-overlapping CIFAR100 dataset except for the ConvMixer where we used the embeddings from the downsampled Imagenet dataset for reasons of comparison. We test the wide applicability of key-value bottlenecks across these different architectures and pre-training schemes, using 256 codebooks with 4096 key-value pairs each, 14-dimensional keys and 10-dimensional values. We additionally report results for the proposed bottleneck and a Linear Probe under an i.i.d. training scheme (presenting all classes simultaneously), with the latter representing a fair oracle upper bound. When learning class-incrementally, we compare against the reported SDMLP performance on the ConvMixer backbone, as well as a 1-layer MLP with 128 hidden dimensions and a linear probe with and without bias term for all backbones.

Results. We summarize our results in Figure 6.3. First, we observe that the key-value bottleneck can indeed successfully learn and integrate new information under such a difficult class-incremental distribution shift consistently (green curves). At the same time, the key-value bottleneck substantially reduces the severe catastrophic forgetting over the standard adaptation approaches (red, brown and purple curves). In line with the theory, we observe *across all encoder backbones* the appealing learning property that the bottleneck yields the same performance at the end of training compared to the benign i.i.d. setting (blue lines). Within the class incremental setting only, we observe that the *final performance* is almost identical, irrespective of the sampled class split. This result suggests that the bottleneck can successfully mitigate the typically very destructive effect of learning under non-stationary training data and might be particularly robust against training curricula that are otherwise adversarial to stochastic gradient-based training. We can partially explain this behaviour by the fact that we freeze the keys during training and use a simple momentum-free SGD optimizer. In this case, any value code will approximatively receive the same gradients over the entire course of training.



Figure 6.4: We demonstrate that the proposed key-value bottleneck method maintains a constant loss measure, directly translating to improved generalization when training class incrementally, as opposed to standard approaches which can over-fit.

We now take a look at the results of the ConvMixer backbone in Figure 6.3 (a). This reflects the same setup as in the SDMLP method [271], the only applicable method we are aware of in this challenging task setting without memory replay or provision of task boundaries. The key-value bottleneck method achieves a final accuracy of 77.3% compared to 71% for the best final accuracy with SDMLP. Additionally, the learning curves with the bottleneck show smooth improvements in performance, whereas the baseline spikes at the introduction of new tasks and exhibits pronounced catastrophic forgetting. We refer to Appendix F.4 for further experiments. Furthermore, we observe consistent results across all other four backbones. The best results are achieved with a CLIP-encoder with 91.3% and the supervised pre-trained ResNet50 backbone with 81.5% final performance (Figure 6.3 b & c). We are similarly getting strong class-incremental learning abilities for both SSL backbones, which were pre-trained without any labels, achieving up to 66.3% in the case of SwAV and 62.5% in the case of DINO (Figure 6.3 d & e). Finally, we believe these results are further encouraging as they demonstrate that the key-value bottleneck is able to achieve strong performance across a wide variety of pre-trained backbones and learned feature spaces, suggesting that it has the potential to scale to even more performant models in the future. Finally, we show in Figure 6.4 that a key-value bottleneck does not over-fit like other standard approaches. A linear probe may perform better when trained on all classes at once in an i.i.d. setting; but when trained incrementally, it over-fits. However, the key-value bottleneck maintains a constant loss measure over both, i.i.d. and class-incremental settings, which leads to better overall performance when training on new classes. This aligns with our analysis of the first term in Theorem 6.3.1.

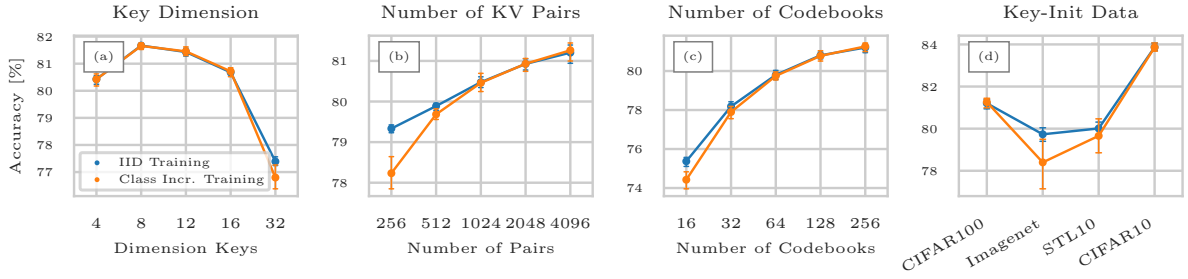


Figure 6.5: Assessing the role of individual model components starting from the base model architecture in Figure 6.3 b. Reducing the dimensionality of keys has a positive effect (a); increasing the number of pairs and codebooks enhances generalization performance (b + c); finally, key-value bottlenecks keys can be initialized from a different dataset without major drop in overall performance (d).

6.5.3 Ablation and Sensitivity Analysis

We now analyze the role of various relevant bottleneck components to better understand its learning abilities, as well as further opportunities and limitations. We point the reader to Appendix F.5.3 for an in-depth analysis of the (key,value) codes learned by the models. In the following paragraphs, we focus on the ResNet50 backbone and analyze the effect of changing one particular component while keeping everything else fixed. We present the results in Figure 6.5.

Dimension of key codes. First, we found that there is an optimal number of dimensions for the keys, around 8 to 12, which can be explained as follows. Having keys with high dimensions requires more data to cover the backbone manifold sufficiently. On the other hand, having keys with too few dimensions means that the keys are scattered in a much smaller space which increases the chances of unintended key sharing among different classes.

Number of key-value pairs. The number of keys used in the bottleneck can greatly affect the performance of the model. We found that increasing the number of key-value pairs leads to better performance, with only a small decrease in performance when using four times fewer pairs. Our analysis also suggests that the model performance will continue to improve asymptotically, though we have not yet reached that point. When using 256 or fewer key-value pairs, we noticed a separation between the performance of the model in i.i.d. and class-incremental settings. This is likely because when there are fewer pairs, more of them are shared among multiple classes, and this allows for a more balanced update of the gradients when training on mini-batches.

Number of codebooks. The number of codebooks is another important factor that controls the capacity of the model, similar to the number of key-value pairs. We observed that using more

codebooks leads to better performance, with a slightly more pronounced effect compared to the number of pairs. Furthermore, increasing the number of codebooks also decreases the training loss term. However, it is important to note that increasing the number of codebooks too much can also increase the value of c in Theorem 6.3.1, although we have not yet observed that in our experiments.

Key initialization dataset. Finally, we tested the robustness of the proposed method by initializing the keys on different datasets, such as STL10 and Imagenet, in addition to CIFAR100 used in previous experiments [111, 288]. We also calculated the performance of an "oracle" key initialization by using the unlabelled CIFAR10 dataset. While there is a slight decrease in performance compared to the oracle, the architecture is robust to different datasets, even if those datasets differ in resolution and scenes.

[111]: Krizhevsky et al. (2009), 'Learning multiple layers of features from tiny images'
[288]: Coates et al. (2011), 'An analysis of single-layer networks in unsupervised feature learning'

6.6 Limitations and Extensions

Pre-trained encoders. The key-value bottleneck method relies on pre-trained encoders that can extract meaningful features shared between the pre-training and actual training data. This is an important requirement that has been shown by prior works in transfer learning [231, 232, 289]. However, as self-supervised pre-trained models from massive datasets become more prevalent, a key-value bottleneck can be used to connect these encoders to downstream tasks and minimize forgetting under distribution changes. Additionally, the method relies on encoders that are able to separate distinct features locally, as supported by recent research [233, 290–292]. Fine-tuning the encoder directly can distort these features and negatively affect out-of-distribution generalization [293]. The key-value bottleneck reduces this issue as encoder features are not influenced by fine-tuning.

Selection and initialization of key codes. The selection and initialization of key codes is an important aspect of the proposed method. The approach of using EMA works well for a range of datasets, but it may have limitations in extreme situations where the dataset is unknown or changes completely, such as in RL. We did not explore these scenarios in this work, but we believe that discrete key-value bottlenecks have the potential to be extended to handle them. For example, the encoder, projection, and keys can be fine-tuned asynchronously with auxiliary unsupervised tasks to handle distribution shifts. Additionally, information about the distance between the codebook input heads and their closest keys can be used to estimate the level of distribution shift and adapt, add, or reinitialize key-value pairs as needed.

[231]: Chen et al. (2020), 'Big self-supervised models are strong semi-supervised learners'
[232]: Azizi et al. (2021), 'Big self-supervised models advance medical image classification'
[289]: Yosinski et al. (2014), 'How transferable are features in deep neural networks?'
[233]: Caron et al. (2021), 'Emerging Properties in Self-Supervised Vision Transformers'
[290]: Huang et al. (2019), 'Unsupervised deep learning by neighbourhood discovery'
[291]: Zhou et al. (2022), 'Understanding the robustness in vision transformers'
[292]: Tian et al. (2022), 'Deeper Insights into ViTs Robustness towards Common Corruptions'
[293]: Kumar et al. (2022), 'Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution'

Decoder architecture When classifying, we found it effective to use a non-parametric decoder without learnable weights. Nevertheless, we can also train a discrete key-value bottleneck with more complex, parametric decoders. When training the model under i.i.d. conditions, we can adjust the decoder weights to improve performance. However, when the training conditions change, e.g. in incremental learning, we need to be more careful. One could make the decoder weights adjustable and augment with various regularization methods or keep them fixed if necessary. Combining key-value bottlenecks with advanced, pre-trained decoders such as generative models is a promising area for future research.

Trade-offs using a bottleneck. Using an information bottleneck involves a trade-off between accuracy and information compression. The proposed bottleneck applies strong compression by propagating only a discrete set of C key-value pairs, preventing changes to the encoder. This allows for training under strong distribution shifts without forgetting, but with slightly lower performance. If the target data distribution is available before (no labels needed), we can improve performance by initializing the keys on this distribution, but there are also other ways to potentially improve further such as parametric decoders or allowing back-propagation into the projection weights and keys through techniques such as sampling or straight-through estimators.

6.7 Conclusion

We proposed a discrete key-value bottleneck architecture that can adapt to input distribution shifts and performs well in class-incremental learning settings. We supported the model's effectiveness theoretically and through empirical validation. Specifically, we showed that the use of key-value pairs in the architecture can reduce the generalization error and prevent over-fitting. The model is able to integrate new information locally while avoiding catastrophic forgetting. We believe the key-value bottleneck could be a promising solution for tackling challenging training scenarios.

Backward-Compatible Prediction Updates: A Probabilistic Approach

7

Abstract

When machine learning systems meet real world applications, accuracy is only one of several requirements. In this paper, we assay a complementary perspective originating from the increasing availability of pre-trained and regularly improving state-of-the-art models. While new improved models develop at a fast pace, downstream tasks vary more slowly or stay constant. Assume that we have a large unlabelled data set for which we want to maintain accurate predictions. Whenever a new and presumably better ML model becomes available, we encounter two problems: (i) given a limited budget, which data points should be re-evaluated using the new model?; and (ii) if the new predictions differ from the current ones, should we update? Problem (i) is about compute cost, which matters for very large data sets and models. Problem (ii) is about maintaining consistency of the predictions, which can be highly relevant for downstream applications; our demand is to avoid negative flips, i.e., changing correct to incorrect predictions. In this paper, we formalize the Prediction Update Problem and present an efficient probabilistic approach as answer to the above questions. In extensive experiments on standard classification benchmark data sets, we show that our method outperforms alternative strategies along key metrics for backward-compatible prediction updates.

About This Chapter

This chapter has been adapted from **Träuble** et al. [6].

Backward-Compatible Prediction Updates: A Probabilistic Approach

Frederik Träuble, Julius Von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf and Peter Gehler.

NeurIPS 2021: Advances in Neural Information Processing Systems.

7.1 Introduction 101
7.2 The Prediction Updates Problem Setting 104
7.3 Our Method 105
7.4 Discussion: Extensions and Limitations 109
7.5 Related Work 110
7.6 Experiments 111
7.7 Conclusion 119

CRediT author statement

Frederik Träuble: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Julius von Kügelgen**: Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing **Matthäus Kleindessner**: Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing **Francesco Locatello**: Writing - Original Draft, Writing - Review & Editing **Bernhard Schölkopf**: Resources, Writing - Review & Editing, Supervision **Peter Gehler**: Conceptualization, Methodology, Validation, Investigation, Resources, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration, Funding acquisition

7.1 Introduction

The machine learning (ML) community develops new models at a fast pace: for example, just in the year 2020, the state-of-the-art on ImageNet has changed at least five times [29, 294–297]. As reproducibility has increasingly been scrutinized [298–300], it is now common practice to release pre-trained models upon publication. In this work we take the perspective of an owner of an unlabelled data set who is interested in keeping the best possible predictions at all times. When a new pre-trained model is released, we face what we refer to as the *Prediction Update Problem*: (i) decide which points in the data set to re-evaluate with the new model, and (ii) integrate the new, possibly contradicting, predictions. For this task, we postulate the following three desiderata:

1. The prediction updates should improve overall accuracy.
2. The prediction updates should avoid introducing new errors.
3. The prediction updates should be as cheap as possible since the target data set could be huge.

We consider the setting in which the target data set for which we wish to maintain predictions is fully unlabelled (i.e., the ground-truth labels are unknown) and may come from a different distribution than the one on which models have been pre-trained, but with overlap in the label space. This is a transductive or semi-supervised problem, but, due to computational constraints, we avoid any model fitting or fine-tuning and rely solely on the predictions of the pre-trained models that are released over time. Typically, these models exhibit increased performance on their labelled training domain (e.g., the ImageNet validation or test set) as evidence for being good candidates for re-evaluation.

Clearly, one goal of updating the predictions stored for the target data set is to improve overall performance, e.g., top-k accuracy for classification. At the same time, the stored predictions may form an intermediate step in a larger ML pipeline or are accessible to users. This is the reason for our second desideratum: we would like to be *backward-compatible*, i.e., new predictions should not flip previously correct predictions (*negative flips*). Finally, we aim to reduce computational cost during inference and to avoid evaluating the entire data set which may be prohibitive in practice and unnecessary if we are already somewhat certain about a prediction.

In this paper, we motivate and formalize the *Prediction Update Problem* and describe its relation to various relevant research areas like ensemble learning, domain adaption, active learning, and others. We propose a probabilistic approach that maintains a posterior

[29]: Dosovitskiy et al. (2020), 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'
[294]: Xie et al. (2020), 'Self-training with noisy student improves imagenet classification'
[295]: Touvron et al. (2019), 'Fixing the train-test resolution discrepancy: Fixefficientnet'
[296]: Foret et al. (2021), 'Sharpness-aware Minimization for Efficiently Improving Generalization'
[297]: Pham et al. (2021), 'Meta pseudo labels'
[298]: Pineau et al. (2018), 'ICLR2018 reproducibility challenge'
[299]: Sinha et al. (2020), 'NeurIPS 2019 Reproducibility Challenge'
[300]: Pineau et al. (2019), 'ICLR Reproducibility Challenge 2019'

distribution over the unknown true labels by combining all previous model re-evaluations. Based on these uncertainty estimates, we devise an efficient *selection strategy* which only chooses those examples with highest posterior label entropy for re-evaluation in order to reduce computational cost. Furthermore, we consider different prediction-update strategies to decide whether to change the stored predictions, taking asymmetric costs for negative and positive flips into account. Using the task of image classification as a case study, we perform extensive experiments on common benchmarks (ImageNet, CIFAR10, and ObjectNet) and demonstrate that our approach achieves competitive accuracy and introduces much fewer negative flips across a range of computational budgets, thus showing that our three desiderata are not necessarily at odds.

Contributions We highlight the following contributions:

- We introduce the *Prediction Update Problem* which addresses some common, but previously unaddressed challenges faced in real world ML systems (Section 7.2).
- We propose a probabilistic, model-agnostic approach for the *Prediction Update Problem*, based on Bayesian belief estimates of the true label combined with an efficient selection and different prediction-update strategies (Section 7.3).
- We contextualise this understudied problem setting as well as our method with related work (Section 7.4 & Section 7.5) and discuss several extensions and limitations (Section 7.4).
- We demonstrate that our approach successfully outperforms alternative approaches and accomplishes all our desiderata in experiments across multiple common benchmark datasets (CIFAR-10, ImageNet, and ObjectNet) and practically relevant scenarios (Section 7.6).

7.1.1 Backward-Compatible Prediction Systems

In real world ML applications, empirical performance is only one of several requirements. When humans interact with automatic predictions, they will start to build mental models of how these models operate and whether and when their predictions can be trusted. This is described as Human-AI teams by [301] who argue to "make the human factor a first-class consideration of AI updates".

An example from [301] is autopilot functionality in cars for which drivers will build expectations in which driving situations the autopilot is safe to engage. It is important not to violate these assumptions when updating the models over the air. AI assisted medical decision processes are another example of a high stake application where medical professionals need to understand when systems can be trusted.

Consider the example of automatically tagging images in a user's photo collection. Those tags are used for example in photo search. As models progress, the overall accuracy on all uploaded images may increase, but for any single user the experience can deteriorate if previously correct searches now show wrong results. Even worse, if errors fluctuate over the user's photo collection as the result of prediction-updates, the user's trust will be eroded. This "cost" is asymmetric and the negative experience may outweigh the benefit of better predictions on other images.

In contrast to carefully curated and labelled ML benchmarks, many real-world data sets are magnitudes larger (up to billions of samples) and entirely unlabelled. Having no feedback which predictions are correct is a common scenario: consider any type of private data such as health data, photo collections, or personal information. Because the data is private, we can neither train on it, nor collect feedback, nor observe the effect of predictions. On the other hand, such data is valuable to an individual: she has an interest to keep it up to date with the best possible predictions. Since it is of little consolation to her if an update of the model improves predictions on average but on her data it gets worse, the update costs are asymmetric. Service providers often rely on models pre-trained on a different data set, and the desire to be backward-compatible arises naturally in this setting [301–304].

This is an understudied problem where progress will have large impact. ML systems are becoming pervasive, and their accuracy will continue to increase. Being able to seamlessly transfer them to existing data will be crucial for real-world ML systems.

[301]: Bansal et al. (2019), 'Updates in Human-AI teams: Understanding and addressing the performance/-compatibility tradeoff'
[302]: Yan et al. (2021), 'Positive-Congruent Training: Towards Regression-Free Model Updates'
[303]: Shen et al. (2020), 'Towards backward-compatible representation learning'
[304]: Srivastava et al. (2020), 'An Empirical Analysis of Backward Compatibility in Machine Learning Systems'

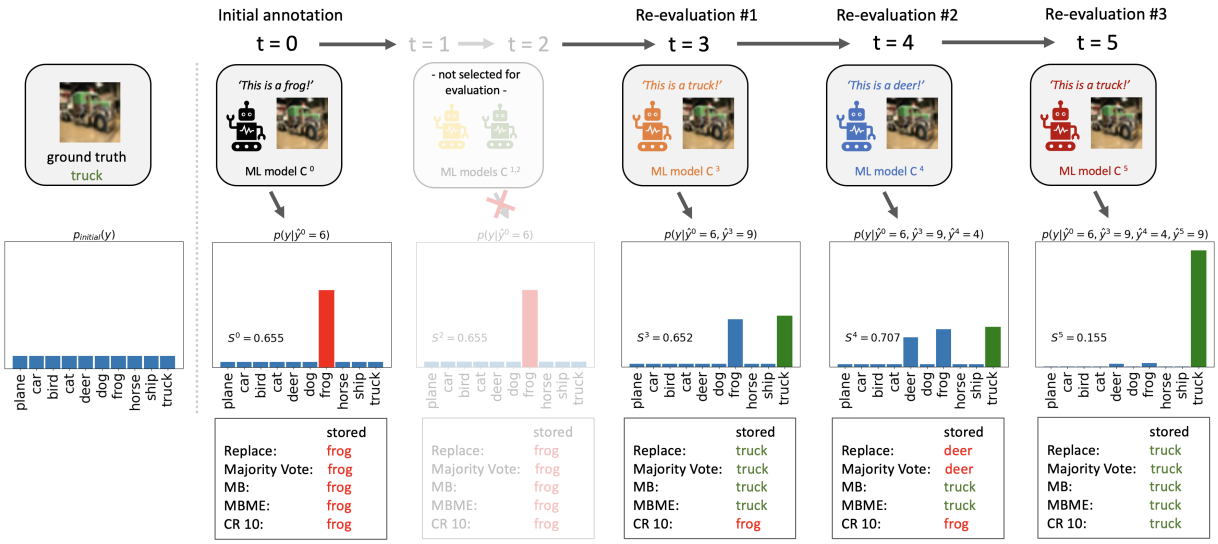


Figure 7.1: Overview of our proposed Bayesian approach to the Prediction Update Problem. Starting from a uniform prior, we maintain a posterior distribution $p(y = k|\hat{y}^{0:t})$ (*middle*) over the unknown true label y of an unlabelled sample which takes the predictions $\hat{y}^{0:t}$ from new ML models C^t (*top*) arriving over time $t = 0, ..., T$ into account. Given a limited compute budget B^t, we re-evaluate those samples with highest posterior label entropy S^t at each time step, e.g., the example shown is first selected in time step $t = 3$ after the initial annotation at $t = 0$. We then consider different strategies for deciding whether to update the stored prediction (*bottom*) based on our changed beliefs. Note that the non-probabilistic baselines "Replace" (always update to last prediction) and "Majority Vote" (resolve ties by using the last prediction) incorrectly update the stored prediction from "truck" to "deer" in step $t = 4$. Our strategies (MB, MBME, CR-10) which rely on the estimated label posterior, on the other hand, avoid such a *negative flip*, which is one of our key goals.

7.2 The Prediction Updates Problem Setting

Target Data Set We are given a large, unlabelled target data set $\mathcal{D}^{\text{targ}} = \{\mathbf{x}_n\}_{n=1}^N$ comprising N independent and identically distributed (i.i.d.) observations $\mathbf{x}_n \in \mathcal{X} \subseteq \mathbb{R}^d$ drawn from a target distribution $\mathbb{P}_{\mathbf{X}}^{\text{targ}}$. The ground-truth labels $y_n \in \mathcal{Y} = \{1, ..., K\}$ distributed according to $\mathbb{P}_{Y|\mathbf{X}}^{\text{targ}}$ are *not observed*. Note that we are particularly interested in a scenario where N may be extremely large.

Models Over time $t = 0, 1, ..., T$ we successively gain access to classifiers $C^0, C^1, \ldots, C^T : \mathcal{X} \to \mathcal{Y}$ which have been trained on a labelled data set $\mathcal{D}^{\text{src}}$ from a potentially different source distribution $\mathbb{P}_{\mathbf{X},Y}^{\text{src}}$ over $\mathcal{X} \times \mathcal{Y}$. For simplicity, we assume that the observation space $\mathcal{X}$ and label space $\mathcal{Y}$ are shared. We consider both the standard scenario where the models $\{C^t\}_{t=0}^T$ are trained on a labelled set from the same domain ($\mathbb{P}^{\text{src}} = \mathbb{P}^{\text{targ}}$); and the transfer scenario where we deploy a model trained on a labelled ML benchmark to a different data set ($\mathbb{P}^{\text{src}} \neq \mathbb{P}^{\text{targ}}$). We assume that $\{C^t\}_{t=0}^T$ are improving in performance on the training data set. Therefore, denoting by A_t the estimated accuracy of C^t on $\mathbb{P}_{\mathbf{X},Y}^{\text{src}}$, we have $A^t \leq A^{t+1}\ \forall t$. As motivating example, consider an object recognition task in the wild and let C_t be the winning entry of the ImageNet competition in year t.

Labelling To relate the source and target distributions and to justify applying $\{C^t\}_{t=0}^T$ to our target data set $\mathcal{D}^{\text{targ}}$, we make the commonly used covariate shift assumption $\mathbb{P}^{\text{targ}}_{Y|\mathbf{X}} = \mathbb{P}^{\text{src}}_{Y|\mathbf{X}}$, i.e. the conditional label distribution is shared across source and target distributions [305, 306].* We denote the *predicted* label by C^t for $\mathbf{x}_n$ by $\hat{y}_n^t = C^t(\mathbf{x}_n)$ and the *stored* prediction for $\mathbf{x}_n$ after time step t by l_n^t. The target data set is then initially fully labelled by C^0, i.e., $l_n^0 := \hat{y}_n^0$.

[305]: Shimodaira (2000), 'Improving predictive inference under covariate shift by weighting the log-likelihood function'
[306]: Sugiyama et al. (2012), *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*

Objective As new classifiers $\{C^t\}_{t\geq 1}$ become available, our main objective is to maintain the best estimates $\{l_n^t\}_{n=1}^N$ on our target data set at all times and improve overall accuracy, while, at the same time, maintaining backward compatibility by minimising the number of *negative flips*, i.e., the number of previously correctly stored predictions that are incorrectly changed. The key challenge is that no ground truth labels for our target data set are available, so that we have no feedback on which predictions are correct and which are wrong. For each test sample $\mathbf{x}_n$ and each time step $t \geq 1$, we thus need to decide whether or not to update the previously stored prediction l_n^{t-1} based on the current and previous model predictions $\hat{y}_n^t$ and $\hat{y}_n^{0:t-1}$, respectively.

Limited Evaluation Budget Re-evaluating all samples (so-called *backfilling*) can be very costly and requires significant resources. Since we consider N to be very large, we also consider a limited budget of at most $B^t \leq N$ sample re-evaluations for step t. We thus additionally need to decide how to allocate this budget and select a subset of samples to be re-evaluated by C^t at every step.

7.3 Our Method

Having specified the setting, we next describe our proposed method for the Prediction Update Problem. We start by providing a Bayesian approach for maintaining and updating our beliefs about the unknown true labels as new predictions become available (Section 7.3.1), followed by describing strategies for selecting candidate samples for re-evaluation (Section 7.3.2) and for updating the stored predictions based on our changed beliefs (Section 7.3.3). Our framework is summarised in Figure 7.1.

* In the context of image classification, this means that images of the same object under different environmental conditions (i.e., different $\mathbb{P}_{\mathbf{X}}$) will always share the same label (i.e., same $\mathbb{P}_{Y|\mathbf{X}}$). Note that such covariate shift may also lead to changes in $\mathbb{P}_Y$ (label/target shift) and/or $\mathbb{P}_{\mathbf{X}|Y}$ (conditional shift) [307, 308].

7.3.1 Bayesian Approach

Since the true labels $\{y_n\}_{n=1}^N$ are unknown to us, we treat them as random quantities over which we maintain uncertainty estimates. We then perform Bayesian reasoning to update our beliefs as new evidence in the form of predictions $\hat{y}_n^t$ from newly-available classifiers C^t arrives over time $t = 1, \ldots, T$. In standard Bayesian notation, the true labels y_n thus take the role of unknown parameters θ and the predictions $\hat{y}_n^t$ of data x. Since $\mathcal{D}^{\text{targ}}$ is sampled i.i.d., we reason about each label y_n independently of the others, i.e., the following is the same for all n.

Prior Lacking label information on the target data set, we choose a uniform prior over $\mathcal{Y}$ for all y_n, i.e.,

$$p(y_n = k) = 1/K, \ \forall k \in \mathcal{Y}.$$

If (estimates of) the class probabilities on $\mathcal{D}^{\text{targ}}$ are available, we may instead use these as a more informative prior.

Likelihood Next, we need to specify a likelihood function $p(\hat{y}_n^{0:T} | y_n = k)$ for the observed model predictions $\hat{y}_n^{0:T}$ given a value k of the true label y_n. We make the following simplifying assumption.

Assumption 7.3.1 (Conditionally independent classifiers) *The different classifiers' predictions $\hat{y}_n^{0:T}$ are conditionally independent given the true label y_n, i.e., the likelihood factorizes as*

$$p(\hat{y}_n^{0:T} | y_n = k) = \prod_{t=0}^{T} p(\hat{y}_n^t | y_n = k). \tag{7.1}$$

In a standard Bayesian setting, this corresponds to the assumption of conditionally independent observations given the parameters; we refer to Section 7.4 for further discussion. The main advantage of Assumption 7.3.1 is that the factors $p(\hat{y}_n^t | y_n = k)$ on the RHS of (7.1) have a natural interpretation: these are the (normalised) confusion matrices π^t of the classifiers C^t, i.e., we denote by

$$\pi^t(i, k) := p(\hat{y}^t = i | y = k),$$

the probability that C^t predicts class i given that the true label is k, which is the same for all n; see below and Section 7.4 for more details on how we estimate π^t in practice.

Posterior At every time step $t \geq 0$, we can then compute our posterior belief about the true label y_n given model predictions $\hat{y}_n^{0:t}$ according to Bayes rule,

$$p(y_n = k | \hat{y}_n^{0:t}) = \frac{\pi^t(\hat{y}_n^t, k) p(y_n = k | \hat{y}_n^{0:t-1})}{\sum_{i \in \mathcal{Y}} \pi^t(\hat{y}_n^t, i) p(y_n = i | \hat{y}_n^{0:t-1})} \tag{7.2}$$

where we have used Assumption 7.3.1 to write

$$p(\hat{y}_n^t | y_n = k, \hat{y}_n^{0:t-1}) = p(\hat{y}_n^t | y_n = k) = \pi^t(\hat{y}_n^t, k).$$

The posterior at step $t-1$ acts as prior for step t, so we do not have to store all previous predictions.

Estimating Confusion Matrices In practice, π^t are generally not known and we instead use their (maximum likelihood) estimates $\hat{\pi}^t$ from the source distribution. If the number of classes K is large compared to the amount of labelled source data,[†] we only estimate the diagonal elements $\hat{\pi}_{kk}^t$ (i.e., the class-specific accuracies) and set the $K(K-1)$ off-diagonal elements to be constant,

$$\hat{\pi}^t(i,k) = \frac{1-\hat{\pi}^t(k,k)}{K-1} \qquad \forall i \neq k,$$

so that $\sum_{i=1}^K \hat{\pi}^t(i,k) = 1 \; \forall k \in \mathcal{Y}$. We refer to Section 7.4 for further discussion on the estimation of π^t.

7.3.2 Selecting Candidates for Re-evaluation

Given the label posteriors computed according to (7.2), we compute the Shannon entropies [309]

$$S_n^t = -\sum_{k \in \mathcal{Y}} p(y_n = k | \hat{y}_n^{0:t}) \log p(y_n = k | \hat{y}_n^{0:t}),$$

[309]: Shannon (1948), 'A mathematical theory of communication'

which provide a simple measure of uncertainty in the true label y_n after step t. We then select and re-evaluate the B^t samples with highest posterior label entropy S_n^t to update our beliefs.

7.3.3 Prediction-Update Strategies

Finally, we need a strategy for deciding whether and how to update the previously stored prediction l_n^{t-1} based on our new beliefs. We consider three such prediction-update strategies.

MaxBelief (MB) The simplest approach is to always update based on the maximum a posteriori belief, i.e.,

$$l_n^t := \hat{l}_n^t = \operatorname*{argmax}_{k \in \mathcal{Y}} \; p(y_n = k | \hat{y}_n^{0:t}).$$

We refer to this strategy as MaxBelief (MB).

MaxBeliefMinEntropy (MBME) A slightly more sophisticated approach is to also take the change in posterior entropy into account

[†] For example, on ImageNet we have $K = 1000$ which would require estimating 1 million parameters.

and only update when it has decreased:

$$l_n^t := \begin{cases} \hat{l}_n^t & \text{if} \quad S_n^t < S_n^{t-1}, \\ l_n^{t-1} & \text{otherwise.} \end{cases}$$

We refer to this strategy as MaxBeliefMinEntropy (MBME).

CostRatio (CR) So far, we have not taken the assumed larger penalty for negative flips into account. We therefore now develop a third approach based on asymmetric flip costs. We denote the cost of a negative flip (NF) by $c^{\text{NF}} > 0$ and that of a positive flip (PF) by $c^{\text{PF}} < 0$.

We need to decide whether to update the previously stored prediction l_n^{t-1} based on our updated beliefs $p(y_n = k|\hat{y}_n^{0:t})$. Denote the MAP label estimate after step t by

$$\hat{l}_n^t = \text{argmax}_{k \in \mathcal{Y}}\ p(y_n = k|\hat{y}_n^{0:t}).$$

If $\hat{l}_n^t = l_n^{t-1}$ there is no reason to change the stored prediction. Suppose that $\hat{l}_n^t \neq l_n^{t-1}$. We then need to reason about the (estimated) positive and negative flip probabilities when changing the stored prediction from l_n^{t-1} to $\hat{l}_n^t$. A positive flip occurs if $\hat{l}_n^t$ is the correct label (and hence l_n^{t-1} is not):

$$\hat{p}_n^{\text{PF}}(l_n^{t-1} \to \hat{l}_n^t) = p(y_n = \hat{l}_n^t|\hat{y}_n^{0:t})$$

A negative flip occurs if l_n^{t-1} is correct (and hence $\hat{l}_n^t$ is not):

$$\hat{p}_n^{\text{NF}}(l_n^{t-1} \to \hat{l}_n^t) = p(y_n = l_n^{t-1}|\hat{y}_n^{0:t})$$

If neither l_n^{t-1} nor $\hat{l}_n^t$ are the correct label, the flip is inconsequential which we assume incurs zero cost. The estimated cost of changing the stored prediction from l_n^{t-1} to $\hat{l}_n^t$ is thus:

$$\hat{c}(l_n^{t-1} \to \hat{l}_n^t) = c^{\text{NF}}\hat{p}_n^{\text{NF}}(l_n^{t-1} \to \hat{l}_n^t) + c^{\text{PF}}\hat{p}_n^{\text{PF}}(l_n^{t-1} \to \hat{l}_n^t).$$

We only want to change the prediction if $\hat{c}(l_n^{t-1} \to \hat{l}_n^t) < 0$, i.e.,

$$\frac{\hat{p}_n^{\text{PF}}(l_n^{t-1} \to \hat{l}_n^t)}{\hat{p}_n^{\text{NF}}(l_n^{t-1} \to \hat{l}_n^t)} = \frac{p(y_n = \hat{l}_n^t|\hat{y}_n^{0:t})}{p(y_n = l_n^{t-1}|\hat{y}_n^{0:t})} > -\frac{c^{\text{NF}}}{c^{\text{PF}}} \tag{7.3}$$

leading to the following update rule:

$$l_n^t := \begin{cases} \hat{l}_n^t, & \text{if} \quad \hat{l}_n^t = l_n^{t-1}, \\ \hat{l}_n^t, & \text{if} \quad \hat{l}_n^t \neq l_n^{t-1} \wedge \hat{c}(l_n^{t-1} \to \hat{l}_n^t) < 0, \\ l_n^{t-1} & \text{otherwise.} \end{cases}$$

Note that (7.3) has an intuitive interpretation: we only want to update the currently stored prediction (thus potentially risking a

negative flip) if our belief in a different label is larger than that in the current one by a factor exceeding $|c^{\text{NF}}/c^{\text{PF}}|$. We therefore refer to this strategy as CostRatio (CR).

7.4 Discussion: Extensions and Limitations

We discuss current limitations of our method and propose extensions to address them in future work.

Soft vs. Hard Labels Our approach presented in Section 7.3 assumes deterministic classifiers which output hard labels, i.e., only the most likely class. This allows for maximum flexibility and a wide range of classifier models that can be used in conjunction with this method. However, our Bayesian framework can easily be adapted to also allow for probabilistic classifiers which output soft labels, i.e., vectors of class probabilities. We included some additional exploratory experiments utilizing softmax probabilities in Appendix G.11. Since deep neural networks are known to have unreliable uncertainty estimates [310–313], we deliberately choose to work with hard labels. If, however, well-calibrated probabilistic classifiers are available (and can be scaled to huge data sets), taking this additional information into account will likely lead to more accurate posterior estimates and thus better performance.

[310]: MacKay (1995), 'Bayesian neural networks and density networks'
[311]: Szegedy et al. (2014), 'Intriguing properties of neural networks'
[312]: Goodfellow et al. (2015), 'Explaining and harnessing adversarial examples'
[313]: Nguyen et al. (2015), 'Deep neural networks are easily fooled: High confidence predictions for unrecognizable images'

Assumption of Conditionally-Independent Classifiers Since the models $\{C^t\}$ are typically trained and developed on the same data and may even build on insights from prior architectures, our assumption of conditionally independent predictions on $\mathcal{D}^{\text{targ}}$ does likely not hold exactly in practice. It should therefore rather be understood as an approximation that enables tractable posterior inference. Our experiments (Section 7.6) suggest that it is a useful approximation that yields competitive performance. Properly incorporating estimated model correlations may yield further improvements. We refer the interested reader to Appendix G.1 for a more detailed discussion.

Confusion Matrix Estimates Unless labelled data from $\mathbb{P}^{\text{targ}}$ is available, the confusion matrices $\{\pi^t\}$ need to be estimated from $\mathbb{P}^{\text{src}}$. This is only an approximation because they may change as a result of $\mathbb{P}^{\text{src}}_{\mathbf{X}} \neq \mathbb{P}^{\text{targ}}_{\mathbf{X}}$, and taking such shifts into account could yield more accurate posterior estimates. For this, one may use ideas from the field of *unsupervised domain adaptation* [306, 314, 315]. One could use an importance-weighting approach [305] to give more weight to points which are representative of $\mathbb{P}^{\text{targ}}_{\mathbf{X}}$ when estimating π^t from $\mathbb{P}^{\text{src}}_{\mathbf{X},Y}$. As an example, in further experiments in Appendix G.4 we studied estimating the off-diagonal elements using Laplace smoothing [316, 317],

[306]: Sugiyama et al. (2012), *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*
[314]: Pan et al. (2009), 'A survey on transfer learning'
[315]: Ganin et al. (2015), 'Unsupervised domain adaptation by backpropagation'
[305]: Shimodaira (2000), 'Improving predictive inference under covariate shift by weighting the log-likelihood function'
[316]: Good (1953), 'The population frequencies of species and the estimation of population parameters'
[317]: Robbins (1956), 'An empirical Bayes approach to statistics'

Other Selection Strategies Consider an ambiguous image that could be either a zucchini or a cucumber [318]. Such a sample would have large label entropy and could thus potentially be selected for re-evaluation again and again. To overcome this hurdle, one could decompose label uncertainty into epistemic (reducible) and aleatoric (irreducible) uncertainty [319, 320] and only re-evaluate samples with high aleatoric uncertainty, i.e., those with high expected information gain [321]. Such considerations also play a role in the field of *active learning* [322, 323]

[318]: Beyer et al. (2020), 'Are we done with ImageNet?'

[319]: Der Kiureghian et al. (2009), 'Aleatory or epistemic? Does it matter?'

[320]: Kendall et al. (2017), 'What uncertainties do we need in Bayesian deep learning for computer vision?'

[321]: Lindley (1956), 'On a measure of the information provided by an experiment'

[322]: Settles (2009), 'Active learning literature survey'

[323]: Yan et al. (2016), 'Active learning from imperfect labelers'

Growing Data Set Size Our method is not constrained to fixed data set sizes and can accommodate for the addition of new data. New samples can be added at any time using an uniform prior over labels. Given their high initial entropy, they would then be naturally selected for (re-)evaluation first.

Adaptive Budgets Currently, we consider a fixed local budget of B^t re-evaluations at every time step. A possible extension would be to allow for a global budget of B^{total} evaluations spread over all time-steps, i.e., to devise a strategy for deciding whether to (a) keep re-evaluating or (b) save budget for the next better model, potentially using techniques from reinforcement learning [324].

[324]: Sutton et al. (2018), *Reinforcement learning: An introduction*

On the Cost of "Neutral" Flips For simplicity, we have assumed that "neutral" flips (i.e., changing a label estimate from an incorrect to a different incorrect one) bear no cost. However, as motivated in Section 7.1, it is well conceivable that even such neutral flips have a cost due to the potential to disrupt downstream robustness. If this is the case, it can easily be incorporated into our CR update strategy.

7.5 Related Work

Besides the aforementioned connections, our problem setting bears resemblance to several other areas of ML. In the following, we discuss the main differences and commonalities.

Backward Compatibility The term was first introduced by Bansal et al. [301] in the context of humans making decisions based on an AI's prediction (e.g., medical expert systems or driver supervision in semi-autonomous vehicles). They contextualise that even though an AI's predictive performance might increase overall, *incompatible* predictions in updated models severely hurt overall performance and trust, and propose to penalize negative flips w.r.t an older model when training a newer model. Yan et al. [302] show that with standard training, there can be a significant number of negative flips, even if the two models only differ in their random initializations. They then reduce the number of negative flips by

giving more weight to training points that are correctly classified by the reference model, which they call 'positive-congruent training'. Previous work on backward-compatible learning is concerned with training a *new* model. Here, we focus on updating the stored predictions rather than updating the stored models. This makes our approach more generally applicable and complements the use-cases of backward-compatible learning. Backward compatibility was further studied empirically by Srivastava et al. [304] who emphasize that this also causes problems for large multi-component AI systems. They propose two key metrics to characterize backward compatibility: (i) Backward Trust Compatibility (BTC), first mentioned in [301], measuring the fraction of predictions that are still predicted correctly after a model update; and (ii) Backward Error Compatibility (BEC), which corresponds to the probability that an incorrect prediction after an update is not new.

Ensemble Learning Ensemble methods aim to combine several ML models into a single model with higher performance than each of the individual models. Common techniques are boosting [325], bagging [326], or Bayesian model averaging [327]. Our approach falls into the latter category. We compute the posterior probability (7.2) in the same way as the well-known Naive Bayes combiner [328]. The classifier corresponding to our MB strategy goes back to at least Nitzan and Paroush [329] and has been thoroughly analyzed [330]. There are also Bayesian techniques that avoid Assumption 7.3.1, but these either make some parametric assumptions [331] or assume a very special form of dependence [332].

[325]: Freund et al. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting'

[326]: Breiman (1996), 'Bagging predictors'

[327]: Hoeting et al. (1999), 'Bayesian model averaging: a tutorial'

[328]: Kuncheva (2014), *Combining pattern classifiers: methods and algorithms*

[330]: Berend et al. (2015), 'A finite sample analysis of the Naive Bayes classifier.'

[331]: Kim et al. (2012), 'Bayesian classifier combination'

[332]: Boland et al. (1989), 'Modelling dependence in simple and indirect majority systems'

7.6 Experiments

We now evaluate our Bayesian approach to the Prediction Update Problem against different baselines using the task of image classification as a case study.

7.6.1 Experimental Setup

Data Sets We use the three widely accepted benchmark data sets ImageNet1K [333] (1K classes, 50k validation set), ObjectNet [183] (313 classes, 50k validation set) and CIFAR-10 [111] (10 classes, 10k validation set). To imitate our assumed setting of deploying pre-trained models to an unlabelled target data set, we only use the corresponding validation sets as $\mathcal{D}^{\text{targ}}$. The ground truth labels are only used post-hoc to compute performance metrics and are not seen during the T update steps. Of the 313 classes in ObjectNet, 113 are shared with ImageNet, corresponding to a subset of 18,547

[333]: Deng et al. (2009), 'Imagenet: A large-scale hierarchical image database'

[183]: Barbu et al. (2019), 'ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models'

[111]: Krizhevsky et al. (2009), 'Learning multiple layers of features from tiny images'

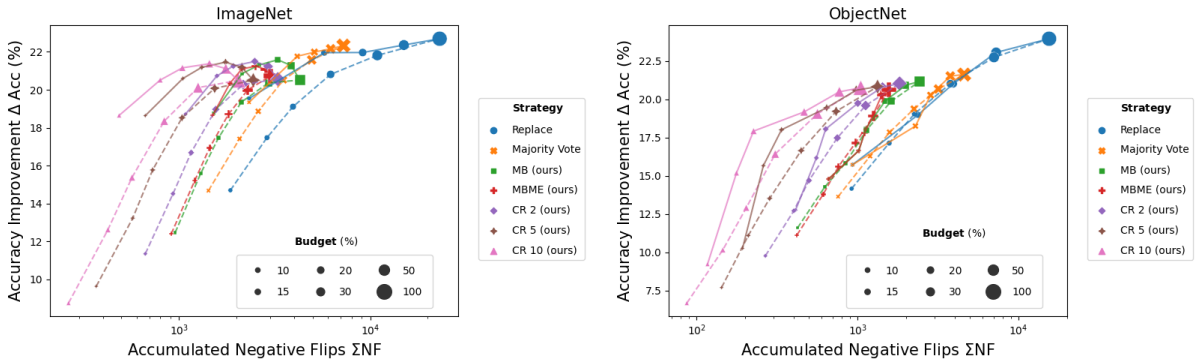


Figure 7.2: Comparison of prediction-update strategies across different budgets after $T = 16$ prediction-update steps for ImageNet → ImageNet (left) and ImageNet → ObjectNet (right). Dashed lines correspond to the ablation using a random selection strategy.

images. ObjectNet images exhibit more realistic variations than those in ImageNet. It only has a test set and thus constitutes a challenging transfer scenario for object recognition models. We deploy ImageNet-pre-trained models both on ImageNet and on the above subset of ObjectNet, thus simulating the cases that the source and target distributions are the same or different, respectively. For the former, we split the ImageNet validation set in half and use one half to estimate π^t and the other as $\mathcal{D}^{\text{targ}}$. For the latter, we estimate π^t from the full ImageNet validation set and evaluate on ObjectNet. We thus assume the covariate shift case, where the conditional label distribution is shared across source and target distributions.

Models & Architectures To emulate the setting of sequentially improving classifiers arriving over time, we use the following 17 models and architectures with many of them setting a new "state-of-the-art" on ImageNet at the time they were first introduced: AlexNet [27]; VGG-11, 13, 16, and 19 [28]; ResNet-18, 34, 50, 101, and 152 [25]; SqueezeNet [334]; GoogLeNet [335]; InceptionV3 [336]; MobileNetV2 [337]; DenseNet-121 and 169 [338], and ResNeXt-101 32x8d [339]. For ease of reproducibility, we use pre-trained models from the torchvision model zoo [340] and [341].

Performance Metrics Recall that our goal is to: (i) improve overall accuracy, (ii) avoid negative flips, and (iii) use as few re-evaluations as possible. To assess these different aspects, we report the following metrics: (i) *final accuracy* of the stored predictions (**Acc**) and *accuracy improvement* over the initial accuracy of C^0 (Δ**Acc**); (ii) the *cumulative number of negative flips* from time $t = 0$ to T ($\sum$ **NF**), the *average negative flip rate* experienced per iteration, i.e., $\frac{\sum \text{NF}}{N \cdot T}$ (**NFR**), and the ratio of accumulated positive to negative flips (**PF / NF**); (iii) the evaluation budget available to each strategy as percentage of the data set size, i.e., a budget of 10 means that 10% of all samples can be re-evaluated at each time step: $B^t = 0.1N, \forall t$; finally, we measure the connective backward compatibility between (i) and

[27]: Krizhevsky et al. (2012), 'Imagenet classification with deep convolutional neural networks'

[28]: Simonyan et al. (2015), 'Very deep convolutional networks for large-scale image recognition'

[25]: He et al. (2016), 'Deep residual learning for image recognition'

[334]: Iandola et al. (2016), 'SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size'

[335]: Szegedy et al. (2015), 'Going deeper with convolutions'

[336]: Szegedy et al. (2016), 'Rethinking the inception architecture for computer vision'

[337]: Sandler et al. (2018), 'Mobilenetv2 Inverted residuals and linear bottlenecks'

[338]: Huang et al. (2017), 'Densely connected convolutional networks'

[339]: Xie et al. (2017), 'Aggregated residual transformations for deep neural networks'

[340]: Paszke et al. (2019), 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'

[341]: Phan (2021), *huyvnphan/PyTorch_CIFAR10*

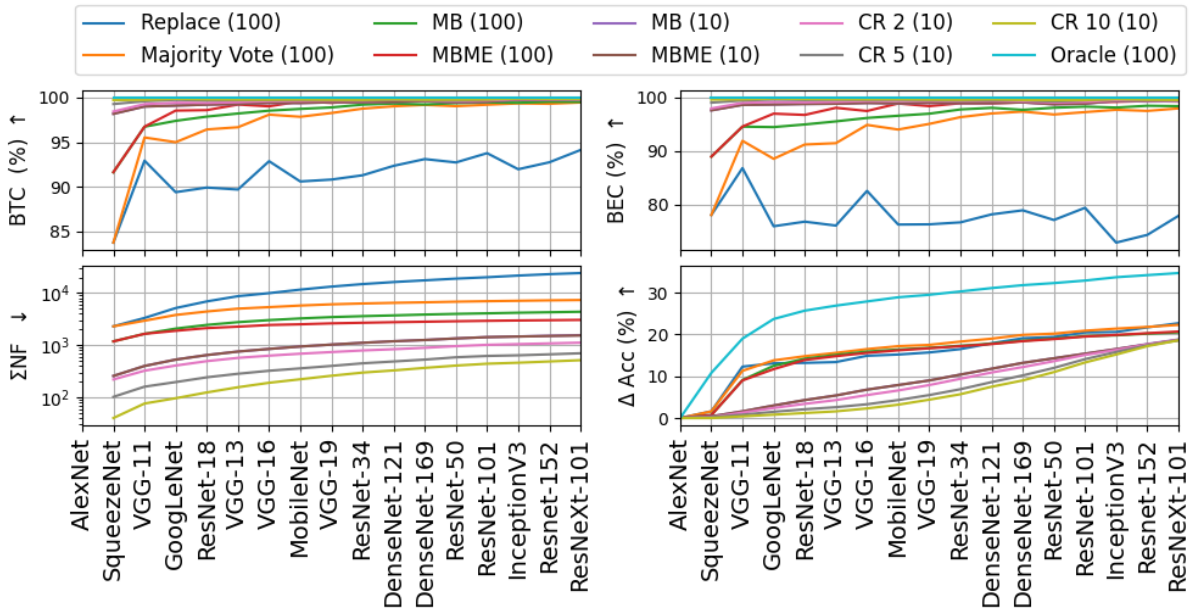


Figure 7.3: Temporal evolution of several key metrics for ImageNet → ImageNet over $T = 16$ prediction-update steps for a subset of strategies and budgets.

(ii) via Backward Trust Compatibility (**BTC**) and Backward Error Compatibility (**BEC**) [304]. We refer to Appendix G.2 for a formal definition of these scores.

Baselines and Oracle We compare our method against two baselines: (i) **Replace** always updates the stored prediction with that predicted by the most recent classifier (a.k.a. *backfilling*); (ii) **Majority Vote** takes into account previous model predictions and updates the stored prediction according to the majority prediction. In case of a tie, the prediction of the most recent classifier is chosen. For reference, we also compare our method against an **Oracle**, which performs a prediction update if and only if this would lead to a positive flip; it thus incurs zero negative flips by definition (knowing the ground truth label). We emphasize that, in practice, we do not have that information in our setting.

Selection- and Prediction-Update Strategies For all methods, we select $B^t \leq N$ samples using the posterior label entropy selection strategy from Section 7.3.2, but also compare with randomly selecting samples for re-evaluation. For all experiments involving the random selection strategies we run the same experiment for five different random seeds each and report the average. We use the prediction-update strategies MB, MBME and CR from Section 7.3.3 and consider cost ratios of $|c^{\text{NF}}/c^{\text{PF}}| \in \{2, 5, 10\}$ for the latter (e.g., CR 2).

Table 7.1: Results for ImageNet → ImageNet: all metrics refer to final performance for the improving model sequence from Figure 7.3. The character **E** or **R** in front of the strategy indicates that selection for re-evaluation is based on the entropy criterion or sampled randomly.

	Strategy	Avg. **BTC** ↑	Avg. **BEC** ↑	**Acc** (%) ↑	Δ**Acc** (%) ↑	Σ NF ↓	NFR (%) ↓	PF / NF ↑
	Oracle	100	100	91.2	34.7	0	0	-
Budget = 100%	Replace	91.37	77.71	**79.2**	**22.7**	24214	6.05	1.2
	Majority Vote	97.18	93.95	78.9	22.3	7352	1.84	1.8
	MB	98.32	96.45	77.1	20.5	4378	1.09	2.2
	MBME	98.78	97.69	77.3	20.7	3057	0.76	2.7
	CR 2	98.72	97.19	77.1	20.6	3368	0.84	2.5
	CR 5	99.06	97.82	77.1	20.5	2520	0.63	3
	CR 10	**99.22**	**98.15**	77	20.5	**2112**	**0.53**	**3.4**
Budget = 30%	R:Replace	97.56	94.53	77.4	20.8	6546.4	1.64	1.8
	R:Majority Vote	98.6	97.18	77.1	20.5	3616.4	0.9	2.4
	E:Replace	96.53	91.01	**78.5**	**22**	9708	2.43	1.6
	E:Majority Vote	98.03	95.63	**78.5**	**22**	5232	1.31	2.1
	E:MB	98.71	97.25	78.1	21.6	3375	0.84	2.6
	E:MBME	98.98	98.04	77.8	21.2	2577	0.64	3.1
	E:CR 2	99.02	97.86	78	21.5	2578	0.64	3.1
	E:CR 5	99.32	98.43	78	21.5	1831	0.46	3.9
	E:CR 10	**99.44**	**98.67**	77.9	21.4	**1517**	**0.38**	**4.5**
Budget = 10%	R:Replace	99.22	98.63	71.3	14.7	1958.4	0.49	2.9
	R:Majority Vote	99.4	98.98	71.2	14.7	1481.4	0.37	3.5
	E:Replace	99.04	98.12	**76.1**	**19.5**	2468	0.62	3
	E:Majority Vote	99.06	98.18	75.9	19.3	2417	0.6	3
	E:MB	99.38	98.89	75.3	18.8	1557	0.39	4
	E:MBME	99.38	98.92	75.2	18.7	1533	0.38	4
	E:CR 2	99.55	99.22	75.3	18.7	1118	0.28	5.2
	E:CR 5	99.72	99.51	75.2	18.6	700	0.18	7.7
	E:CR 10	**99.79**	**99.64**	75.2	18.6	**515**	**0.13**	**10.1**

7.6.2 Results for ImageNet → ImageNet

In Figure 7.3, we show the temporal evolution of backwards compatibility scores, negative flips and accuracy gains for prediction-updates on the ImageNet validation set for a subset of strategies and budgets. A complete account of final performances with additional metrics is shown in Table 7.1.

For the evolution of Δ**Acc** in Figure 7.3, we observe that, unsurprisingly, strategies with 100% budget experience a more rapid gain in accuracy than those with 10%. Among the budget-constrained strategies, the CR strategy with large cost ratio shows the slowest increase, which makes sense as it requires a substantial change in posterior belief for updating a stored prediction and is thus more conservative. Interestingly, however, the *final* accuracies only differ marginally across both strategies and budgets which is also apparent from the minor differences in the Δ**Acc** column of Table 7.1. For the evolution of Σ NF in Figure 7.2 (left), we observe a clear separation of strategies with a natural ordering from least

Table 7.2: Results for ImageNet → ObjectNet: all metrics refer to final performance for the improving model sequence from Figure 7.4 respectively. The character **E** or **R** in front of the strategy indicates that selection for re-evaluation is based on the entropy criterion or sampled randomly.

	Strategy	Avg. **BTC** ↑	Avg. **BEC** ↑	**Acc** (%) ↑	**ΔAcc** (%) ↑	Σ NF ↓	NFR (%) ↓	PF / NF ↑
	Oracle	100	100	50.5	42.6	0	0	-
Budget = 100%	Replace	72.65	92.61	**31.9**	**24**	16669	5.62	1.3
	Majority Vote	89.99	98.02	29.6	21.6	4690	1.58	1.9
	MB	94.46	98.96	29.1	21.2	2477	0.83	2.6
	MBME	95.86	99.34	28.6	20.6	1599	0.54	3.4
	CR 2	95.92	99.21	29	21	1876	0.63	3.1
	CR 5	97.18	99.41	28.8	20.8	1372	0.46	3.8
	CR 10	**97.82**	**99.54**	28.7	20.8	**1084**	**0.37**	**4.6**
Budget = 30%	R:Replace	92.19	98.26	**29**	**21**	4070.6	1.37	2
	R:Majority Vote	94.91	99.02	27.3	19.4	2346.6	0.79	2.5
	E:Replace	91.75	98.14	**29**	**21**	4316	1.45	1.9
	E:Majority Vote	93.54	98.76	28.2	20.3	2970	1	2.3
	E:MB	96.24	99.35	27.8	19.9	1565	0.53	3.4
	E:MBME	96.64	99.48	26.9	18.9	1280	0.43	3.7
	E:CR 2	97.42	99.55	27.7	19.7	1074	0.36	4.4
	E:CR 5	98.43	99.71	27.4	19.4	689	0.23	6.2
	E:CR 10	**98.91**	**99.79**	27.1	19.2	**504**	**0.17**	**8.1**
Budget = 10%	R:Replace	97.49	99.6	22.1	14.2	996.6	0.34	3.6
	R:Majority Vote	97.86	99.68	21.6	13.6	808.8	0.27	4.1
	E:Replace	97.5	99.6	**23.7**	**15.7**	996	0.34	3.9
	E:Majority Vote	97.5	99.6	**23.7**	**15.7**	996	0.34	3.9
	E:MB	98.08	99.72	22.7	14.8	696	0.23	4.9
	E:MBME	98.08	99.72	22.7	14.8	696	0.23	4.9
	E:CR 2	98.68	99.83	20.7	12.8	427	0.14	6.6
	E:CR 5	99.32	99.92	18.2	10.3	197	0.07	10.7
	E:CR 10	**99.57**	**99.95**	17.2	9.2	**122**	**0.04**	**15**

conservative (Replace) to most conservative (CR 10). These relative differences stay mostly constant over time as NFs appear to accumulate approximately linearly (note the log-scale). We find roughly an order of magnitude difference in Σ NF between the best non-probabilistic baseline (Majority Vote) and the best Bayesian method (CR 10). Especially for small budgets of up to 30%, our Bayesian strategies clearly dominate the non-probabilistic baselines both in terms of accuracy and flip metrics, as can be seen from Table 7.1 and Figure 7.2 (left). Moreover, the CR strategy appears to provide control over the number of negative flips via its cost-ratio hyperparameter without adversely affecting final accuracy across a range of budgets, as already observed for a budget of 10% in Figure 7.3. Interestingly, the update rules seem to be optimal when evaluating on less than 100% budget. We attribute this to posterior approximation errors on ImageNet, which is being supported by extensive ablations in the supplement. Regarding backward compatibility (our ultimate goal), we find that **BTC** *and* **BEC** scores reliably outperform the baselines across all budgets. In particular, the CR 10 strategy seems to be especially suitable with

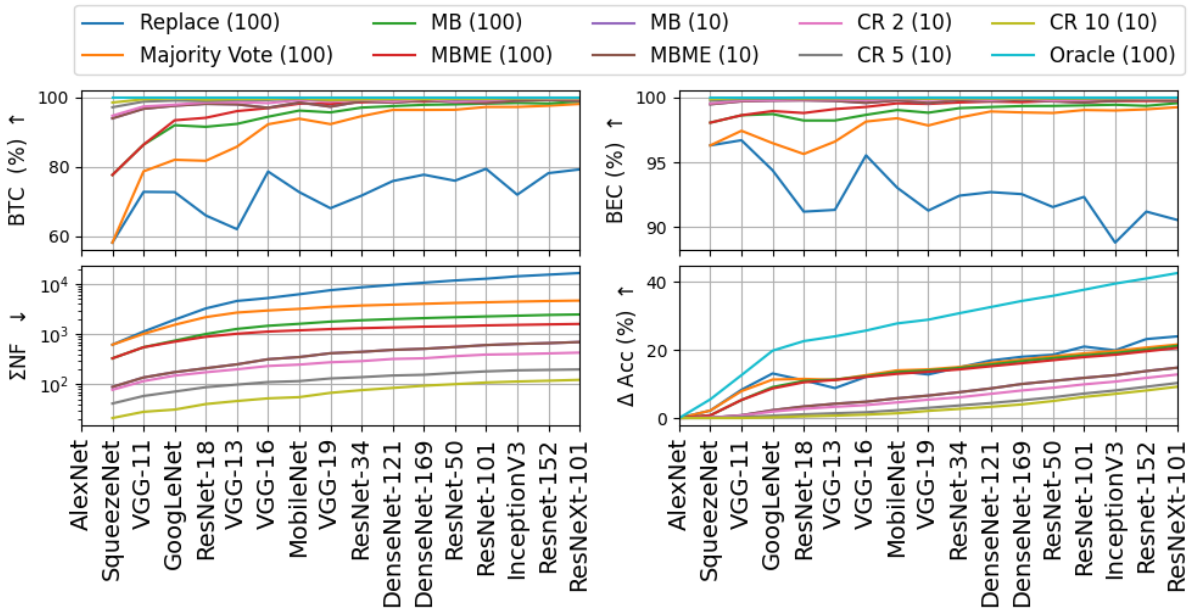


Figure 7.4: Temporal evolution of key metrics for ImageNet → ObjectNet over $T = 16$ prediction-update steps for a subset of strategies and budgets.

scores close to 100%, i.e., oracle performance.

Summary Our method appears to successfully fulfill the three desiderata for backward-compatible prediction-updates in an i.i.d. setting. In particular, our CR strategy seems like the most promising candidate to (i) maintain high accuracy gains and (ii) introduce very few negative flips, when (iii) given only a small compute budget for re-evaluations.

7.6.3 Results for ImageNet → ObjectNet

Results for prediction-updates on ObjectNet are presented (similarly to Section 7.6.2) in Figure 7.4 and Table 7.2 and Figure 7.2 (right). This transfer setting constitutes a much more challenging task. Nevertheless, we observe very similar behaviour to that discussed in Section 7.6.2 and thus only point out the main differences. First, we note that - despite the smaller target data set - the difference in negative flips across different strategies and budgets is even larger on ObjectNet. For example, we observe a reduction in Σ NF of more than two orders of magnitude between Replace (100) and CR (10), and about one order when comparing the two for the same budget. At the same time, differences in accuracy across strategies are also slightly more pronounced, especially for the smallest budget of 10%. Here, the more conservative CR strategies yield lower accuracy gains while MB and MBME maintain competetive accuracy gains. Our strategies are again clearly dominating in terms of backward compatibility w.r.t. BTC and BEC. We remark that these results are agnostic to any potential differences in the label

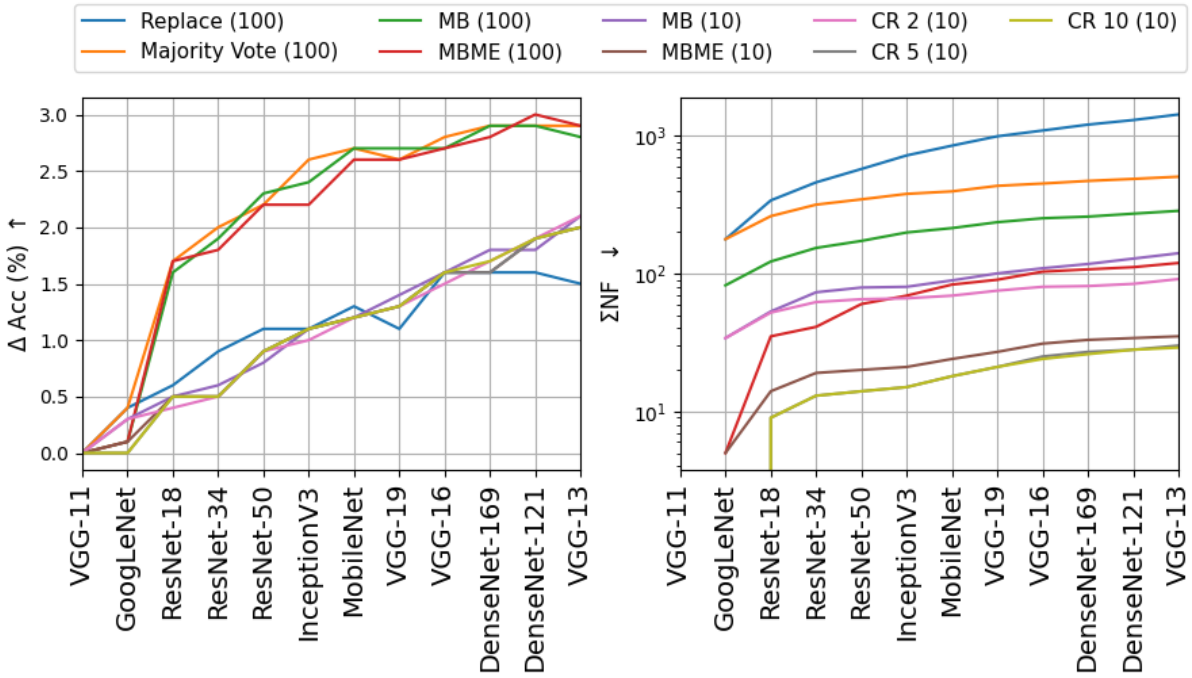


Figure 7.5: Under smaller accuracy gains (experiments on CIFAR-10), we outperform the Replace baseline w.r.t. desideratum (i). Temporal evolution for CIFAR-10 over $T = 11$ prediction-update steps for a subset of strategies and budgets. Our approaches achieve competitive overall accuracy gains (even at 10% of the budget) while introducing only a fraction of negative flips.

space: they are based on a posterior over all 1000 ImageNet classes whereas ObjectNet only contains a subset of 113 of these classes.

Summary Even though the trade-off in accuracy against the back-filling baseline is slightly more pronounced, our approach also fulfills our desiderata in a more challenging transfer scenario. This demonstrates the broad range of relevant future use cases for the Prediction Update Problem.

7.6.4 Further Experiments and Ablations

Results for CIFAR-10 In Figure 7.5 (a) we show the corresponding results on the CIFAR-10 dataset. Here, pre-trained models exhibit a higher level of accuracy ($\approx 93 - 95\%$) and we thus emulate an arguably more realistic scenario with models being released more frequently, thus with smaller accuracy differences. Our method shows very similar trends as we have worked out on ImageNet and ObjectNet.

Interestingly, there is one novel characteristic: due to the presumably less steep increase in accuracy from one model to its successor and fewer class categories, we can form very accurate posterior beliefs which results in accuracy gains of all our methods that even outperform the accuracy-optimizing baselines concerning desider-

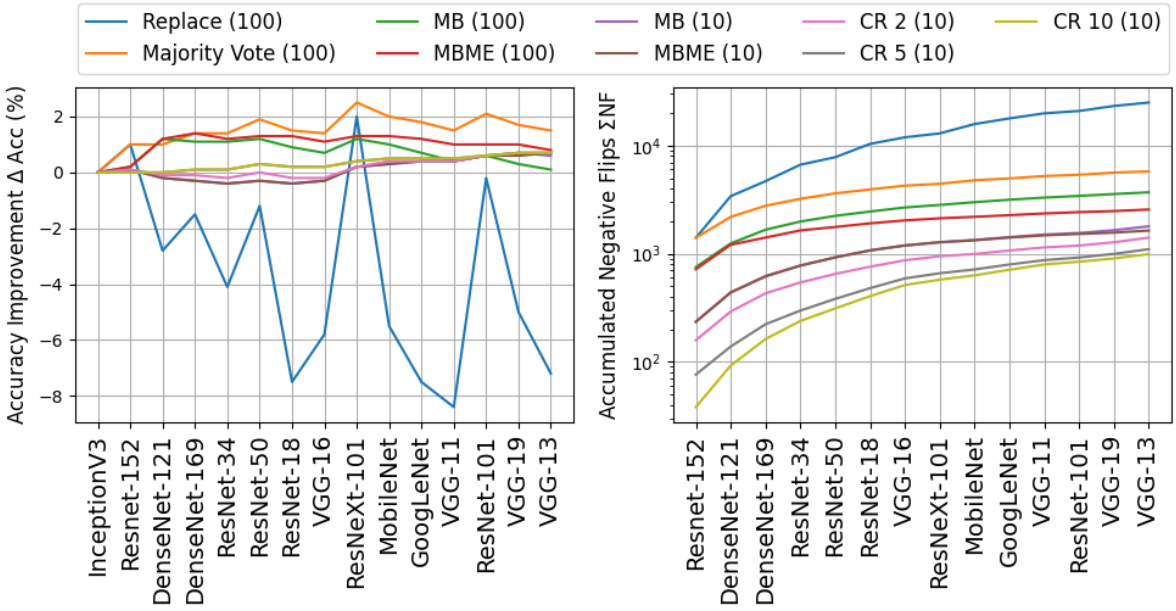


Figure 7.6: Experiment on ImageNet with random model sequence suggesting robustness of our approach to situations where an ordering by performance of may not be available.

atum (i). A more comprehensive analysis of these experiments can be found in Appendix G.5.

Role of the Selection Strategy We also conduct a comparison between our entropy selection and the random baseline for all our methods across a range of budgets - see dashed lines in scatter plots. We find that random selection leads to substantially smaller accuracy gains, but also to fewer negative flips which is intuitive since random selection more often chooses "easy" samples for re-evaluation.

Robustness to Random & Adversary Model Sequences We have assumed that the models C^t are *improving* over time. We thus also consider the scenario where C^t arrive in a *random* or *adversarial* (i.e., strictly deteriorating) order. For the random order (see Figure 7.6 (b)) we find that our methods - unlike, e.g., the replace strategy - achieve strict increases in accuracy while introducing much fewer negative flips. Even in the adversarial case, our methods improve accuracy during the initial steps with much fewer negative flips over the entire history. These findings suggest robustness of our approach to situations where an ordering by performance of C^t may not be available. We refer to Appendix G.3 for more details.

Reducing Re-Evaluations Matters at Scale The re-evaluations of a sample using deep neural network based models clearly dominate computational cost as compared to our method. As an example, the forward pass using the public ImageNet PyTorch models takes up to 550 (biggest VGG and ResNets) times longer than the unoptimized implementation of our method backbone. We analyze this in more detail in Appendix G.9. For very large data sets and with

new models generally increasing in size, reducing the inference budget B is of crucial importance, emphasizing the relevance of desideratum 3.

7.7 Conclusion

The Prediction Update Problem appears frequently in practice and can take different forms. It relates to many different subfields of ML that we have discussed in Section 7.4 and Section 7.5, and there are interesting extensions (structured prediction, adaptive budgets) and improvements (modeling data set structure, across-dataset similarity, domain adaptation, calibration techniques) that need to be worked out. In this work, we have studied the classification case and proposed a Bayesian update rule based on simple assumptions. Empirically, we find improvements along the dimensions we set out to achieve, and we hope that progress on this problem will democratize ML usage even further as it lowers the bar for benefitting from the tremendous progress in model design seen over the last years.

Conclusion 8

8.1 Recap 121
8.2 Key Contributions . . 121
8.3 Broader Context and Future Directions . . . 124
8.4 Personal Reflections . 128

8.1 Recap

Chapters 2 through 5 of this dissertation focused on addressing distribution shifts within real-world scenarios, particularly those with an identifiable underlying structure. We have extensively studied the generalization of neural networks under distribution shifts, aiming to bridge the gap between the constraints of simplistic, idealized benchmarks and practical applications in complex, real-world data environments. Our large-scale empirical studies have not only exposed the capabilities and limitations of current neural network methodologies but also demonstrated various advancements. We have shown how deep neural network models can adapt to complex and often noisy real-world data, marking a step forward from traditional, controlled experimental setups.

Chapters 6 and 7 extended this previous research by exploring learning scenarios that deviate from standard distribution shifts in data generative processes. We proposed a new method to combat continual learning challenges that avoid severe catastrophic forgetting via sparse neural network updates, specifically concerning classification tasks. We discussed problems in scenarios characterized by evolving models and data distributions and proposed approaches that maintain consistency and trust in the predictions of AI systems.

8.2 Key Contributions

Navigating Correlations in Disentangled Representation Learning. In Chapter 2, the research focused on the limitations of current disentanglement methods when applied to realistic data. It was revealed that, despite their potential, these methods struggled to effectively infer and interpret the ground truth factors of more realistic data distributions. Through a combination of empirical studies and theoretical considerations, the chapter demonstrated that current methodologies fall short in unsupervised scenarios, particularly in learning to find and separate underlying factors of variations when correlations between them are significant. However, we also found that limited forms of supervision, such as weak labels, suffice to separate the underlying factors in such disentangled representations. This chapter provided an insight into the latent space of these models that further influenced the

research within Chapters 4 and 5. Apart from this research, several later works have validated these insights since then [8, 64, 342, 343]. Moreover, further works attempted to improve the learning of disentangled representations under correlated structures, e.g., under more relaxed disentanglement conditions bounded to the support of the factors' distribution using Haussdorf distances [344, 345].

Assessing Generalization in RL with Causal Interventions. Chapter 3 introduced *CausalWorld*, a novel benchmark environment for RL within a simulated robotics environment. This environment was designed to facilitate a systematic evaluation of generalization and adaptation abilities in RL agents, particularly under interventions on any underlying environment factors, including the tasks themselves. We highlighted the limitations of prominent RL algorithms, which tended to fail under a wide range of distribution shifts presented in this new benchmark. *CausalWorld* facilitates quantifying adaptation and generalization capabilities in RL, offering a more nuanced understanding of how these agents perform in realistic, dynamically changing environments and their structural causal understanding of it. Beyond building the basis for further work discussed in Chapters 4 and 5, the *CausalWorld* environments have attracted interest across various subsequent studies which have developed methods for challenging and custom-generated tasks [9, 346–349].[1] While our primary goal was to introduce a novel benchmark to evaluate RL agents' generalization capabilities, we encountered limitations in achieving high performance on tasks involving more than two goal objects. This constraint persists, partly due to the dual challenges of mastering precise low-level control over three fingers in the presence of sparse rewards from long-term planning objectives. In contrast, other promising works in this regard such as [350–352] attempt object manipulation tasks by employing basic robot arm grippers, mitigating the complexity of simultaneously learning highly dexterous joint coordination.

Bridging Sim-to-Real Gaps with Disentangled Representations. Building upon the findings of Chapters 2 and 3, Chapter 4 explored the effectiveness of disentangled representation learning approaches in more realistic environments, including sim-to-real shifts. We discovered that existing architectures, developed for simplistic toy benchmarks, needed further scaling to cope with these complex scenarios. Within this chapter, we also identified several predictors that could determine the robustness of representations under sim-to-real transfers beforehand, alleviating the expensive and sometimes impossible nature of model selection. We found that disentanglement appears valuable in mild OOD contexts.

[8]: **Schott** et al. (2022), 'Visual Representation Learning Does Not Generalize Strongly Within the Same Domain'

[64]: Klindt et al. (2021), 'Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding'

[342]: Funke et al. (2022), 'Disentanglement and generalization under correlation shifts'

[343]: Leemann et al. (2022), 'Disentangling embedding spaces with minimal distributional assumptions'

[344]: Wang et al. (2021), 'Desiderata for representation learning: A causal perspective'

[345]: Roth et al. (2023), 'Disentanglement of Correlated Factors via Hausdorff Factorized Support'

[9]: **Mambelli** et al. (2022), 'Compositional Multi-Object Reinforcement Learning with Linear Relation Networks'

[346]: Sontakke et al. (2021), 'Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning'

[347]: Kang et al. (2023), 'Learning Multi-Objective Curricula for Robotic Policy Learning'

[348]: Ding et al. (2022), 'Generalizing goal-conditioned reinforcement learning with variational causal reasoning'

[349]: Allshire et al. (2022), 'Transferring dexterous manipulation from gpu simulation to a remote real-world trifinger'

1: See Appendix A.2 for further details.

[350]: Zadaianchuk et al. (2021), 'Self-supervised Visual Reinforcement Learning with Object-centric Representations'

[351]: Li et al. (2019), 'Towards Practical Multi-Object Manipulation using Relational Reinforcement Learning'

[352]: Seitzer et al. (2021), 'Causal influence detection for improving efficiency in reinforcement learning'

Regarding sim-to-real transfer, however, disentanglement was secondary to a model's performance on approximating the training data, as measured by reconstruction loss or ELBO. **Schott** et al. [8] further validated the influence of a model's architecture and training size for better generalization regarding out-of-distribution factors.[2]

2: See Appendix A.3 for further details.

Pre-trained Representations for Real-World Distribution Shifts. Chapter 5 extended our previous research by examining the transferability of pre-trained representations for the out-of-distribution generalization of RL agents in real-world environments. By conducting a large-scale empirical study, we found that many RL agents leveraging pre-trained structured representations were capable of various forms of OOD generalization. Moreover, it was possible to predict in advance which models would excel and which would not. Specifically, we introduced distribution-specific generalization metrics that can be computed in simulation alone. While our conclusions bear some limitations in exclusively studying frozen VAE representations, leveraging pre-trained encoders for enhanced generalization in downstream RL tasks has become a common theme. It continues benefiting from further scaling and improved encoder architectures [353, 354].

[353]: Taiga et al. (2023), 'Investigating multi-task pretraining and generalization in reinforcement learning'

[354]: Yoon et al. (2023), 'An investigation into pre-training object-centric representations for reinforcement learning'

Addressing Catastrophic Forgetting in Continual Learning. In Chapter 6, the focus shifted to address another crucial shortcoming with current ML models: continual learning under non-stationary distribution shifts, specifically class-incremental learning. A novel neural architecture featuring a discrete key-value bottleneck mechanism was introduced, building upon ideas of discrete representations. The mechanism of the introduced bottleneck allows for sparse, localized, and context-dependent model updates, significantly enhancing the models' adaptability and generalization under changing data distributions. The chapter provided theoretical and empirical evidence to demonstrate the effectiveness of this approach, setting a new state-of-the-art on a challenging continual learning benchmark task. Following its introduction, several studies have built upon the discrete key-value mechanisms to enhance the capability of large models to update information locally when faced with different distribution shifts. This includes extending its application to language tasks using large language models [355, 356] and other areas such as selective model unlearning [12]. Additionally, this work has inspired further research toward solving the grounding problem in object-centric learning [357].

[355]: Hartvigsen et al. (2024), 'Aging with grace: Lifelong model editing with discrete key-value adaptors'

[356]: Dai et al. (2022), 'Lifelong learning for question answering with hierarchical prompts'

[12]: **Shah** et al. (2023), 'Unlearning via Sparse Representations'

[357]: Kori et al. (2024), 'Grounded Object-Centric Learning'

Prediction Updates under Model Shifts. Finally, Chapter 7 deviated from the more traditional problems of dealing with distribution shifts in the input data. Instead, we asked ourselves how to build predictive systems that remain backward compatible when updating them with presumably better ML models. While better models generally improve upon mistakes from previous models, they are, also prone to introducing negative flips, i.e., wrong classifications that did not occur before. We formalized this problem of optimally handling shifts in these output distributions of ML models by introducing the *Prediction Update Problem*. We discussed and presented an effective probabilistic method for simultaneously maintaining computational efficiency and prediction accuracy, especially in avoiding negative flips in predictions. Achieving backward-compatible prediction updates in predictive systems remains an area of high interest [301, 304]. While we have exclusively focused on dealing with scenarios of leveraging readily available pre-trained ML models, another direction of research deals with retraining the neural models themselves towards *backward-compatible representations* [302, 303, 358, 359].

[301]: Bansal et al. (2019), 'Updates in Human-AI teams: Understanding and addressing the performance/-compatibility tradeoff'
[304]: Srivastava et al. (2020), 'An Empirical Analysis of Backward Compatibility in Machine Learning Systems'
[302]: Yan et al. (2021), 'Positive-Congruent Training: Towards Regression-Free Model Updates'
[303]: Shen et al. (2020), 'Towards backward-compatible representation learning'
[358]: Duggal et al. (2021), 'Compatibility-aware heterogeneous visual search'
[359]: Duggal et al. (2022), 'Towards regression-free neural networks for diverse compute platforms'

8.3 Broader Context and Future Directions

While this dissertation could only focus on a small subset of specific aspects of generalization, advances along the entire spectrum of distribution shifts are of high interest. These aspects range from tackling out-of-domain generalization to addressing the intricacies of adversarial attacks [168, 360, 361]. Along each of these directions, ML has made significant advances in recent years, both empirically and practically. In this context, self-supervised learning (SSL) is probably the most relevant approach towards improved generalization [362–364]. A fundamental principle underlying these self-supervised learning approaches is to augment training data and construct almost-free data labels to obtain more learning signals. This way, models can acquire improved representations of the data distributions' underlying structure, enabling subsequent adaptation with only a few or almost no additional examples. In addition, the development and utilization of systematic, challenging benchmarks for distribution shifts have been instrumental in better understanding a wide range of generalization capabilities of proposed ML models. These benchmarks not only test the limits of current models under varied conditions but they also serve as catalysts for ongoing refinement [42, 182, 187, 188, 365–367].

[168]: Arjovsky et al. (2019), 'Invariant risk minimization'
[360]: Shafahi et al. (2019), 'Adversarial training for free!'
[361]: Schneider et al. (2020), 'Improving robustness against common corruptions by covariate shift adaptation'
[362]: Chen et al. (2020), 'A simple framework for contrastive learning of visual representations'
[363]: Misra et al. (2020), 'Self-supervised learning of pretext-invariant representations'
[364]: Grill et al. (2020), 'Bootstrap your own latent-a new approach to self-supervised learning'

Pre-training and Fine-Tuning of Transformers: A Paradigm Shift

Besides advances in self-supervised learning, machine learning has witnessed a major paradigmatic shift with the introduction of the Transformer architecture [26]. Since their introduction, Transformers have demonstrated unparalleled performance in handling any input and output modalities that can be transformed into sets, making them a general-purpose architecture across various domains like vision, language, and robotics. The key to their success lies in their ability to capture long-ranging and complex dependencies in data through self-attention layers, which is further enhanced by a pre-training stage. Because of their advantageous scaling behaviors, pre-training on enormous, diverse datasets equips Transformer models with a comprehensive understanding of the observed data. Common strategies include pre-training for reconstruction of masked images or auto-regressive next-token prediction in language tasks [233, 368–373]. Because the data representations learned by these models cover a vast and rich spectrum of potentially relevant features, the fine-tuning phase of Transformers to adapt them for specialized tasks often requires less additional training data than training a specialist model from scratch. This is particularly evident in large language models (LLMs), where fine-tuning has enabled models to achieve remarkable success across a wide range of language tasks [31–33, 374]. Such adaptability used to require extensive computing resources, but concurrent fine-tuning advancements like parameter-efficient fine-tuning enabled these multi-billion parameter models to specialize on new tasks very quickly [375, 376]. As these models scale, they start to demonstrate a remarkable ability to generalize across a broad spectrum of tasks and distributions – even without any further task fine-tuning – but just by in-context learning through literal task specification and optional few-shot examples at test-time [32]. In that context, such foundation models are trained on training data scales that blur the line between their i.i.d. and out-of-distribution generalization capabilities. The pre-training and fine-tuning paradigm has also significantly impacted the approach to model development. Instead of building models from scratch for each new task, as was the case just a few years ago, the focus has shifted to developing robust, general-purpose base models that can be fine-tuned to specific needs. This paradigm, which has become increasingly accepted as a standard practice in ML research, saves resources and allows for efficient re-utilization of shared information across tasks and domains.

[42]: Hendrycks et al. (2021), 'The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization'
[182]: Koh et al. (2021), 'Wilds: A benchmark of in-the-wild distribution shifts'
[187]: Hendrycks et al. (2019), 'Benchmarking Neural Network Robustness to Common Corruptions and Perturbations'
[188]: Michaelis et al. (2019), 'Benchmarking robustness in object detection: Autonomous driving when winter is coming'
[365]: Peng et al. (2017), 'Visda: The visual domain adaptation challenge'
[366]: Croce et al. (2020), 'Robustbench: a standardized adversarial robustness benchmark'
[367]: Gürtler et al. (2023), 'Real robot challenge 2022: Learning dexterous manipulation from offline data in the real world'
[26]: Vaswani et al. (2017), 'Attention is all you need'
[233]: Caron et al. (2021), 'Emerging Properties in Self-Supervised Vision Transformers'
[368]: Carion et al. (2020), 'End-to-end object detection with transformers'
[369]: He et al. (2022), 'Masked autoencoders are scalable vision learners'
[370]: Zhang et al. (2022), 'Opt: Open pre-trained transformer language models'
[371]: Oquab et al. (2023), 'Dinov2: Learning robust visual features without supervision'
[372]: Kirillov et al. (2023), 'Segment anything'
[373]: Driess et al. (2023), 'Palm-e: An embodied multimodal language model'
[31]: Radford et al. (2019), 'Language models are unsupervised multitask learners'
[32]: Brown et al. (2020), 'Language models are few-shot learners'
[33]: Achiam et al. (2023), 'Gpt-4 technical report'
[374]: Touvron et al. (2023), 'Llama: Open and efficient foundation language models'

Remaining Challenges and Future Directions

While Transformers, and LLMs in particular, demonstrate impressive capabilities in generalizing across various complex tasks, our understanding of the underlying mechanisms that enable such achievements remains elusive. Questions about the specific representations and relations these models learn and how they utilize these for predictions are still open for exploration. Despite their capabilities, these models can sometimes confidently produce incorrect predictions, particularly in scenarios that involve complex planning and reasoning on out-of-distribution data sets [377–379]. This highlights a gap in our comprehension of their operational principles and general reliability.

However, these models' overall remarkable generalization and transfer abilities suggest a deeper underlying mechanism at play. It prompts us to consider whether LLMs, through their extensive training on vast datasets, have managed to distill robust and stable patterns that mirror causal mechanisms more closely than what is achievable by smaller models [380–382]. If confirmed, this emergent phenomenon of implicitly learned (causal) priors could further aid our approach to enhancing model robustness. By isolating and comprehending such robust features, we could further enhance the models' ability to generalize under distributional changes, shifting focus from traditional methods that heavily depend on embedding strict inductive biases or causal frameworks into the neural structures. This shift implies that major breakthroughs in architecture or training techniques might not be essential in the short term; instead, a deeper understanding of the knowledge acquired by these models becomes crucial. Such a deepened mechanistic insight could improve the interpretability and explainability of ML models' predictions and help mitigate their failures. Further, a better understanding of the latent representations within these models could enable precise editing or removal of incorrect or sensitive information, marking this an important area for future research to ensure their reliability.

As the ML research field witnesses continuing advancements from merely scaling up architectures and training them with increasingly vast data sets, we likely approach a potential saturation point of publicly available data for pre-training. While synthetic data and improved adaptation techniques might still further augment their capabilities [383], prevailing constraints imposed by high-quality data availability in various fields remain. This observation underlines the importance of innovating beyond mere reliance on data and compute resources.

[375]: Hu et al. (2022), 'LoRA: Low-Rank Adaptation of Large Language Models'
[376]: Xu et al. (2023), 'Parameter-efficient fine-tuning methods for pre-trained language models: A critical review and assessment'
[32]: Brown et al. (2020), 'Language models are few-shot learners'

[377]: Bubeck et al. (2023), 'Sparks of artificial general intelligence: Early experiments with gpt-4'
[378]: Jin et al. (2023), 'Cladder: Assessing causal reasoning in language models'
[379]: Liu et al. (2023), 'Evaluating the logical reasoning ability of chatgpt and gpt-4'

[380]: Park et al. (2023), 'The Linear Representation Hypothesis and the Geometry of Large Language Models'
[381]: Kıcıman et al. (2023), 'Causal reasoning and large language models: Opening a new frontier for causality'
[382]: Wei et al. (2022), 'Emergent abilities of large language models'

[383]: Anthropic (2024), 'The Claude 3 Model Family: Opus, Sonnet, Haiku'

For example, in robotics, the scarcity of training data remains starkly contrasted to what is available for vision and language processing tasks. This limitation has led to only moderate success in navigating complex benchmark environments, with a persistent gap in RL agents' ability to fully comprehend and adapt to the real world's dynamic unpredictability. Similarly, the fields of medicine and biology face unique challenges, primarily due to the reliance on data predominantly sourced from populations in industrialized countries. This bias results in genetic and ethnic ancestry biases, e.g., within large-scale Genome-Wide Association Studies [384]. Additionally, the genetic data per individual alone is substantial, exceeding the input processing capabilities of current state-of-the-art ML models by far. Such constraints, coupled with the potential for misleading correlations in treatment outcomes, present a range of significant challenges. Nonetheless, our goal remains to ensure everyone benefits equally from advancements in predictive ML models, irrespective of the distributional biases inherent in the training data [13].[3]

[384]: Fitipaldi et al. (2023), 'Ethnic, gender and other sociodemographic biases in genome-wide association studies for the most burdensome non-communicable diseases: 2005–2022'

[13]: **Träuble** et al. (2025), 'Multi-Megabase Scale Genome Interpretation with Genetic Language Models'

3: See Appendix A.1 for further details.

These high-stakes applications illustrate the necessity for further research on models that can reliably generalize and adapt to the inevitable distribution shifts in real-world data. As such, despite the substantial progress made in machine learning, there remains an urgent and ongoing need for methodological innovations to enhance model generalization capabilities. This need is particularly pronounced in the face of distribution shifts, a challenge that remains a significant open problem, especially when we use these methods across high-stakes applications.

8.4 Personal Reflections

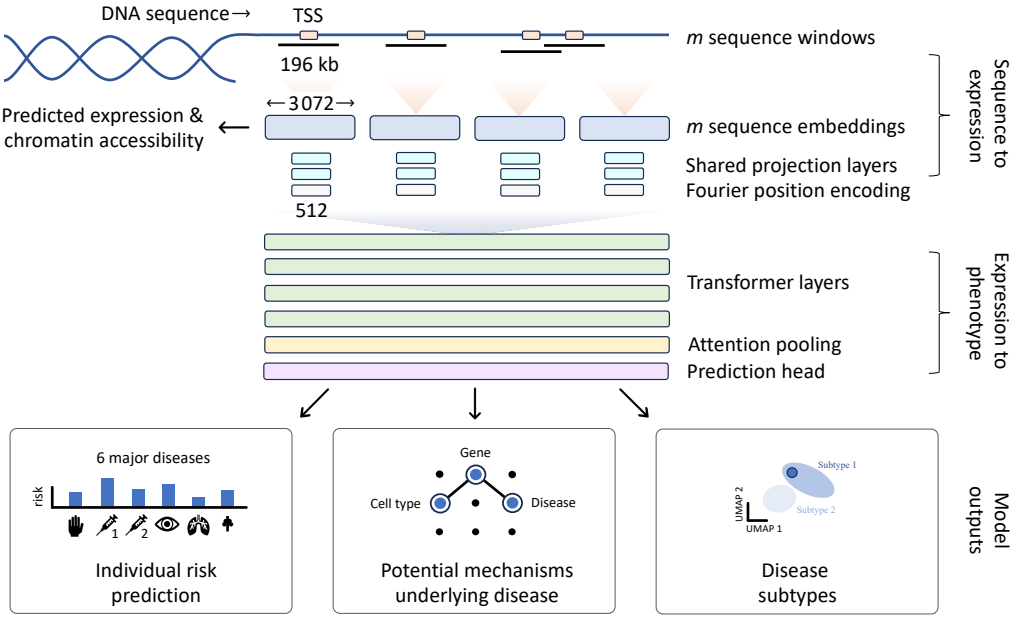
As I reflect on my time throughout my Ph.D., I am amazed by the continued and profound advancements over the last few years. When I started, the field of AI was already attracting considerable interest, primarily fueled by breakthroughs in deep learning on well-defined isolated tasks. Despite all this, open challenges appeared to be a long-term goal to solve, particularly those around enhancing generalization and adaptability beyond narrowly defined tasks. In hindsight, the pace and magnitude of progress along these directions in recent years have exceeded any of my expectations. My motivation to work on these topics was driven by a hope that the methods we develop could become catalysts one distant day, unlocking potentials in domains like health care or robotics. Today, I find myself at a point where this is not just a hope but a developing possibility. The advancements in AI, some of which I have have been fortunate to witness in my research, have been monumental.

In closing and trying to formulate my expectations for the future, I am reminded of a quote by Alan Kay: "The best way to predict the future is to invent it." In ML research nowadays, we are not just passive observers; we are an active part of how the future unfolds, shaping it to a very tiny bit with every line of code we write, every model we build, and every hypothesis we test.

Appendix

Related Projects A

A.1 Multi-Megabase Scale Genome Interpretation with Genetic Language Models



Multi-Megabase Scale Genome Interpretation with Genetic Language Models

Frederik Träuble*, Lachlan Stuart*, Andreas Georgiou*, Pascal Notin, Arash Mehrjou, Ron Schwessinger, Mathieu Chevalley, Kim Branson, Bernhard Schölkopf, Cornelia van Duijn, Debora Marks and Patrick Schwab.

Träuble et al. [13]

Abstract

Understanding how molecular changes caused by genetic variation drive disease risk is crucial for deciphering disease mechanisms. However, interpreting genome sequences is challenging because of the vast size of the human genome, and because its consequences manifest across a wide range of cells, tissues and scales - spanning from molecular to whole organism level. Here, we present Phenformer, a multi-scale genetic language model that learns to generate mechanistic hypotheses as to how differences in genome sequence lead to disease-relevant changes in expression across cell types and tissues directly from DNA sequences of up to 88 million base pairs. Using whole genome sequencing data from more than `150000` individuals, we show that Phenformer generates mechanistic hypotheses about disease-relevant cell and tissue types that match literature better than existing state-of-the-art methods, while using only sequence data. Furthermore, disease risk predictors enriched by Phenformer show improved prediction performance and generalisation

* Shared First Author. Equal contribution.

to diverse populations. Accurate multi-megabase scale interpretation of whole genomes without additional experimental data enables both a deeper understanding of molecular mechanisms involved in disease and improved disease risk prediction at the level of individuals.

Summary and connection to this dissertation

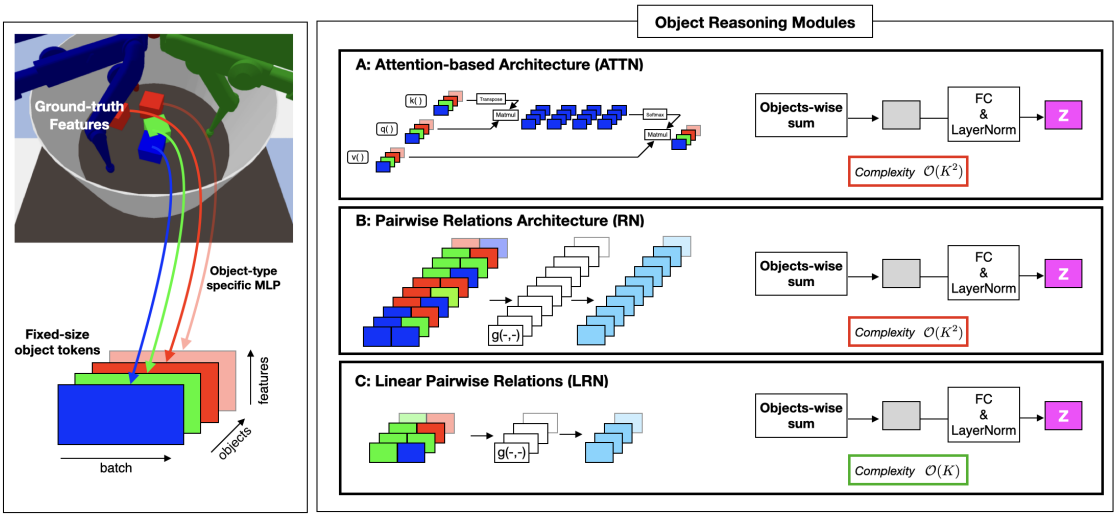
The above work introduces Phenformer, a novel transformer-based neural architecture that leverages unparalleled coverage of individuals' genomic sequences (almost 3% of the genome) to predict disease risk directly. Unlike Genome-Wide Association Studies (GWAS) and derived Polygenic Risk Scores (PRS) that rely on elementary regression model classes which often overfit on non-causal nucleotide polymorphisms within ancestry-biased training data [385–389], Phenformer utilizes gene-expression feature embeddings from a state-of-the-art gene expression model [38]. This approach not only achieves superior risk prediction across six major diseases but also enhances interpretability by providing mechanistic insights into the disease-driving cellular processes.

Phenformer mirrors the information flow in molecular biology. By starting from gene expression embeddings, the model operates on intermediate cellular-grounded processes, which can be associated with low-level mechanisms of disease development. This bottom-up approach enables Phenformer to learn complex relationships between gene expressions and disease outcomes, offering a more fine-grained framework to form hypotheses of genetic influences on disease than GWAS and associated PRS scores.

Furthermore, Phenformer addresses a major limitation of GWAS and existing PRS methods: their poor transferability across various ethnic population groups as training data sets are predominantly composed of individuals of European descent. By demonstrating enhanced performance in predicting risks for non-European ancestries, Phenformer indicates an ability to learn more robust and biologically grounded mechanisms, underscoring its potential for broad applicability in diverse populations.

This work aligns with the core objectives of the dissertation by introducing a first-of-its-kind powerful model class that not only pushes the boundaries of disease risk prediction but also addresses the generalization to distribution shifts in genomic data characterized by varied gene heritage in different ethnicities. The incorporation of Phenformer into this research represents another step towards ML models capable of navigating the complexities of real-world distribution shifts. Ideally, the method contributes to the broader field of genetic research with implications for personalized medicine and understanding disease mechanisms at an individual level.

A.2 Compositional Multi-Object Reinforcement Learning with Linear Relation Networks



Compositional Multi-Object Reinforcement Learning with Linear Relation Networks

Davide Mambelli*, **Frederik Träuble***, Stefan Bauer†, Bernhard Schölkopf† and Francesco Locatello†.

ICLR 2022 Workshop: Workshop on the Elements of Reasoning: Objects, Structure and Causality.

Mambelli et al. [9]

Abstract

Although reinforcement learning has seen remarkable progress over the last years, solving robust dexterous object-manipulation tasks in multi-object settings remains a challenge. In this paper, we focus on models that can learn manipulation tasks in fixed multi-object settings and extrapolate this skill zero-shot without any drop in performance when the number of objects changes. We consider the generic task of bringing a specific cube out of a set to a goal position. We find that previous approaches, which primarily leverage attention and graph neural network-based architectures, do not generalize their skills when the number of input objects changes while scaling as K^2. We propose an alternative plug-and-play module based on relational inductive biases to overcome these limitations. Besides exceeding performances in their training environment, we show that our approach, which scales linearly in K, allows agents to extrapolate and generalize zero-shot to any new object number.

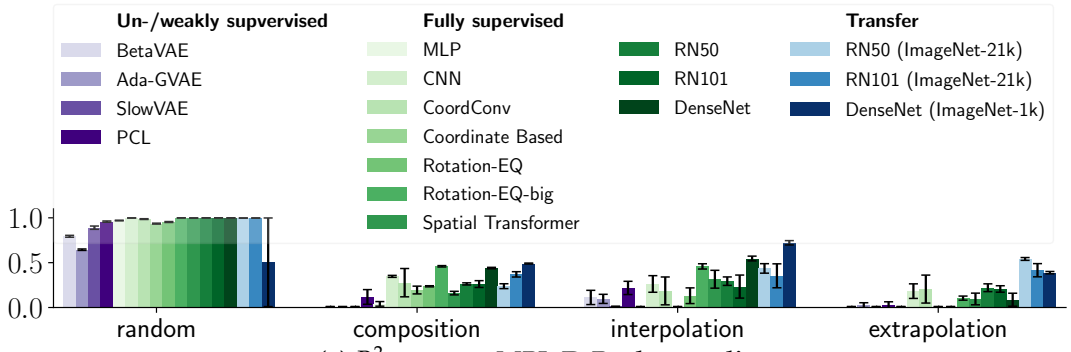
* Shared First Author. Equal contribution.
† Joint Senior Authors. Equal Advising.

Summary and connection to this dissertation

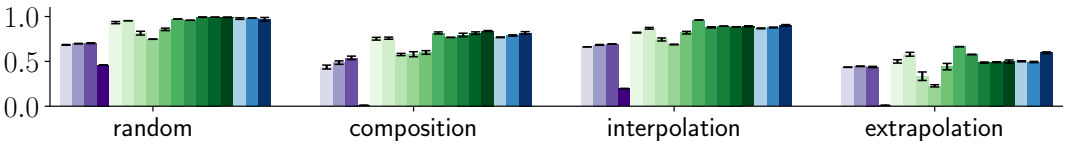
Within this dissertation, our exploration has predominantly focused on the challenge of learning single object manipulation tasks within the TriFinger setup. The research presented in the above work extends this on generalization of RL agents within multi-object environment. In detail, we address the issue of generalizing across a varying number of objects. Specifically, we explore a task of relocating a specified object from a group of objects to a target location in the CausalWorld environment. Unlike humans, who can adaptively solve such tasks involving different numbers of cubes (e.g., moving from three cubes to handling six or ten cubes without a noticeable drop in performance), RL agents have not demonstrated this form of zero-shot extrapolation generalization with changes in object count. Our findings indicate that architectures based on attention mechanisms or graph neural networks, despite their capacity to handle variable multi-object states, exhibit diminishing generalization capabilities as the number of objects in the test environment varies.

To address these limitations, we propose the integration of relational inductive biases that can capture the stable and shared relations between objects. This is achieved through the development of a novel linear relation network encoder architecture. Firstly, we show that the architecture demonstrates robust zero-shot generalization across a broad spectrum of object counts without significant performance degradation. Moreover, the proposed architecture also offers enhanced computational efficiency with a linear complexity $O(K)$ in contrast to the quadratic complexity $O(K^2)$ observed in previous approaches. Lastly, the intermediate representations generated by our model exhibit an interpretative encoder manifold. This work contributes to the overarching theme of the dissertation by examining another dimension of real-world distribution shifts - namely, the adaptation of neural architectures to accommodate a fluctuating set of input objects within a simplified, yet realistic robotic context.

A.3 Visual Representation Learning Does Not Generalize Strongly Within the Same Domain



(a) R^2-score on MPI3D-Real test splits.



(b) R^2-score on CelebGlow test splits.

Visual Representation Learning Does Not Generalize Strongly Within the Same Domain

Lukas Schott, Julius von Kügelgen, **Frederik Träuble**, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello[†] and Wieland Brendel[†].

ICLR 2022: International Conference on Learning Representations.

Schott et al. [8]

Abstract

An important component for generalization in machine learning is to uncover underlying latent factors of variation as well as the mechanism through which each factor acts in the world. In this paper, we test whether 17 unsupervised, weakly supervised, and fully supervised representation learning approaches correctly infer the generative factors of variation in simple datasets (dSprites, Shapes3D, MPI3D) from controlled environments, and on our contributed CelebGlow dataset. In contrast to prior robustness work that introduces novel factors of variation during test time, such as blur or other (un)structured noise, we here recompose, interpolate, or extrapolate only existing factors of variation from the training data set (e.g., small and medium-sized objects during training and large objects during testing). Models that learn the correct mechanism should be able to generalize to this benchmark. In total, we train and test 2000+ models and observe that all of them struggle to learn the underlying mechanism regardless of supervision signal and architectural bias. Moreover, the generalization capabilities of all tested models drop significantly as we move from artificial datasets towards more realistic real-world datasets. Despite their inability to identify the correct mechanism, the models are quite modular as their ability to infer other in-distribution factors remains fairly stable, providing only a single factor is out-of-distribution. These results point

[†] Joint Senior Authors. Equal Advising.

to an important yet understudied problem of learning mechanistic models of observations that can facilitate generalization.

Summary and connection to this dissertation

This study relates with the discussions in Chapter 2 and Chapter 4 of the dissertation, exploring the generalization capabilities of widely used representation learning strategies across visual datasets. A large-scale investigation presented in this work broadens the scope beyond disentanglement learning to include contrastive and supervised learning methods. We also investigated transfer learning and the impact of pre-training on large datasets such as ImageNet. Through a comprehensive empirical analysis, we assess various forms of generalization—interpolation, recombination, and extrapolation—beyond the conventional independent and identically distributed (i.i.d.) scenarios. This research substantially broadens the understanding established by prior, smaller-scale studies such as Montero et al. [390]. The analysis highlights persistent generalization challenges encountered by all models, particularly with the transition to more realistic datasets. To stimulate further progress in addressing these generalization challenges, we introduced a new dataset as a benchmark for the research community. This work not only reinforces the findings related to disentanglement learning discussed in Chapter 2 and Chapter 4 but also underscores the role of pre-training to tackle the complexities of real-world datasets effectively.

Appendix to Chapter 2 B

B.1 Proof of Proposition

Let $f^* : \mathcal{C} \to \mathcal{Z}$ be a diffeomorphism that transforms $p^*(\mathbf{c})$ (the true prior distribution of the factors of variation) into the fixed model prior $p(\mathbf{z})$:

$$p^{*(\mathbf{c})}(\mathbf{c}) = p^{(\mathbf{z})}(f^*(\mathbf{c}))\,|\det \nabla_{\mathbf{c}} f^*(\mathbf{c})|\ . \tag{B.1}$$

Superscripts are included here to clarify which random variable a distribution is defined over, but will be often omitted. The existence of f^* implies that the true generative model $p^*(\mathbf{x} \mid \mathbf{z})$ can be expressed as a composition of a deterministic transformation, $(f^*)^{-1}$, and a stochastic one, $p^*(\mathbf{x} \mid \mathbf{c})$.

Similarly, let $f_\theta : \mathcal{C} \to \mathcal{Z}$ be a diffeomorphism, parameterized by θ, that defines a distribution $p_\theta(\mathbf{c})$ (in general different from $p^*(\mathbf{c})$) with support $\mathcal{C}$:

$$p_\theta^{(\mathbf{c})}(\mathbf{c}) = p^{(\mathbf{z})}(f_\theta(\mathbf{c}))\,|\det \nabla_{\mathbf{c}} f_\theta(\mathbf{c})|\ . \tag{B.2}$$

We will assume that the learned generative model $p_\theta(\mathbf{x} \mid \mathbf{z})$ can be expressed as a composition of the learned deterministic transformation f_θ^{-1} and the true $p^*(\mathbf{x} \mid \mathbf{c})$:

$$\begin{aligned} p_\theta(\mathbf{x}) &= \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \\ &= \int_{\mathbf{c}} p_\theta(\mathbf{x} \mid f_\theta(\mathbf{c})) p_\theta(\mathbf{c}) d\mathbf{c} \\ &= \int_{\mathbf{c}} p^*(\mathbf{x} \mid \mathbf{c}) p_\theta(\mathbf{c}) d\mathbf{c} \end{aligned} \tag{B.3}$$

where $\mathbf{c} = f_\theta^{-1}(\mathbf{z})$. Note that the distribution $p_\theta(\mathbf{c})$ is implicitly learned by learning f_θ, and it represents the learned prior distribution over the true factors of variation.

The expected log likelihood we wish to maximize is

$$\mathbb{E}_{p^*(\mathbf{x})}[\log p_\theta(\mathbf{x})] = -\mathcal{H}(p^*(\mathbf{x})) - D_{\mathrm{KL}}(p^*(\mathbf{x}) \| p_\theta(\mathbf{x}))$$

where the differential entropy $\mathcal{H}(p^*(\mathbf{x}))$ is constant with respect to the model parameters, and can

therefore be ignored. The KL term can be rewritten as

$$\begin{aligned} D_{\mathrm{KL}}(p^*(\mathbf{x})\|p_\theta(\mathbf{x})) &= \int_{\mathbf{x}} p^*(\mathbf{x}) \log \frac{p^*(\mathbf{x})}{p_\theta(\mathbf{x})} d\mathbf{x} \\ &= \int_{\mathbf{x}} \int_{\mathbf{c}} p^*(\mathbf{x} \mid \mathbf{c}) p^*(\mathbf{c}) \log \frac{p^*(\mathbf{c}) \frac{p^*(\mathbf{x} \mid \mathbf{c})}{p^*(\mathbf{c} \mid \mathbf{x})}}{p_\theta(\mathbf{c}) \frac{p^*(\mathbf{x} \mid \mathbf{c})}{p^*(\mathbf{c} \mid \mathbf{x})}} d\mathbf{c}\, d\mathbf{x} \\ &= \int_{\mathbf{x}} \int_{\mathbf{c}} p^*(\mathbf{x} \mid \mathbf{c}) p^*(\mathbf{c}) \log \frac{p^*(\mathbf{c})}{p_\theta(\mathbf{c})} d\mathbf{c}\, d\mathbf{x} \\ &= \int_{\mathbf{c}} p^*(\mathbf{c}) \log \frac{p^*(\mathbf{c})}{p_\theta(\mathbf{c})} \int_{\mathbf{x}} p^*(\mathbf{x} \mid \mathbf{c}) d\mathbf{x}\, d\mathbf{c} \\ &= D_{\mathrm{KL}}(p^*(\mathbf{c})\|p_\theta(\mathbf{c}))\,. \end{aligned} \tag{B.4}$$

Note that, since the KL divergence is always non-negative, the maximum likelihood corresponds to $D_{\mathrm{KL}}(p^*(\mathbf{x})\|p_\theta(\mathbf{x})) = 0$.

Let a matrix be σ-diagonal if there exists a permutation σ that makes it diagonal. Since by assumption $p^*(\mathbf{c})$ does not factorize while $p(\mathbf{z})$ does, it follows that $\nabla_{\mathbf{c}} f^*(\mathbf{c})$ (the Jacobian of f^*) is not σ-diagonal.* However, if the representations $\mathbf{z}$ are disentangled w.r.t. the true factors $\mathbf{c}$ then the Jacobian of f_θ is σ-diagonal. Thus, $f_\theta(\mathbf{c})$ cannot be equal to $f^*(\mathbf{c})$ almost everywhere. This in turn means that $D_{\mathrm{KL}}(p^*(\mathbf{c})\|p_\theta(\mathbf{c})) > 0$, hence $\mathbb{E}_{p^*(\mathbf{x})}[\log p_\theta(\mathbf{x})] < \mathbb{E}_{p^*(\mathbf{x})}[\log p^*(\mathbf{x})]$. This proves that if the generative model is disentangled w.r.t. the true factors then its expected likelihood is less than the optimal likelihood.

On the other hand, in the general case in which the representations are not necessarily disentangled, we can choose θ such that $f_\theta(\mathbf{c}) = f^*(\mathbf{c})$ almost everywhere, which implies that $D_{\mathrm{KL}}(p^*(\mathbf{c})\|p_\theta(\mathbf{c})) = 0$. Thus, there exists an entangled model that has optimal likelihood.

We have proved that (i) if the generative model is constrained to be disentangled then the optimal likelihood cannot be achieved, and (ii) if it is *not* constrained to be disentangled then the optimal likelihood *can* be achieved. Equivalently, the optimal likelihood can be attained if and only if the generative model is entangled w.r.t. the true generative factors.

B.2 Implementation Details

Unsupervised Disentanglement methods. For the sake of comparison, the considered disentanglement methods in this work cover the full collection of state-of-the-art approaches in `disentanglement_lib` from [65] based on representations learned by VAEs. The set contains six different methods that enforce disentanglement of the representation by equipping the loss with different regularizers that aim at enforcing the special structure of the posterior aggregate encoder distribution. A detailed description of the regularizer forms used in this work, specifically β-VAE [57], FactorVAE [59], AnnealedVAE [58], DIP-VAE-I, DIP-VAE-II [61] and β-TC-VAE [60] is provided in Locatello et al. [65]. We use the same encoder and decoder architecture with 10 latent dimensions for every model.

* If $p(\mathbf{z})$ factorizes and $\nabla_{\mathbf{c}} f^*(\mathbf{c})$ is σ-diagonal, then $p^*(\mathbf{c})$ also factorizes. Thus, since by assumption $p^*(\mathbf{c})$ does *not* factorize, either $\nabla_{\mathbf{c}} f^*(\mathbf{c})$ is not σ-diagonal or $p(\mathbf{z})$ does not factorize. Because the latter is false by assumption, it must be that $\nabla_{\mathbf{c}} f^*(\mathbf{c})$ (the Jacobian of f^*) is *not* σ-diagonal.

The source code for our experiments can be found at:

https://github.com/ftraeuble/disentanglement_lib

Joint distributions of correlated factors in datasets. In Figure B.1 we show the joint probability distributions of the correlated pair of FoV for all datasets and correlation strengths considered in this study. Dataset A, B and C were designed with correlated factors of variation that are ordinal for a natural visual interpretation of the traversals. In contrast, datasets D and E contain a correlated factor of variation that has no such natural ordering.

Pairwise entanglement metric. We base the computation of the pairwise entanglement metric on a procedure developed in Locatello et al. [391]: Starting from the GBT feature importance matrix between encoded FoV and latent codes, a bipartite graph between latent codes and FoVs is constructed such that the edge weights are the corresponding matrix elements. After deleting all edges that have weight smaller than a given threshold, one counts the number of disconnected sub graphs having more than one vertex and how many FoV are still connected with at least one latent. When repeating this computation for different thresholds, one can track at which particular threshold a given pair of FoV is merged with each other in this bipartite graph, resulting in the metric we are reporting. A pair of FoV that are being merged at a higher threshold are statistically more related to each other via shared latent code dimensions. This computation can be not only based on the GBT feature importance matrix but likewise on weight matrices inferred from the mutual information.

Unfairness between a pair of FoV. The scores reported are based on a notion of demographic parity for predicting a target variable y given a protected and sensitive variable s. Both y and s can be associated with a factor of variation here. Rather than using the global total variation average as defined in Locatello et al. [70], we report the individual demographic parities for the correlated factors specifically.

Disentanglement metrics. To measure disentanglement of a learned representation, various metrics have been proposed, each requiring access to the ground truth labels. The BetaVAE score is based upon the prediction of a fixed factor from the disentangled representation using a linear classifier [57]. The FactorVAE score is intended to correct for some failures of the former by utilizing majority vote classifiers based on a normalized variance of each latent dimension [59]. The SAP score represents the mean distance between the classification errors of the two latent dimensions that are most predictable [61]. For the MIG score, one computes the mutual information between the latent representation and the ground truth factors and calculates the final score using a normalized gap between the two highest MI entries for each factor. Finally, a disentanglement score proposed by Eastwood and Williams [84], often referred to as DCI score, is calculated from a dimension-wise entropy reflecting the usefulness of the dimension to predict a single factor of variation.

Table B.1: Pairwise entanglement scores help to uncover still-existent correlations in the latent representation. Mean of the pairwise entanglement scores for the correlated pair (red) and the median of the uncorrelated pairs. We see that stronger correlation leads to statistically more entanglement latents across all datasets studied compared to their baseline pairwise entanglement where the data exhibits no correlations (blue). Each score is the mean across 180 models for each dataset and correlation strength. Scores are based on GBT feature importance; scores in brackets are based on the Mutual Information.

corr. strength		$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.7$	$\sigma = \infty$ (uc)
Shapes3D (A)	object size - azimuth	0.38 (0.28)	0.26 (0.25)	0.13 (0.2)	0.08 (0.17)
	median other pairs	0.09 (0.2)	0.09 (0.2)	0.09 (0.19)	0.08 (0.18)
dSprites (B)	orientation - position x	0.17 (0.34)	0.16 (0.31)	0.14 (0.24)	0.11 (0.14)
	median other pairs	0.13 (0.16)	0.13 (0.18)	0.13 (0.19)	0.13 (0.15)
MPI3D (C)	First DOF - Second DOF	0.2 (0.54)	0.19 (0.52)	0.17 (0.5)	0.16 (0.49)
	median other pairs	0.16 (0.25)	0.16 (0.25)	0.15 (0.26)	0.15 (0.25)
Shapes3D (D)	object color - object size	0.29 (0.38)	0.28 (0.31)	-	-
	median uncorrelated pairs	0.07 (0.11)	0.07 (0.11)	-	-
Shapes3D (E)	object color - azimuth	0.25 (0.43)	0.23 (0.3)	-	-
	median uncorrelated pairs	0.1 (0.15)	0.09 (0.15)	-	-

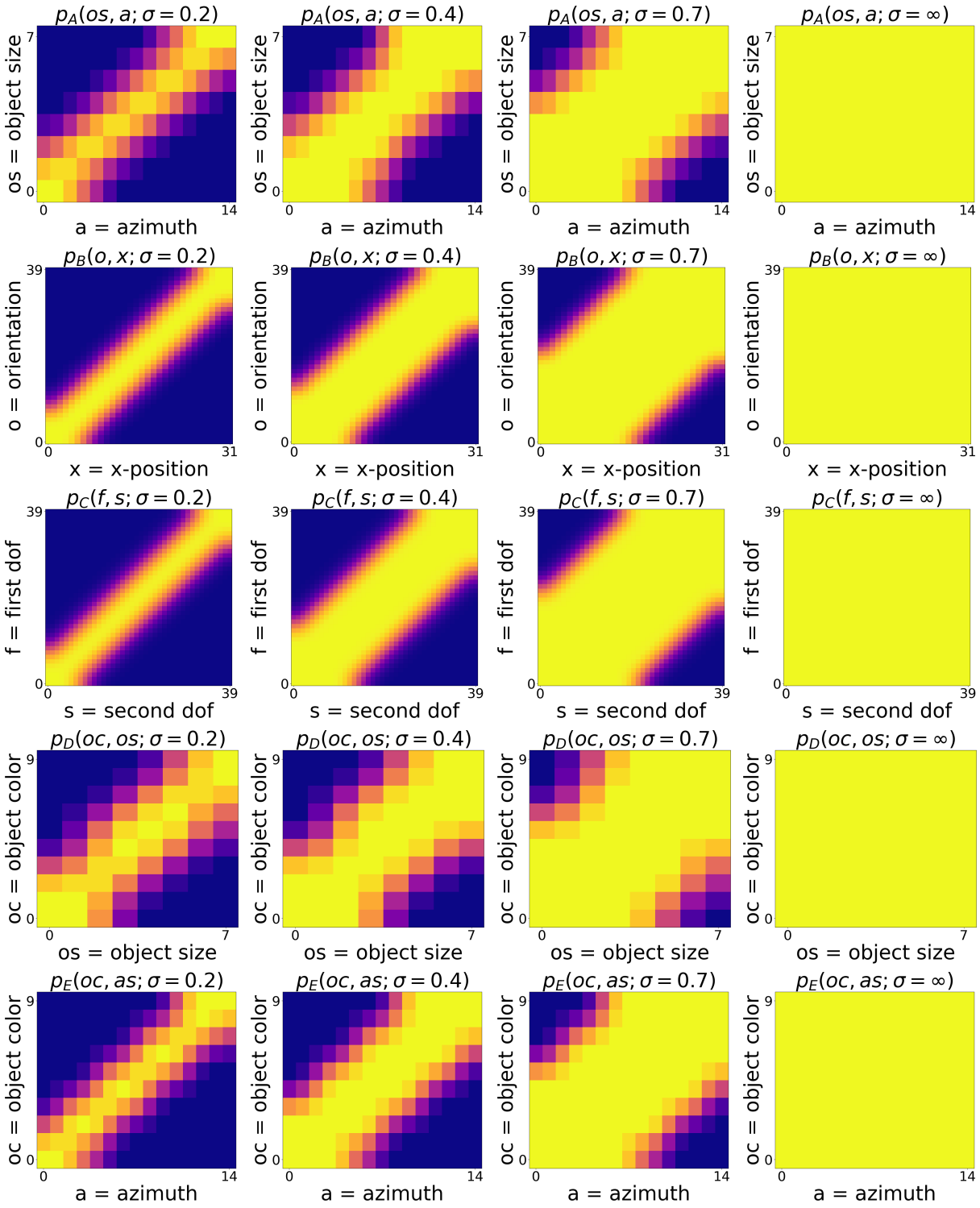


Figure B.1: Probability distributions for sampling training data in the correlated pair of FoVs in the respective datasets (A, B, C, D, E) considering correlation strengths of $\sigma = 0.2$, $\sigma = 0.4$, $\sigma = 0.7$ and $\sigma = \infty$, the uncorrelated limit (from left to right).

B.3 Additional Results - Unsupervised Disentanglement under Correlated Factors

B.3.1 Unsupervised Disentanglement under Correlated Factors

Latent structure and pairwise entanglement. Our hypothesis that the latent representations are less correlated if the correlation strength is weaker is shown for a model on Shapes3D (A) with weak correlation in Figure B.2. Here the latent traversals do not mirror the major and minor axis of the correlated joint distribution.

To make our conclusion more sound we perform an empirical analysis of the pairwise entanglement metrics for the correlated pair vs. the median of all other pairs across the entire unsupervised study on all datasets and models trained. Table B.1 shows the results of this analysis, aggregated across all disentanglement models. To avoid missing any particular disentanglement regularizer that might disentangle the correlated pair but is hidden among the combined aggregation, we also separately report the thresholds for each of the six disentanglement models in Table B.2. We can clearly see that the correlated pair has a much higher entanglement than the rest of the pairs in the trained models across the full board, thus confirming our conclusion that inductive bias of current SOTA unsupervised disentanglement learners is insufficient. Another pairwise metric that tracks the correlation strength in our scenario is the unfairness score between the correlated pair of factors that is being shown for datasets A, B and C in Figure B.3.

Shortcomings of existing metrics. Following recent studies, we evaluate the trained models with the help of a broad range of disentanglement metrics that aim at quantifying overall success by a single scalar measure. Perhaps surprisingly, Figure B.4 and Figure B.5 show no clear trend among all implemented disentanglement scores w.r.t. correlation strength. The metrics have been evaluated by both, either sampling from the correlated data distribution or from the uncorrelated distribution. Given our extensive analysis of latent entanglements of the correlated FoV pair from above, we thus argue that common disentanglement metrics are limited in revealing those when correlations are introduced into the training data and we partly account this to the averaging procedures across many FoV with these pairs. Note that regarding BetaVAE and FactorVAE this observed trend is to some degree expected as they would yield perfect disentanglement scores even if we would take the correlated ground truth factors or a linear transformation in the case of BetaVAE as the representation.

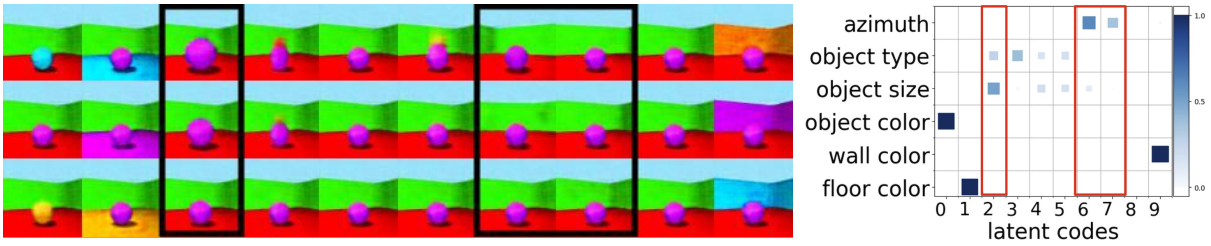


Figure B.2: We show latent traversals (left) of the best DCI score model among all 180 trained models with weak correlation ($\sigma = 0.7$) in object size and azimuth. The traversals in latent codes 2, 6, and 7 (highlighted in black), suggest that these dimensions encode no mixture of azimuth and object size compared to the models with stronger correlation. This is supported by the GBT feature importance matrix of this model (right).

Table B.2: Pairwise entanglement scores from Table B.1 separated along each disentanglement regularizer. Mean of the pairwise entanglement scores for the correlated pair (corr. pair) and the median of the uncorrelated pairs (uncorr. pairs). We see that stronger correlation leads to statistically more entanglement latents across all datasets and regularizers studied compared to their baseline pairwise entanglement where the data exhibits no correlations. Each pairwise score is the mean across 30 models for each dataset and correlation strength. Scores are based on GBT feature importance.

(a) $\beta-$VAE

corr. strength		0.2	0.4	0.7	∞ (uc)
Shapes3D (A)	corr. pair	0.38	0.25	0.14	0.08
	uncorr. pairs	0.08	0.09	0.09	0.07
dSprites (B)	corr. pair	0.18	0.16	0.14	0.12
	uncorr. pairs	0.14	0.14	0.14	0.12
MPI3D (C)	corr. pair	0.22	0.19	0.18	0.16
	uncorr. pairs	0.16	0.15	0.15	0.14
Shapes3D (D)	corr. pair	0.28	0.3	-	-
	uncorr. pairs	0.08	0.07	-	-
Shapes3D (E)	corr. pair	0.24	0.25	-	-
	uncorr. pairs	0.11	0.09	-	-

(b) Factor-VAE

corr. strength		0.2	0.4	0.7	∞ (uc)
Shapes3D (A)	corr. pair	0.48	0.3	0.1	0.03
	uncorr. pairs	0.07	0.06	0.07	0.04
dSprites (B)	corr. pair	0.23	0.2	0.16	0.12
	uncorr. pairs	0.14	0.15	0.14	0.13
MPI3D (C)	corr. pair	0.23	0.22	0.19	0.18
	uncorr. pairs	0.15	0.15	0.14	0.15
Shapes3D (D)	corr. pair	0.36	0.33	-	-
	uncorr. pairs	0.02	0.03	-	-
Shapes3D (E)	corr. pair	0.3	0.28	-	-
	uncorr. pairs	0.1	0.08	-	-

(c) Annealed-VAE

corr. strength		0.2	0.4	0.7	∞ (uc)
Shapes3D (A)	corr. pair	0.32	0.25	0.13	0.11
	uncorr. pairs	0.09	0.09	0.11	0.1
dSprites (B)	corr. pair	0.17	0.16	0.14	0.1
	uncorr. pairs	0.14	0.15	0.14	0.15
MPI3D (C)	corr. pair	0.17	0.17	0.15	0.15
	uncorr. pairs	0.15	0.15	0.15	0.14
Shapes3D (D)	corr. pair	0.33	0.28	-	-
	uncorr. pairs	0.07	0.08	-	-
Shapes3D (E)	corr. pair	0.25	0.19	-	-
	uncorr. pairs	0.1	0.1	-	-

(d) β-TC-VAE

corr. strength		0.2	0.4	0.7	∞ (uc)
Shapes3D (A)	corr. pair	0.41	0.26	0.09	0.05
	uncorr. pairs	0.07	0.09	0.06	0.05
dSprites (B)	corr. pair	0.18	0.15	0.13	0.11
	uncorr. pairs	0.14	0.14	0.13	0.12
MPI3D (C)	corr. pair	0.24	0.22	0.19	0.17
	uncorr. pairs	0.18	0.17	0.15	0.15
Shapes3D (D)	corr. pair	0.3	0.29	-	-
	uncorr. pairs	0.05	0.06	-	
Shapes3D (E)	corr. pair	0.23	0.23	-	-
	uncorr. pairs	0.09	0.07	-	

(e) Dip-VAE-I

corr. strength		0.2	0.4	0.7	∞ (uc)
Shapes3D (A)	corr. pair	0.38	0.24	0.14	0.07
	uncorr. pairs	0.1	0.1	0.11	0.09
dSprites (B)	corr. pair	0.13	0.13	0.12	0.11
	uncorr. pairs	0.11	0.11	0.11	0.11
MPI3D (C)	corr. pair	0.16	0.15	0.14	0.14
	uncorr. pairs	0.13	0.13	0.13	0.13
Shapes3D (D)	corr. pair	0.27	0.25	-	-
	uncorr. pairs	0.06	0.06	-	-
Shapes3D (E)	corr. pair	0.22	0.22	-	-
	uncorr. pairs	0.1	0.1	-	-

(f) Dip-VAE-II

corr. strength		0.2	0.4	0.7	∞ (uc)
Shapes3D (A)	corr. pair	0.28	0.23	0.19	0.14
	uncorr. pairs	0.12	0.11	0.11	0.11
dSprites (B)	corr. pair	0.14	0.14	0.13	0.12
	uncorr. pairs	0.14	0.14	0.13	0.13
MPI3D (C)	corr. pair	0.22	0.21	0.18	0.17
	uncorr. pairs	0.15	0.14	0.15	0.15
Shapes3D (D)	corr. pair	0.23	0.2	-	-
	uncorr. pairs	0.13	0.12	-	-
Shapes3D (E)	corr. pair	0.23	0.19	-	-
	uncorr. pairs	0.12	0.13	-	-

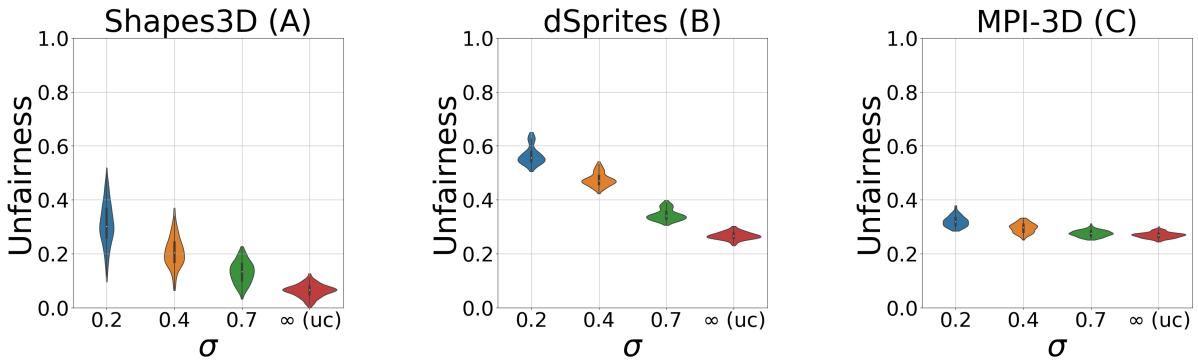


Figure B.3: Disentangled representations trained on correlated data are anti-correlated with higher fairness properties. The plots show the mean unfairness scores between the correlated factors with decreasing correlation strength for Shapes3D (A), dSprites (B) and MPI3D-real (C).

As these metrics are defined with respect to the whole set of underlying ground truth FoV and employ various averaging techniques to form a single scalar measure we want to investigate how much the observed latent entanglements are hidden though imperfect disentanglement of other factors. Thus, we evaluate the same metrics but only on the two correlated FoV excluding all other remaining factors. Indeed, as can be seen from Figure B.6, only DCI tracks the entangled latents under this reduced disentanglement score, while the others show no or only weak trends. We refer the interested reader to Locatello et al. [391], where a detailed discussion is provided why MIG is not tracking the latent entanglement we observed.

B.3.2 Generalization Under Changing Correlations

Generalization Properties In order to support our conclusion that disentanglement methods can generalize towards unseen FoV configurations we show in Figure B.7 latent traversals originating from OOD point number 2 with smallest object size and largest azimuth. We observe that changes in the remaining factors reliably yield the expected reconstructions.

Emphasizing the generalization results from the main paper, we are visualizing the latent spaces with similar extrapolation and generalization capabilities of four additional models from the two strongest correlation dataset variants of Shapes3d (D) and Shapes3d (E) in Figure B.9 and Figure B.8. These latent spaces further support that OOD examples are meaningfully encoded into the existing structure of the latent space and that the decoder is equally capable of generating observations from such unseen representations.

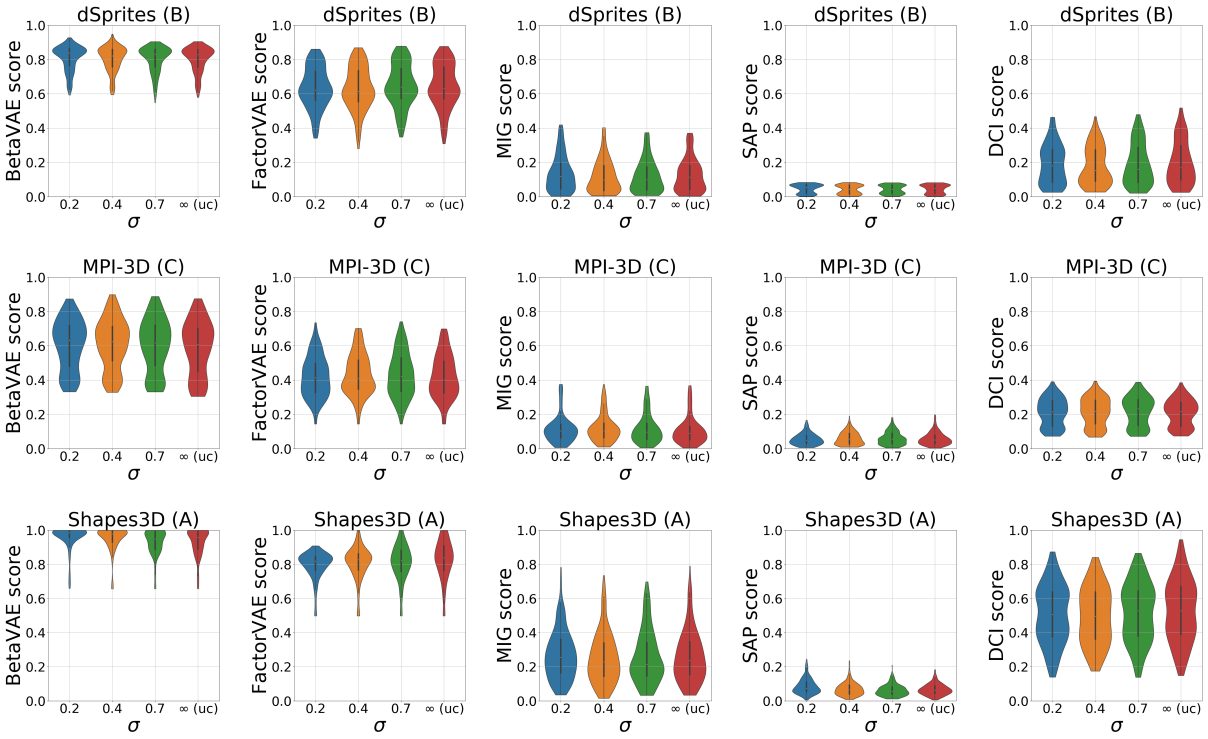


Figure B.4: Standard global disentanglement metrics evaluated on the correlated training set showing no clear trend for different correlation strengths.

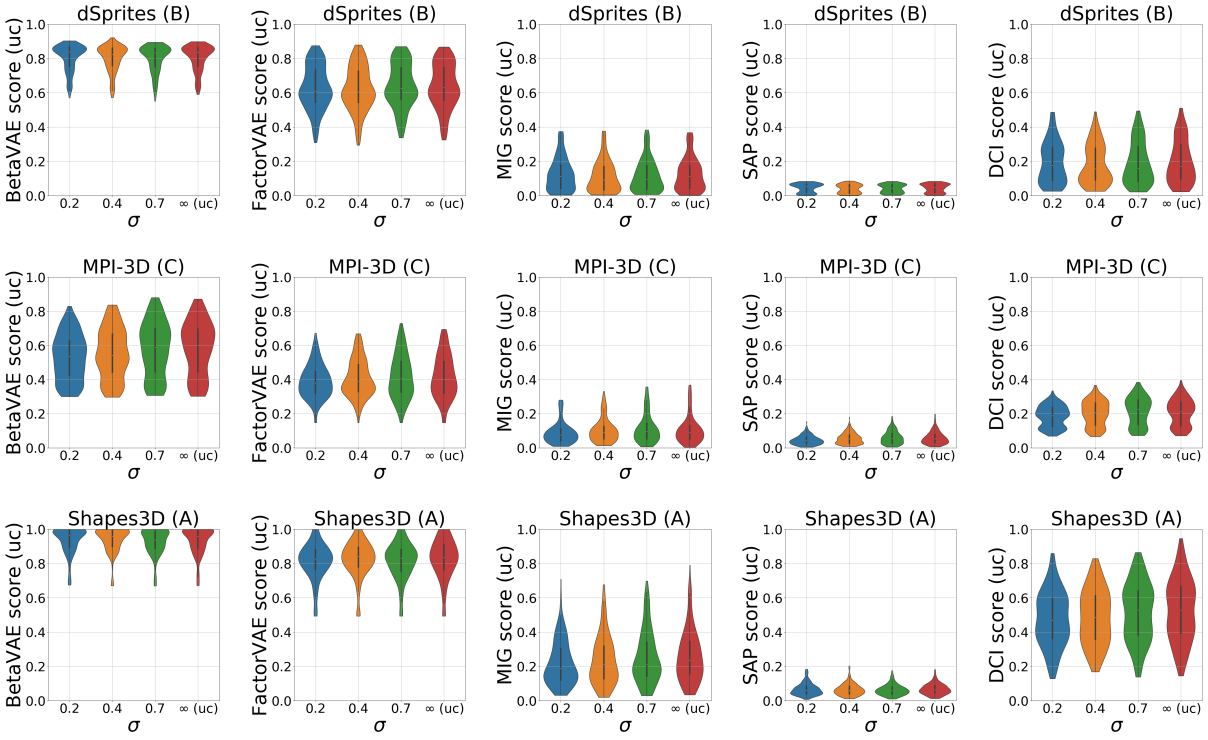


Figure B.5: Standard global disentanglement metrics evaluated on the uncorrelated (uc) dataset set showing no clear trend for different correlation strengths.

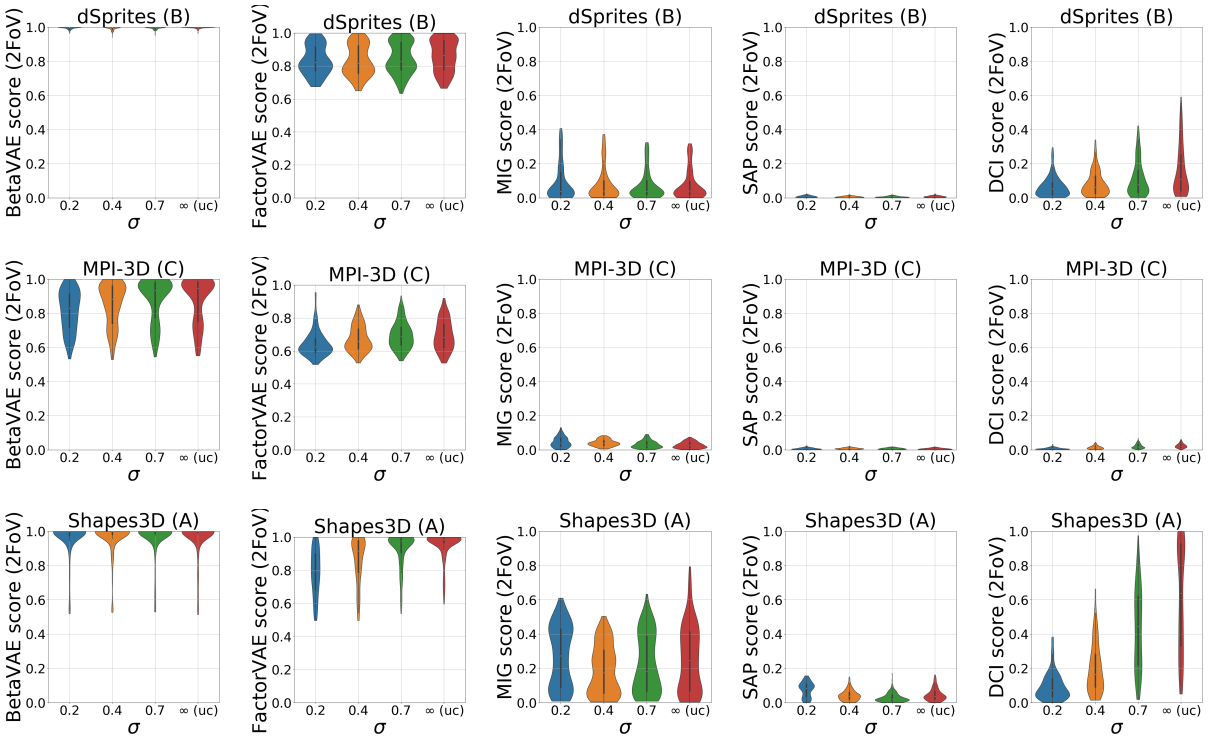


Figure B.6: When disentanglement metrics are only evaluated regarding the 2 correlated FoV we can observe the still persisting entanglement in the latents using DCI score.

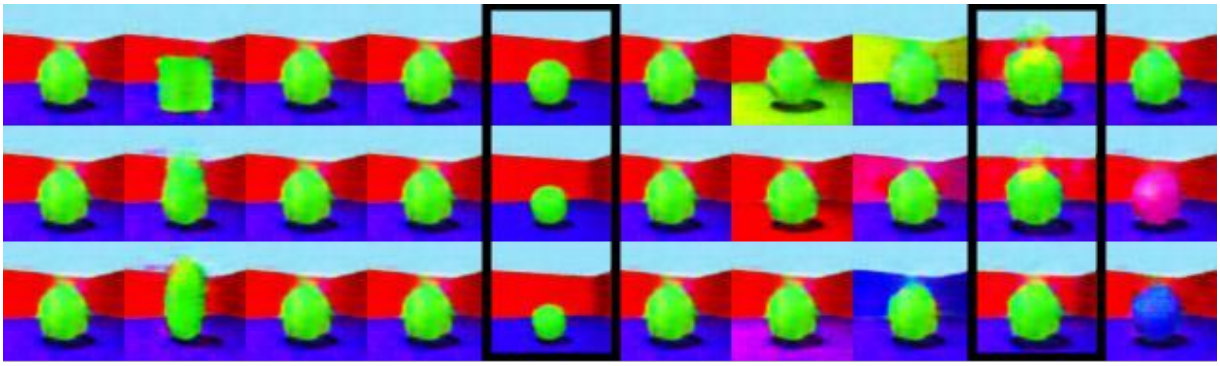


Figure B.7: Generalization capabilities towards out-of-distribution test data. Latent traversals from an observations the model has never seen during training. The starting point corresponds to a factor configuration in point number 2 from Figure 2.4 Shown are the results of the model with highest DCI score among all 180 trained models on Shapes3d (A) with a very restricted correlation strength $\sigma = 0.2$ in object size and azimuth

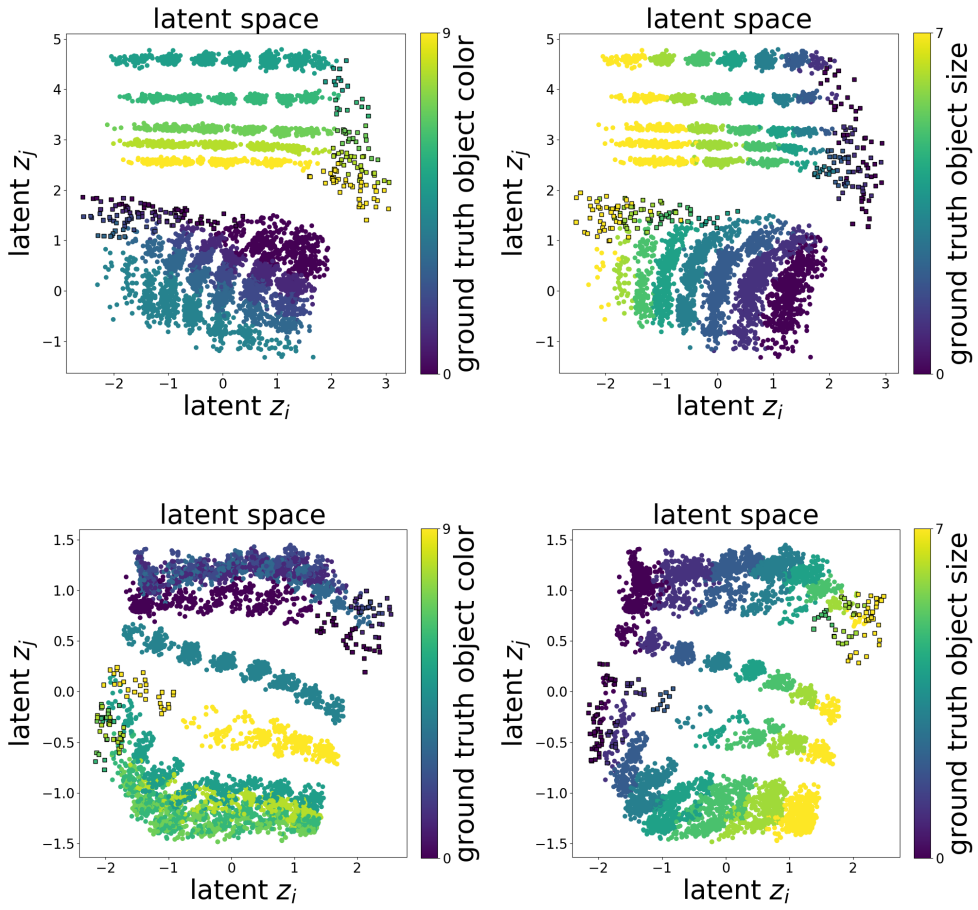


Figure B.8: Latent space distribution of the two entangled dimensions of the best DCI model in Shapes3d (E) with $\sigma = 0.2$ (top row), in Shapes3d (E) with $\sigma = 0.4$ (bottom row). Latent codes sampled from correlated observations (circle without edge) and (2) latent codes sampled with an object color-azimuth configuration not encountered during training (squares with black edge). Each column shows the ground truth values of the two correlated factors by color.

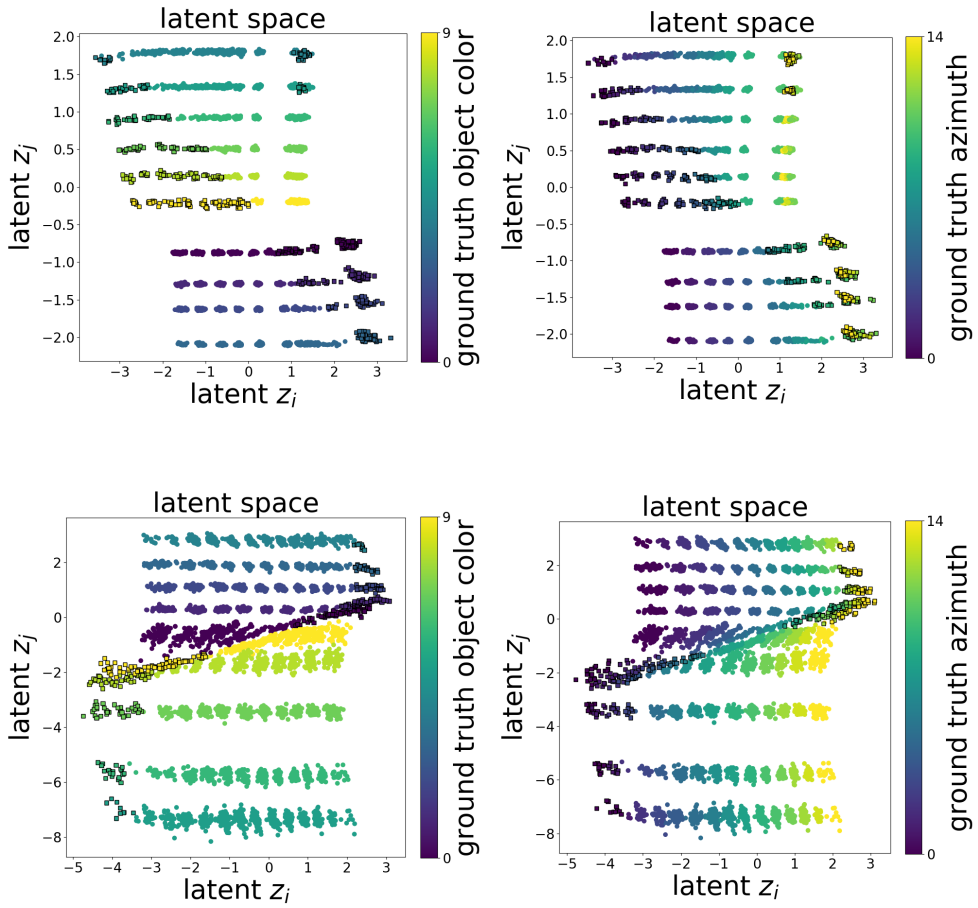


Figure B.9: Latent space distribution of the two entangled dimensions of the best DCI model in Shapes3d (D) with $\sigma = 0.2$ (top row) and in Shapes3d (D) with $\sigma = 0.4$ (bottom row). Latent codes sampled from correlated observations (circle without edge) and (2) latent codes sampled with an object color-size configuration not encountered during training (squares with black edge). Each column shows the ground truth values of the two correlated factors by color.

B.4 Additional Results - Recovering Factorized Latent Code

B.4.1 Post-hoc alignment correction with few labels

In Figure B.10, we see the axis alignment of the correlated latent space after fast adaptation using linear regression on a model trained on Shapes3D (A). Fast adaptation with linear regression substitution fails in some settings: when no two latent dimensions encode the applied correlation isolated from the other latent codes, or when the correlated variables do not have a unique natural ordering (e.g. color or categorical variables). Additionally, the functional form of the latent manifolds beyond the training distribution is unknown and in general expected to be nonlinear. We test the possibility of fast adaptation in this case using as substitution function a one-hidden layer MLP classifier of size 100 on the correlated Shapes3D variants. Under this method, we sample the few labels from a uniform independent distribution. A small number of such samples could practically be labeled manually. Using only 1000 labeled data points for our fast adaptation method shows a significant reduction in entanglement thresholds for the correlated pair (Table B.3).

B.4.2 Alignment during training using weak supervision

Using the studied weakly supervision Ada-GVAE method with $k = 1$ from Locatello et al. [67], we showed that weak supervision can provide a strong inductive bias capable of finding the right factorization and resolving spurious correlations for datasets of unknown degree of correlation. Besides the results shown on Shapes3D (A) in the main paper, representative latent space visualizations that show strong axis alignment across all three correlation variants in Shapes3D (A, D, E) are shown in Figure B.11. This study contains a total of 360 trained models.

In addition to the experiment from the main paper where pairs are constructed solely from the correlated observational data, we want to study two scenarios where we have some intervention capabilities on the FoV to generate training pairs. The resulting distribution of FoVs (still exhibiting correlations) in these pairs depends on whether the correlation between two pairs is due to a causal link or due to a common confounder.

Scenario I-1: We assume there is a confounder (which is not among the observable factors in the data) causing a spurious correlation between the pair of correlated factors. Then, the correlation is broken whenever our interventional sampling procedure yields a pair where the changing FoV is

Table B.3: Mean of the pairwise entanglement scores for the correlated pair (red) and the median of the uncorrelated pairs (based on GBT feature importance) for all pairs of variables in Shapes3D (D) (top), Shapes3D (E) (middle) and Shapes3D (A) (bottom) all with correlation strength $\sigma = 0.2$. Each pairwise score is the mean across 180 models for each dataset and correlation strength. First column is the unsupervised baseline without any fast adaptation and the second column shows that fast adaption using a one-hidden layer MLP reduces these correlations with as little as 1000 labels.

dataset	labels	0	1000
A ($\sigma = 0.2$)	object size - azimuth	0.37	0.26
	median uncorrelated pairs	0.09	0.1
D ($\sigma = 0.2$)	object color - object size	0.3	0.16
	median uncorrelated pairs	0.07	0.07
E ($\sigma = 0.2$)	object color - azimuth	0.25	0.2
	median uncorrelated pairs	0.1	0.11

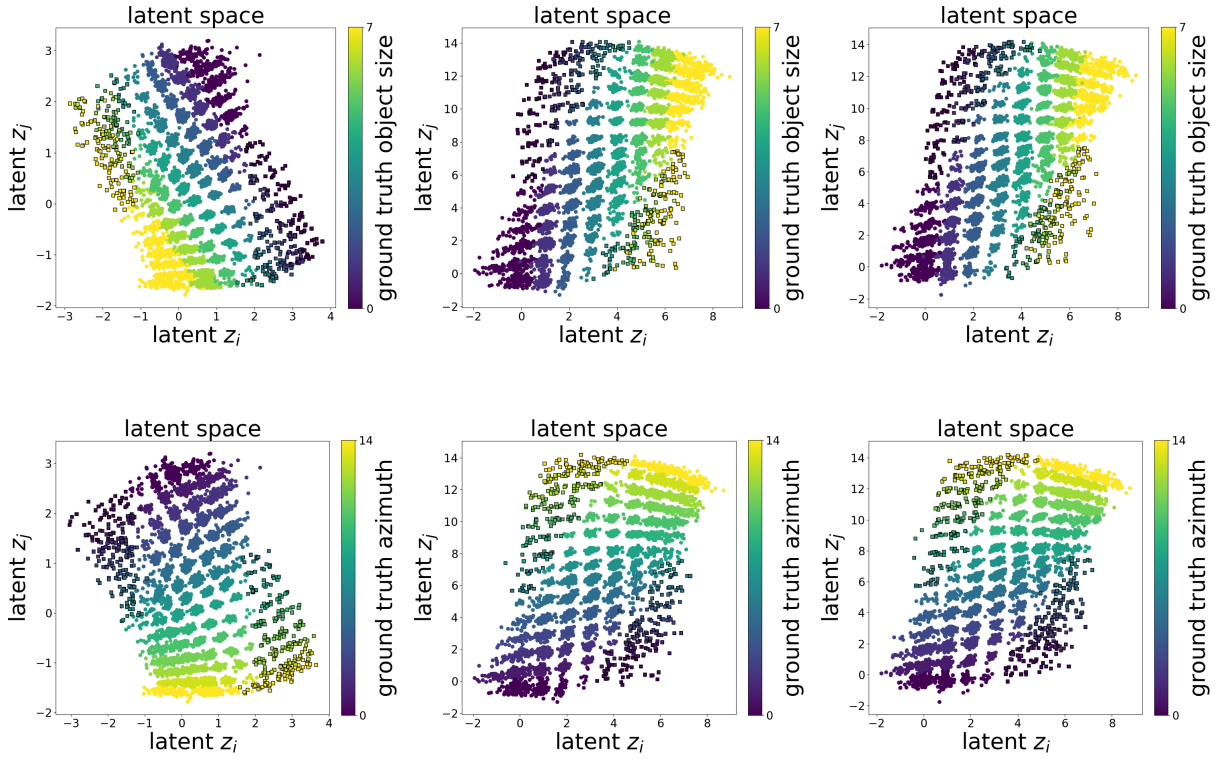


Figure B.10: Latent space distribution of the two entangled dimensions of the best DCI model in Shapes3D (A). Latent codes sampled from correlated observations (circle without edge) and (2) latent codes sampled with an object size-azimuth configuration not encountered during training (squares with black edge). Left column shows the latent space of the two correlated factors by color. Middle and right column show the fast adapted space using linear regression and 100 or 1000 labels respectively.

one of the correlated ones. In that case, the value of the changing variable is sampled uniformly in the second observation of the pair. Note that this still means that the vast majority of sampled pairs exhibit correlated FoV as in most cases the changing factor will be one of the other independent uncorrelated FoV.

As under the default scenario from the main paper, we consistently observe high disentanglement models, often achieving perfect DCI score irrespective of correlations in the data set. This is depicted together with some selected latent space visualizations that show strong axis alignment in Figure B.12. The latent spaces of the correlated FoV in the train data tend to strongly align their coordinates with the ground truth label axis. We chose 10 random seeds per configuration in this study, yielding 720 models in total.

Scenario I-2: Let us assume c_1 causes c_2 in our examples, which manifests as the studied linear correlation. If we intervene on (or "fix") all factors except for the effect c_2, we cannot sample uniformly in c_2 as it is causally affected by c_1. Intervening on all factors but c_1, however, allows us to sample any value in c_1 as it is not causally affected by c_2. To test the hypothesis that this constraint also allows for disentangling the correlation, we trained on Shapes3D (A) and sampled pairs consistent with this causal model. Besides observing visually disentangled factors in the latent traversals, we show a summary of our results in Figure B.13 with the same significant improvements regarding disentangling the correlated FoVs. Besides the correlation strengths used throughout the paper, we additionally trained the same models using a very strong correlation of $\sigma = 0.1$. The study of scenario I-2 thus comprises 300 models.

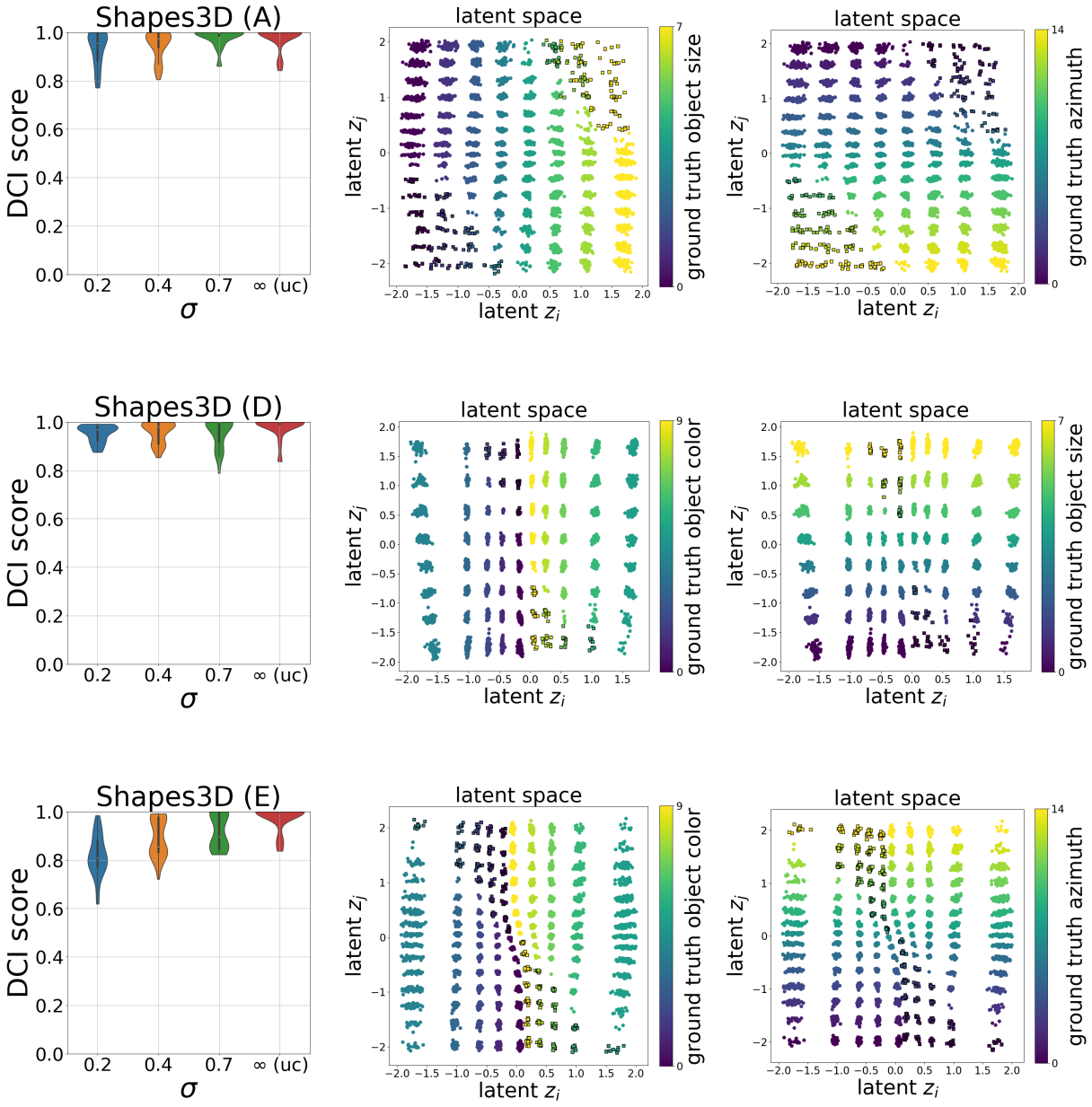


Figure B.11: Left: For the weakly supervised scenario using correlated observational data trained models on Shapes3D (A), (D) and (E) correlating object color and azimuth learn consistently improved, often perfect, disentangled representation across all correlation strengths. Right: Latent dimensions of a best DCI model trained on strongly correlated observational data. Representations are strongly axis-aligned with respect to both of the correlated variables ground truth values (right).

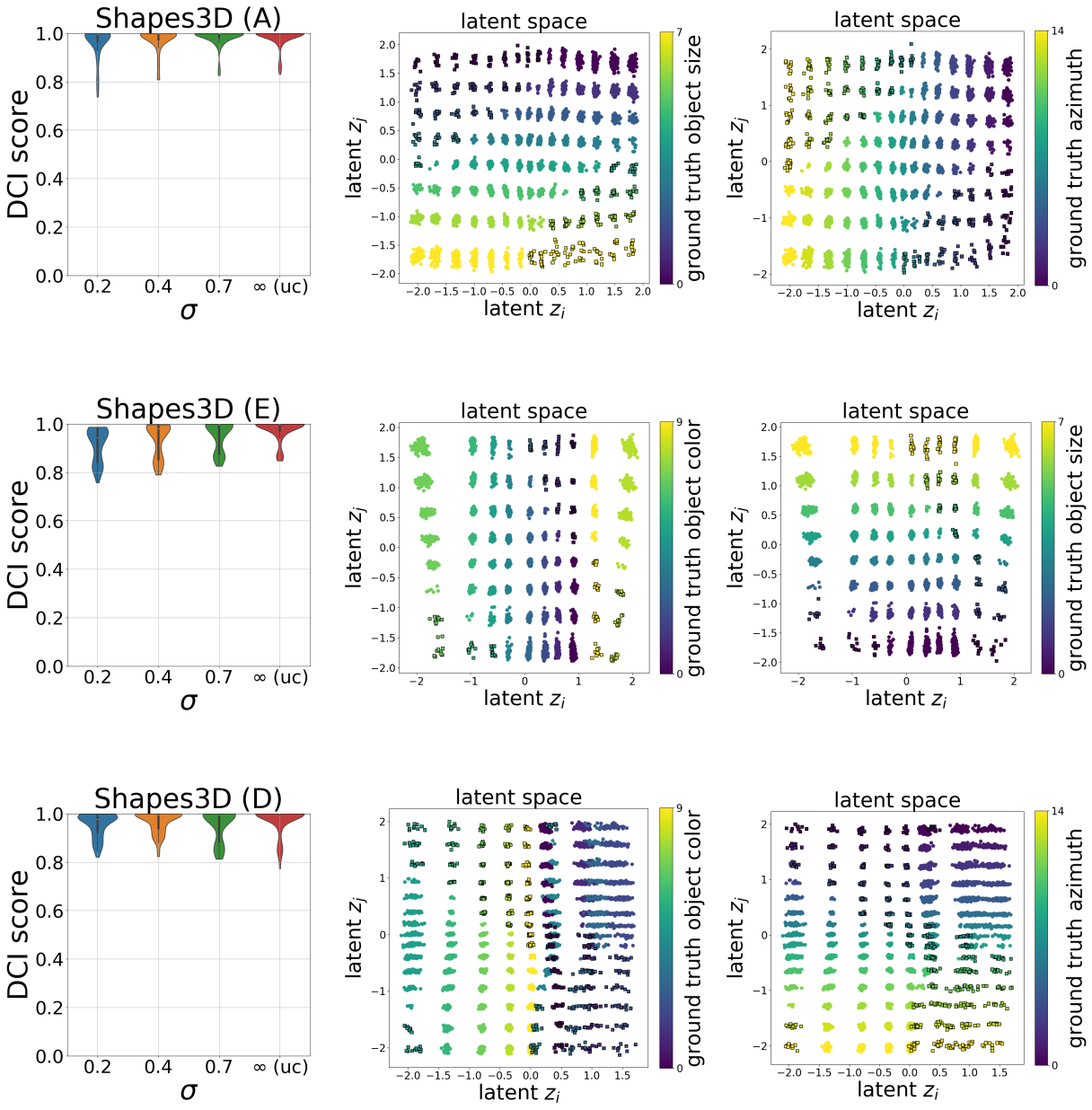


Figure B.12: Left: For the weakly supervised scenario with intervening capabilities (Scenario I-1) trained models on Shapes3D (A), (D) and (E) correlating object color and azimuth learn consistently improved, often perfect, disentangled representation across all correlation strengths. Right: Latent dimensions of a best DCI model with strong correlation ($\sigma = 0.2$). Representations are strongly axis-aligned with respect to both of the correlated variables ground truth values.

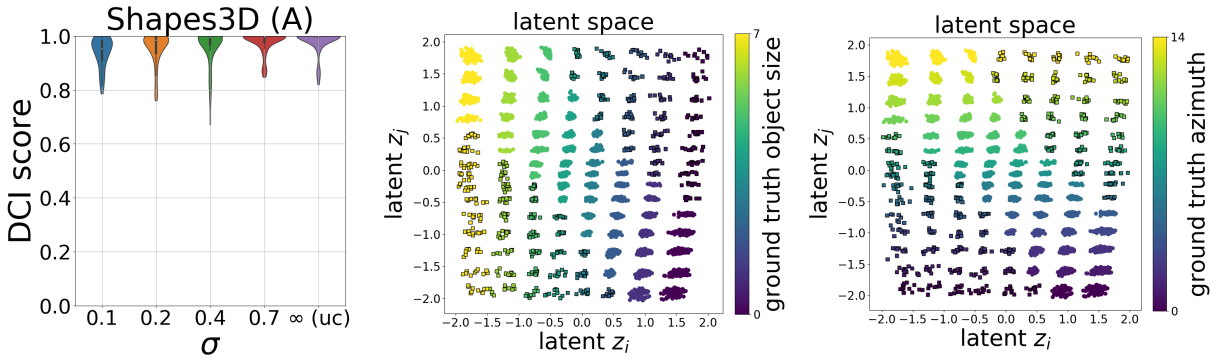


Figure B.13: DCI scores and latent spaces show strong disentanglement using weak supervision with intervening capabilities (Scenario I-2) - even under the stronger assumption that sampling of observation pairs follow its causal generative model. We show the learned latent space encoding of the two correlated factors of variation for a model on Shapes3D with $\sigma = 0.1$.

Appendix to Chapter 3 C

C.1 Observations

`CausalWorld` offers two observation modes, "structured" and "pixel". When using "pixel" mode, 6 images are returned consisting of the current images rendered from 3 different views on top of the TriFinger platform, showing the current state of the environment, as well as the 3 equivalent goal images rendered from the same points of view, showing the goal shape that the robot have to build by the end of the episode (see Figure C.1). The structured mode follows a concatenation of features that is detailed in Figure C.2.

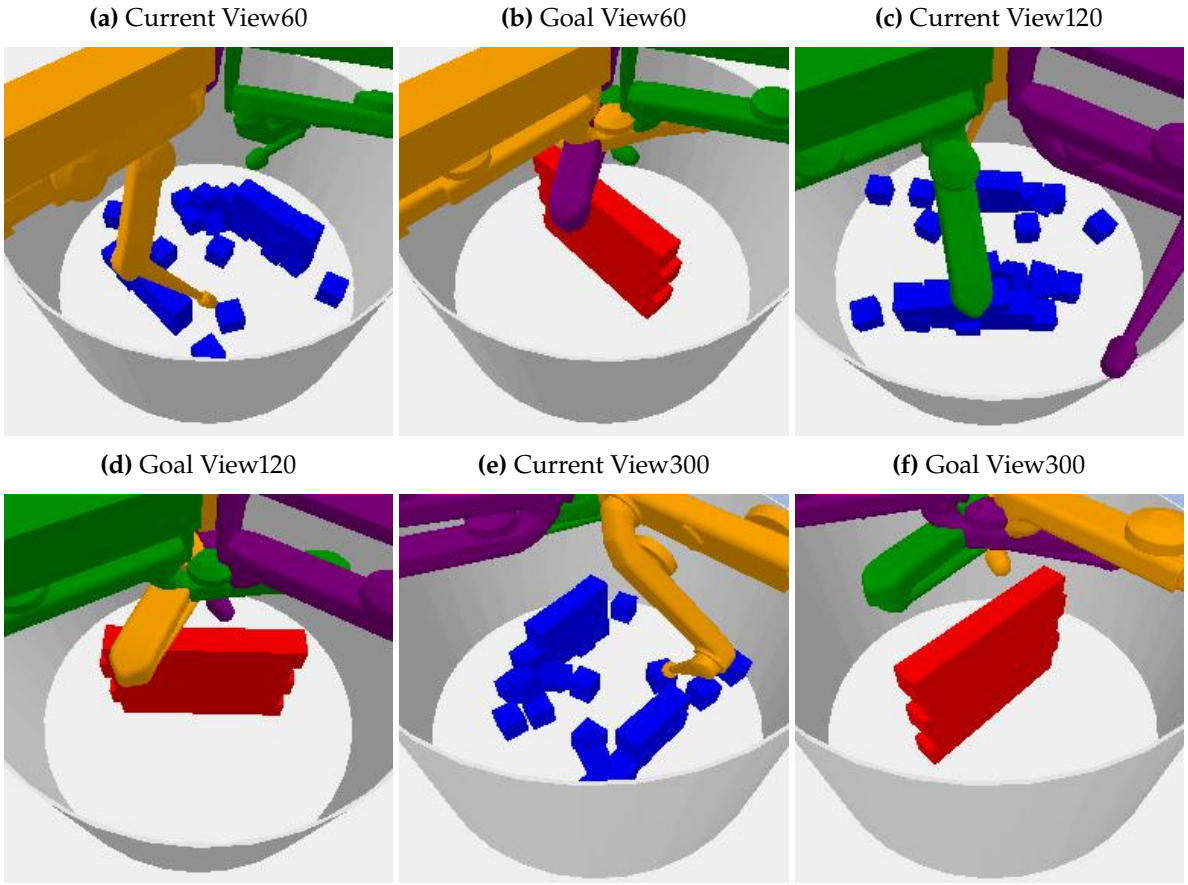


Figure C.1: Example "pixel" mode observations returned at each step of the environment.

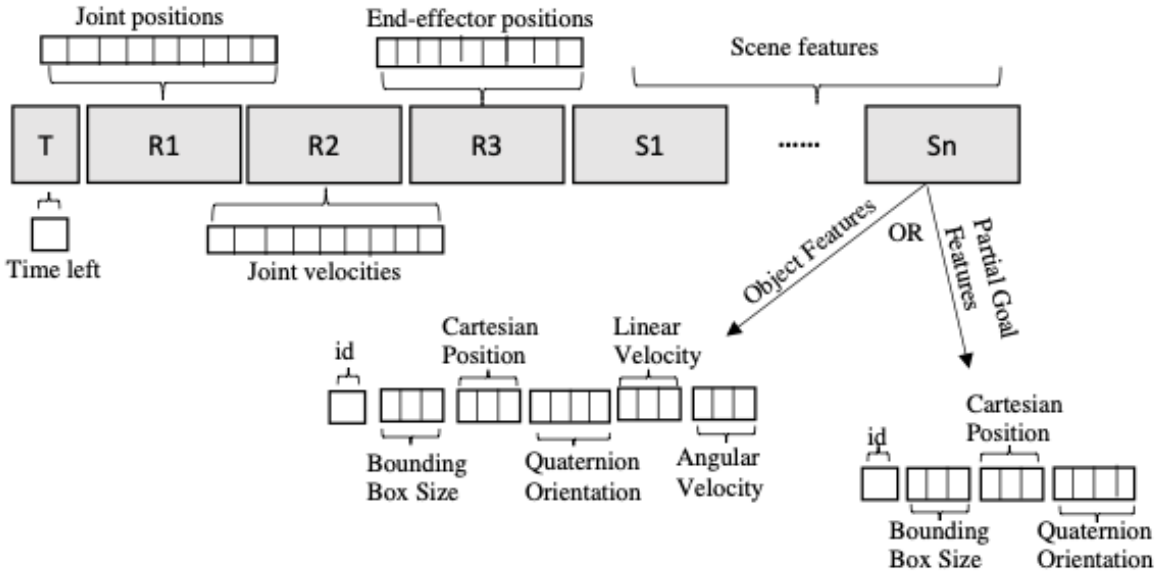


Figure C.2: Structured observation description. For the scene features, all the blocks feature vector are concatenated first. Following that the partial goals feature vector are concatenated in the same order. Lastly, if there is any obstacles/ fixed blocks, their feature vectors are concatenated at the end following the same description as the partial goal features.

C.2 TriFinger Platform

The robot from [128] shown in Figure C.3 is open-sourced and can be reproduced and built in any research lab; since its inexpensive (about $5000) with the aim of speeding up sim-to-real research.

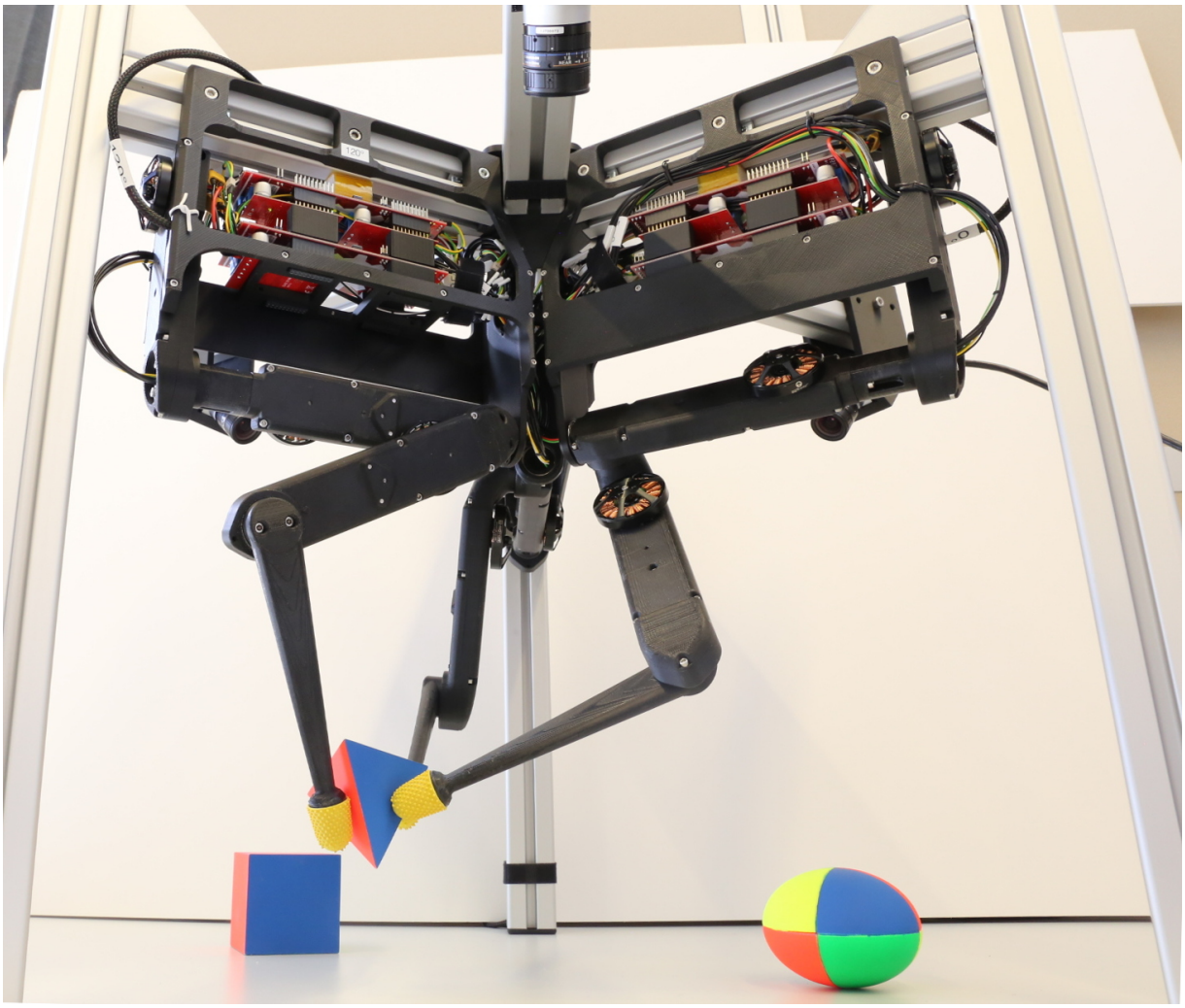


Figure C.3: The TriFinger platform.

C.3 Task Generators

1. *Pushing*: task where the goal is to push one block towards a goal position with a specific orientation; restricted to goals on the floor level.
2. *Picking*: task where the goal is to pick one block towards a goal height above the center of the arena; restricted to goals above the floor level.
3. *Pick And Place*: task where the arena is divided by a fixed long block and the goal is to pick one block from one side of the arena to a goal position with a variable orientation on the other side of the fixed block.
4. *Stacking2*: task where the goal is to stack two blocks above each other in a specific goal position and orientation.

5. *Towers*: task where the goal is to stack multiple n blocks above each other in a specific goal position and orientation - exactly above each other creating a tower of blocks.
6. *Stacked Blocks*: task where the goal is to stack multiple n blocks above each other in an arbitrary way to create a stable structure. The blocks don't have to be exactly above each other; making it more challenging than the ordinary towers task since the its harder to come up with a stable structure that covers the goal shape volume.
7. *Creative Stacked Blocks*: exactly the same as the *Stacked Blocks* task except that the first and last levels of the goal are the only levels shown or "imposed" and the rest of the structure is not explicitly specified, leaving the rest of the goal shape to the imagination of the agent itself; this is considered the most challenging since its it needs the agent to understand how to build stable structures and imagine what can be filled in the middle to connect the two levels in a stable way.
8. *General*: the goal shape is an arbitrary shape created by initially dropping an arbitrary number of blocks from above the ground and waiting till all blocks come to a rest position where this becomes the goal shape that the agent needs to fill up afterwards.

Table C.1: Description of a subset of the high level variables, exposed in CausalWorld, and their corresponding spaces, h refers to the height of the block.

Variable	Sub Variable	Space **A**	Space **B**
gravity[z]	-	$[-10, -7]$	$[-7, -4]$
floor friction	-	$[0.3, 0.6]$	$[0.6, 0.8]$
stage friction	-	$[0.3, 0.6]$	$[0.6, 0.8]$
stage color [rgb]	-	$[0.0, 0.5]^3$	$[0.5, 1]^3$
floor color [rgb]	-	$[0.0, 0.5]^3$	$[0.5, 1]^3$
joint positions	-	$[[-1.57, -1.2, -3.0]^3, [-0.69, 0, 0]^3]$	$[[-0.69, 0, 0]^3, [1.0, 1.57, 3.0]^3]$
block	size	$[0.055, 0.075]^3$	$[0.075, 0.095]^3$
block	color	$[0.0, 0.5]^3$	$[0.5, 1]^3$
block	mass	$[0.015, 0.045]$	$[0.045, 0.1]$
block	position (cyl.)	$[[0, -\pi, h/2], [0.11, \pi, 0.15]]$	$[[0.11, -\pi, h/2], [0.15, \pi, 0.3]]$
goal cuboid	size	$[0.055, 0.075]^3$	$[0.075, 0.095]^3$
goal cuboid	color	$[0.0, 0.5]^3$	$[0.5, 1]^3$
link	color	$[0.0, 0.5]^3$	$[0.5, 1]^3$
link	mass	$[0.015, 0.045]$	$[0.045, 0.1]$

Table C.2: Example of task generators' specific high level variables, exposed in CausalWorld, and their corresponding spaces.

Task Generator	Variable	Space **A**	Space **B**
Picking	goal height	$[0.08, 0.20]$	$[0.20, 0.25]$
Towers	tower dims	$[[0.08, 0.08, 0.08], [0.12, 0.12, 0.12]]$	$[[0.12, 0.12, 0.12], [0.20, 0.20, 0.20]]$

Table C.3: Description of the dense rewards applied in our experiments. The following notation was applied: $v^t \in \mathbf{R}^3$ joint velocities, $e_i^t \in \mathbf{R}^3$ i-th end-effector positions, $o_i^t \in \mathbf{R}^3$ i-th block position, $g_i^t \in \mathbf{R}^3$ i-th goal block position, $d^t(o, e) = \sum_i ||e_i^t - o^t||$ the distance between end-effectors and the block, $\Delta_{o,e}^t = d^t(o, e) - d^{t-1}(o, e)$ the distance difference w.r.t. the previous timestep. The target height parameter t for pick and place is 0.15 if block and goal are of different height. Otherwise, t is half the goal height.

Task generators	Dense reward
Pushing	$-750\Delta^t(o_1, e) - 250\Delta^t(o_1, g_1)$
Picking	$-750\Delta^t(o_1, e) - 250\Delta^t(o_{1,z}, g_{1,z}) - 125\Delta^t(o_{1,x,y}, g_{1,x,y}) - 0.005\|\|v^t - v^{t-1}\|\|$
Pick and Place	$-750\Delta^t(o_1, e) - 50\Delta^t(o_{1,x,y}, g_{1,x,y}) - 250(\|o_{1,z}^t - t\| - \|o_{1,z}^{t-1} - t\|) - 0.005\|\|v^t - v^{t-1}\|\|$
Stacking	$\mathbf{1}_{d^t(o_1,e)>0.02}(-750\Delta^t(o_1, e) - 250\Delta^t(o_1, g_1)) + \mathbf{1}_{d^t(o_1,e)<0.02}(-750\Delta^t(o_2, e) - 250(\|o_{2,z}^t - g_{2,z}^t\| - \|o_{1,z}^{t-1} - g_{2,z}^t\|) - \mathbf{1}_{o_{2,z}^t - g_{2,z}^t > 0} 125\Delta^t(o_{2,x,y}, g_{2,x,y})) - 0.005\|\|v^t - v^{t-1}\|\|$

C.4 Training Details

The experiments were carried out using the `stable baselines` implementation of PPO, SAC and TD3 (see Table C.4). We used a 2 layer MLP Policy [256,256] for all the policies. PPO was trained on 20 workers up to 100 million timesteps in parallel and SAC as well as TD3 were trained serially for 10 million timesteps.

Table C.4: PPO, SAC and TD3 Hyper-Parameters used in the discussed experiments.

Parameter	PPO	SAC	TD3
discount	0.99	0.95	0.96
batch size	120000	256	128
learning rate	2.5e-4	1e-4	1e-4
entropy coef.	0.01	1e-3	-
value function coef.	0.5	-	-
gradient clipping (max)	0.5	-	-
n minibatches per update	40	-	-
n training epochs	4	-	-
target entropy	-	auto	-
buffer size	-	1000000	500000
tau	-	0.001	0.02

C.5 Additional Results

Another example of evaluating generalization across the various axes in `CausalWorld` using the previously discussed protocols on the Pushing task is illustrated in Figure C.4. Evaluation scores along multiple protocols regarding the tasks Pushing, Picking, Pick and Place and Stacking2 are shown in Figure C.5.

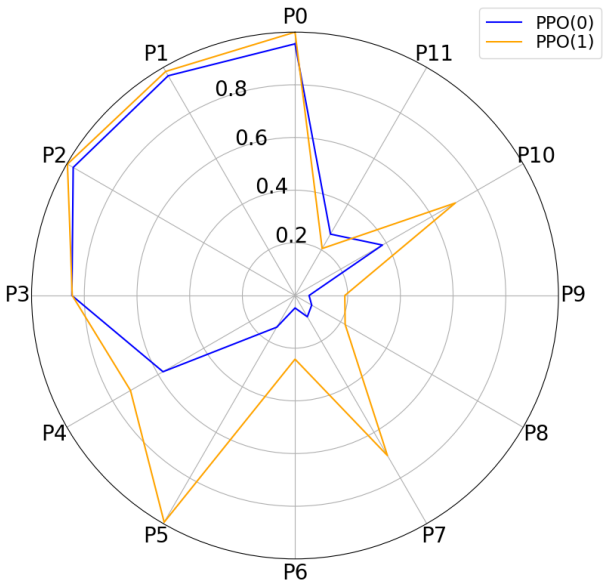


Figure C.4: An example of evaluating generalization across the various axes in `CausalWorld` across the various axes using the previously discussed protocols. Here we compare the mean last fractional success of two agents trained on the Pushing task under different curricula using PPO.

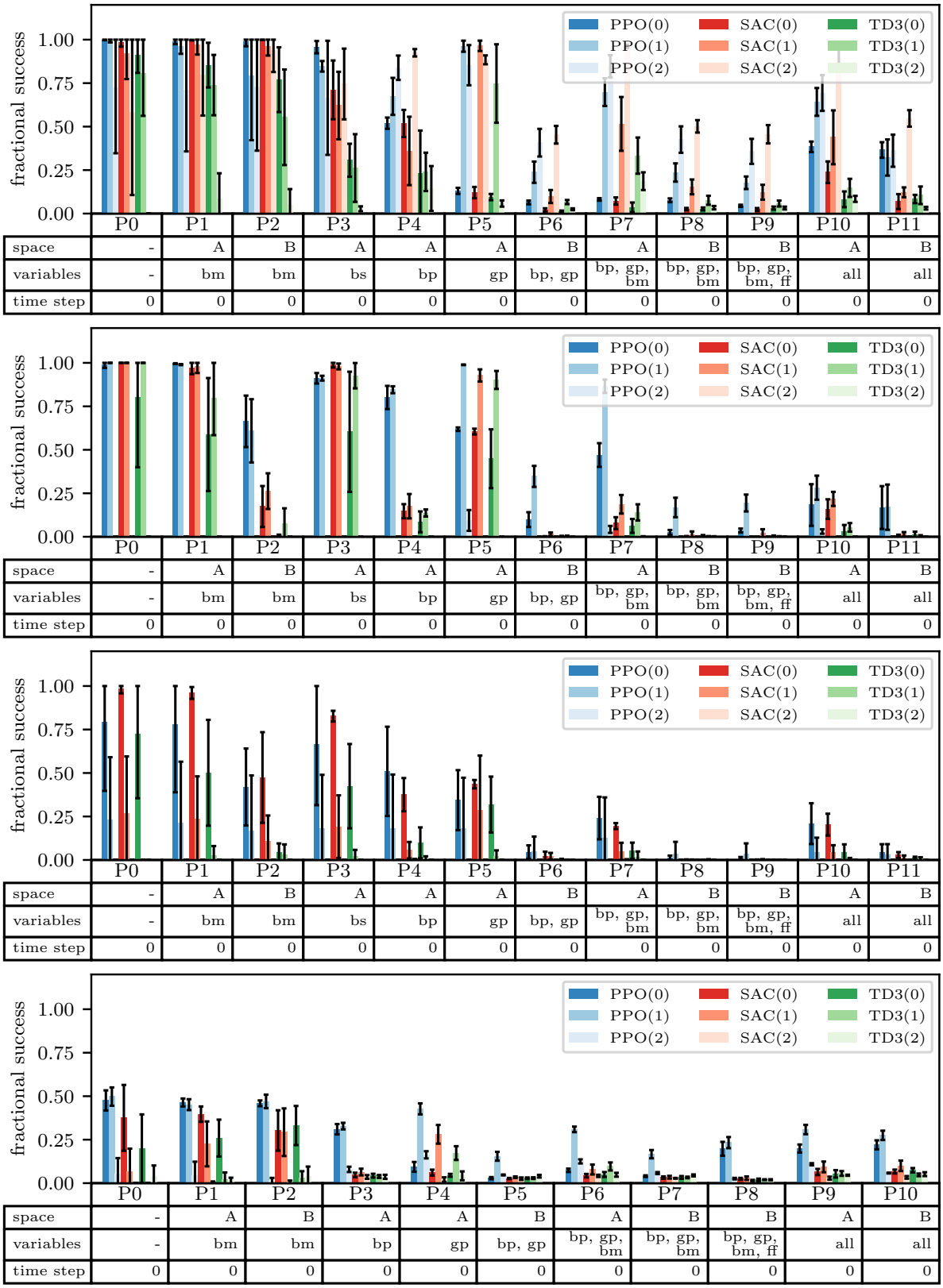


Figure C.5: Evaluation scores, for pushing, picking, pick and place and stacking2 baselines, from top to bottom respectively. Each protocol was evaluated for 200 episodes and each bar is averaged over five models with different random seeds [bp block pose, bm block mass, bs block size, gp goal pose, ff floor friction].

C.6 Causality in Reinforcement Learning

Definition C.6.1 *A partially observable Markov decision processes (POMDP) is defined by the tuple* $(S, A, T, R, \Omega, O, \gamma, \rho_0, H)$ *with states* $s \in S$*, actions* $a \in A$ *and observations* $o \in \Omega$ *determined by the state and action of the environment* $O(o|s, a)$*.* $T(s_{t+1}|s_t, a_t)$ *is the transition probability distribution function,* $R(s_t, a_t)$ *is the reward function,* γ *is the discount factor,* $\rho_0(s)$ *is the initial state distribution at the beginning of each episode, and* H *is the time horizon per episode. The objective of RL algorithms is to learn a policy* $\pi(a_t|h_t)$ *with history* $H_t = (O_1, A_1, ..., A_{t-1}, O_t)$ *that maximizes the discounted expected reward* $J(\pi) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{H} \gamma^t r_t\right]$*.*

Definition C.6.2 *A structural causal model (SCM)* M *over* $X = (X_1, ..., X_N)$ *is given by a DAG* G *over nodes* X*, independent noise RVs* $U = (U_1, ..., U_N)$ *with distributions* P_{U_i} *and functions* $f_1, ..., f_N$ *such that* $X_i = f_i(pa_i, U_i)$*, where* $pa_i \subset X$ *are the parents of* X_i *in* G*. An SCM entails a distribution* P *with density* p *over* (X, U)*.*

Definition C.6.3 *An intervention* I *in an SCM* M *consists of replacing the RHS* $f_i(pa_i, U_i)$ *by* $f_i^I(pa_i^I, U_i)$ *with* $i \subseteq \{1, ..., N\}$ *where* pa_i^I *are the parents in a new DAG* G^I*. The resulting SCM is denoted with* $M^{do(I)}$ *with distribution* $P^{do(I)}$ *and density* $p^{do(I)}$*.*

Representing POMDPs as SCMs: A POMDP can be presented as an SCM M by unrolling the POMDP through time by expressing all conditional distributions, e.g the transition kernel $P_{S_{t+1}|S_t,A_t}$, as deterministic functions (which reflects the causal mechanisms of the system) with independent noise variables U, such as $S_{t+1} = f_{st}(S_t, A_t, U_{st})$. This is always possible using auto-regressive formalization as shown in [140]. The distribution P^{π} over trajectories T determined by the SCM, which means running a different policy μ (instead of π) in the environment is an intervention by itself $I(\pi \to \mu)$ resulting in model distribution $P^{do(I(\pi \to \mu))}$. Given the SCM, one can reason about the alternative outcomes of different actions or even different environment properties (such as friction) (known as counterfactual inference) which is not possible if we only model the system as a conditional distribution $P_{O|A,U_e}$.

Generalizing agents: An agent should generalize well to changes in the environment if it achieves an equal performance after an intervention I is performed on the SCM M resulting in $M^{do(I)}$. This can be accomplished for instance as explained in [46], by reusing the learned causal mechanisms. For example, if an intervention was performed on the colour of one of the available blocks in the environment, it won't change the underlying causal mechanisms. Thus, this sort of interventions on a non causal variable shouldn't affect the resulting actions of the agent. Nevertheless, this is often not the case with RL agents trained using visual observations. [392] showed recently that algorithms which were designed specifically for Meta-RL can display strong overfitting when they are evaluated on challenging visual tasks. On the other hand, if an intervention was performed on a causal variable with respect to a policy π, such as the mass of an object, the resulting action sequence will be expected to change for robust control. Additionally, if an intervention is performed on some of the goal variables, the reward function might also change and a robust controller should also be able to react to it accordingly.

Appendix to Chapter 4 D

D.1 Implementation Details

Training. We train the β-VAEs by maximizing the following objective function:

$$\mathcal{L}_{VAE}^{\beta} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \leq \log p(\boldsymbol{x})$$

with $\beta > 0$ using the Adam optimizer [393] with default parameters. We use a batch size of 64 and train for 400k steps. The learning rate is initialized to 1e-4 and halved at 150k and 300k training steps. We clip the global gradient norm to 1.0 before each weight update. Following Locatello et al. [65], we use a Gaussian encoder with an isotropic Gaussian prior for the latent variable, and a Bernoulli decoder. Our implementation of weakly supervised learning is based on Ada-GVAE [67], but uses a symmetrized KL divergence:

$$\tilde{D}_{\mathrm{KL}}(p, q) = \frac{1}{2} D_{\mathrm{KL}}(p\|q) + \frac{1}{2} D_{\mathrm{KL}}(q\|p)$$

to infer which latent dimensions should be aggregated.

The noise added to the encoder's input consists of two independent components, both i.i.d. Gaussian with zero mean: one is independent for each subpixel (RGB) and has standard deviation 0.03, the other is a 8×8 pixel-wise (greyscale) noise with standard deviation 0.15, bilinearly upsampled by a factor of 16. The latter has been designed (by visual inspection) to roughly mimic observation noise in the real images due to complex lighting conditions.

Neural architecture. Architectural details are provided in Tables D.1 and D.2, and Figure D.1 provides a high-level overview. In preliminary experiments, we observed that batch normalization, layer normalization, and dropout did not significantly affect performance in terms of ELBO, model samples, and disentanglement scores, both in the unsupervised and weakly supervised settings. On the other hand, layer normalization before the posterior parameterization (last layer of the encoder) appeared to be beneficial for stability in early training. While using an architecture based on residual blocks leads to fast convergence, in practice we observed that it may be challenging to keep the gradients in check at the beginning of training.* In order to solve this issue, we resorted to a simple scalar gating mechanism in the residual blocks [176] such that each residual block is initialized to the identity.

Datasets and OOD evaluation. Because we evaluate OOD generalization in terms of cube color hue (except in the sim-to-real case), we first sampled 8 color hues at random from the 12 specified in Figure 4.1. The chosen hues are: $[0°, 120°, 150°, 180°, 210°, 270°, 300°, 330°]$. Then, the dataset D used for training VAEs is generated by randomly sampling values for the factors of variation from Figure 4.1, with the color hue restricted to the above-mentioned values. This makes OOD2

* This instability may also be exacerbated in probabilistic models by the sampling step in latent space, where a large log variance causes the decoder input to take very large values. Intuitively, this might be a reason why layer normalization before latent space appears to be beneficial for training stability.

Table D.1: Encoder (left) and decoder (right) architectures. The latent space dimensionality is denoted by d, and $K = 3$ indicates the number of image channels. Last line in the encoder architecture: the fully connected layer parameterizing the log variance of the approximate posterior distributions of the latent variables has custom initialization. The weights are initialized with 1/10 standard deviation than the default value, and the biases are initialized to -1 instead of 0. Empirically, this together with (learnable) LayerNorm was beneficial for training stability at the beginning of training.

Encoder	
Operation	**Output Shape**
Input	128×128×K
Conv 5x5, stride 2, 64 ch.	64×64×64
LeakyReLU(0.02)	—
2x ResidualBlock(64)	—
Conv 1x1, 128 channels	64×64×128
AveragePool(2)	32×32×128
2x ResidualBlock(128)	—
AveragePool(2)	16×16×128
2x ResidualBlock(128)	—
Conv 1x1, 256 channels	16×16×256
AveragePool(2)	8×8×256
2x ResidualBlock(256)	—
AveragePool(2)	4×4×256
2x ResidualBlock(256)	—
Flatten	4096
LeakyReLU(0.02)	—
FC(512)	512
LeakyReLU(0.02)	—
LayerNorm	—
2x FC(d)	2d

Decoder	
Operation	**Output Shape**
Input	d
FC(512)	512
LeakyReLU(0.02)	—
FC(4096)	4096
Reshape	4×4×256
2x ResidualBlock(256)	—
BilinearInterpolation(2)	8×8×256
2x ResidualBlock(256)	—
Conv 1x1, 128 channels	8×8×128
BilinearInterpolation(2)	16×16×128
2x ResidualBlock(128)	—
BilinearInterpolation(2)	32×32×128
2x ResidualBlock(128)	—
Conv 1x1, 64 channels	32×32×64
BilinearInterpolation(2)	64×64×64
2x ResidualBlock(64)	—
BilinearInterpolation(2)	128×128×64
LeakyReLU(0.02)	—
Conv 5x5, K channels	128×128×K

evaluation possible, specifically OOD2-A where the learned predictors are tested on representations extracted from images with held-out values of the cube hue.

For evaluation of out-of-distribution generalization, we train the downstream predictors on a subset $D_1 \subset D$ of the representation training set. The downstream training set D_1 is sampled at random from D but only contains a (not necessarily proper) subset of the 8 cube colors. This subset contains 1 color in the OOD1-A case, 4 colors in OOD1-B and OOD1-C, all 8 colors in OOD2 (in this case D_1 is simply a random subset of D). Then we test the downstream predictors on a set D_2 distributionally different from D_1 in terms of cube color (all OOD1 scenarios as well as OOD2-A) or sim-to-real (OOD2-B). In the OOD1 case, D_2 is also a subset of D and is generated the same way. In each OOD1 case, the test set D_2 is paired with its corresponding D_1 that was used to train the downstream predictors. D_2 contains all colors in D minus those in D_1. In the OOD2-A case, D_2 is a separate dataset containing 5k simulated images like those in D, except that these only contain the 4 colors that were left out from the VAE training set D (hue in $[30°, 60°, 90°, 240°]$). In the OOD2-B case, the set D_2 is the dataset of real images. Following previous work (e.g. the GBT10000 metric in Locatello et al. [65]), the training set D_1 and test set D_2 for downstream tasks contain 10k and 5k images, respectively, except in the OOD2-B case, where the size is limited by the size of the real dataset.

The datasets can be downloaded: http://people.tuebingen.mpg.de/ei-datasets/iclr_transfer_-paper/robot_finger_datasets.tar

Table D.2: Architecture of one residual block. The scalar gate is implemented by multiplying the tensor by a learnable scalar parameter before adding it to the block input. Initializing the residual block to the identity by setting this parameter to zero has been originally proposed by Bachlechner et al. [176]. The tensor shape is constant throughout the residual block.

Residual Block
Input: shape $H \times W \times C$
LeakyReLU(0.02)
Conv 3x3, C channels
LeakyReLU(0.02)
Conv 3x3, C channels
Scalar gate
Sum with input

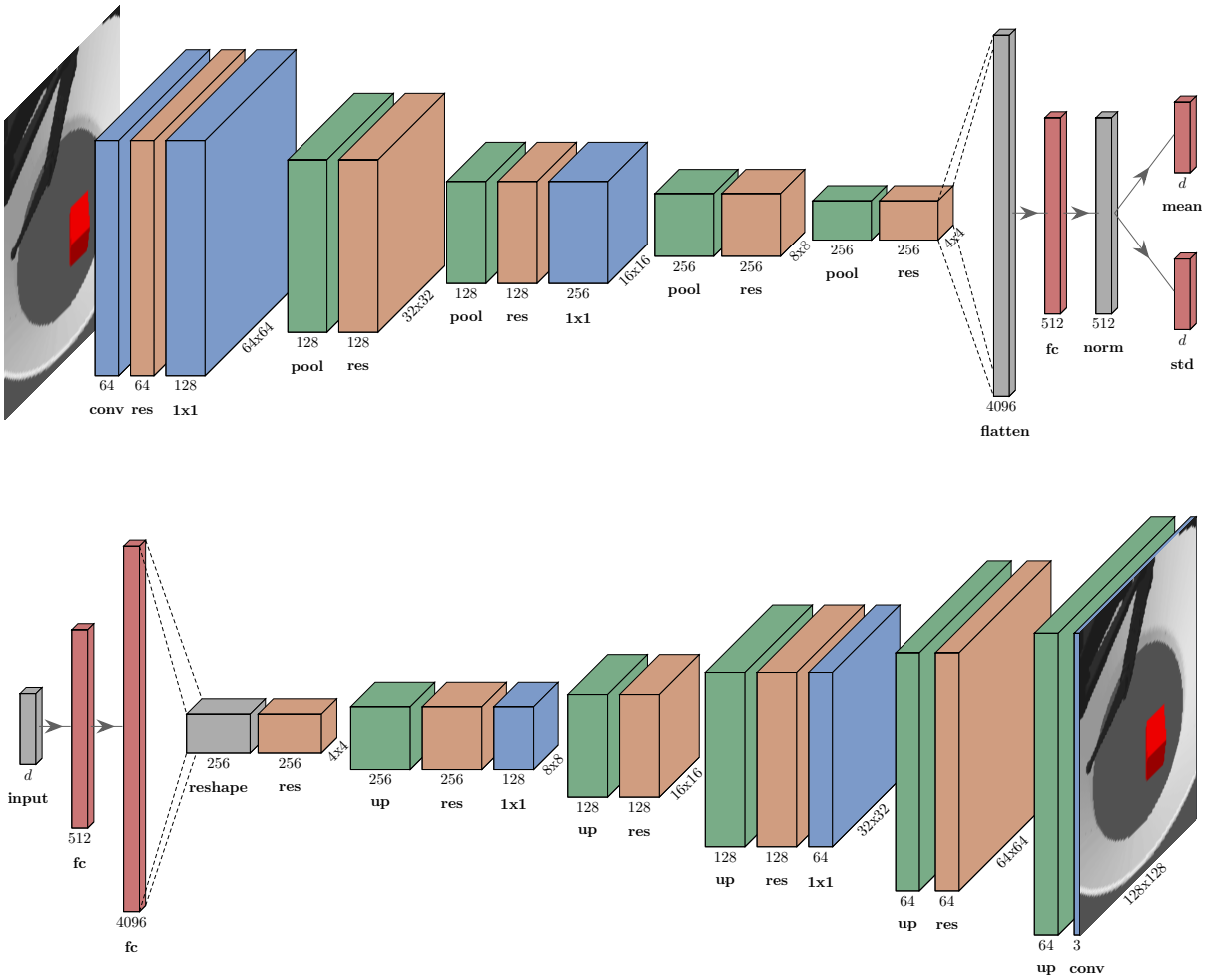


Figure D.1: Schemes of the encoder (top) and decoder (bottom) architectures. In both schemes, information flows left to right. Blue blocks represent convolutional layers: those labeled "conv" have 5x5 kernels and stride 2, while those labeled "1x1" have 1x1 kernels. Each orange block represents a pair of residual blocks (implementation details of a residual block are provided in Table D.2). Green blocks in the encoder represent average pooling with stride 2, and those in the decoder denote bilinear upsampling by a factor of 2. Red blocks represent fully-connected layers. The block labeled "norm" indicates layer normalization. Dashed lines denote tensor reshaping.

D.2 Additional Results

D.2.1 Dataset Correlations

Figure D.2 and Figure D.3 highlight the feasible states and density regarding the 2nd and 3rd DoF in the proposed dataset.

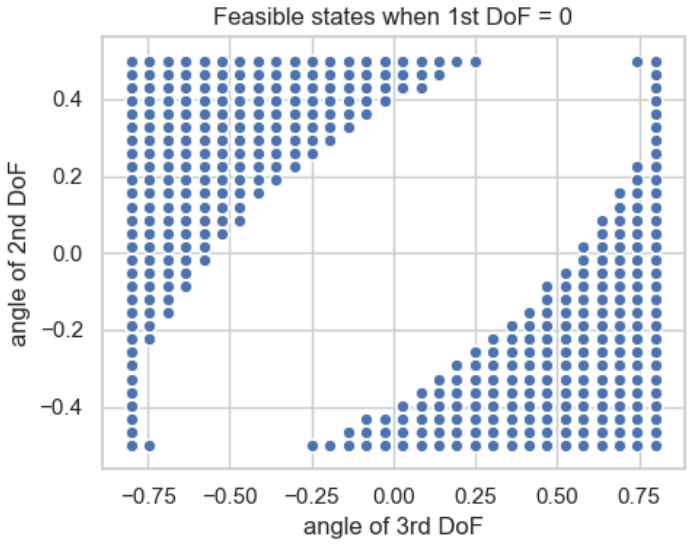


Figure D.2: Feasible states of the 2nd and 3rd DoF when the angle of the 1st DoF is 0. Angles are in radians.

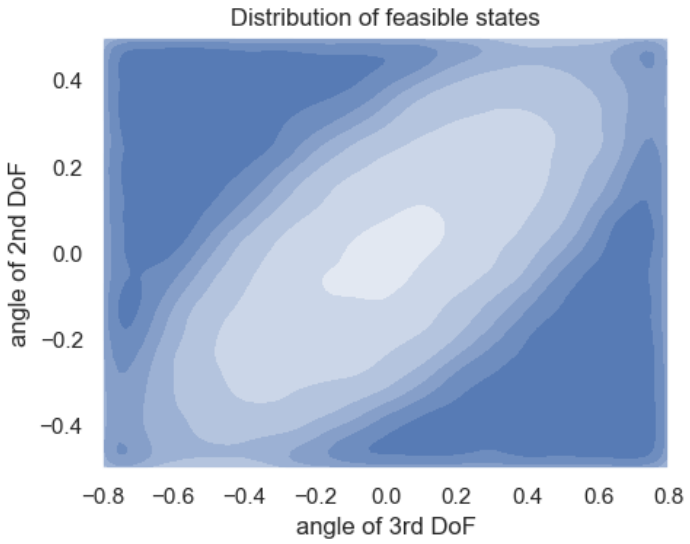


Figure D.3: Density of feasible states of 2nd and 3rd DoF over the whole training dataset. Darker shades of blue indicate regions of higher density. Angles are in radians.

D.2.2 Samples and Reconstructions

Figure D.4 and Figure D.5 show generated samples and input reconstructions by a trained model.

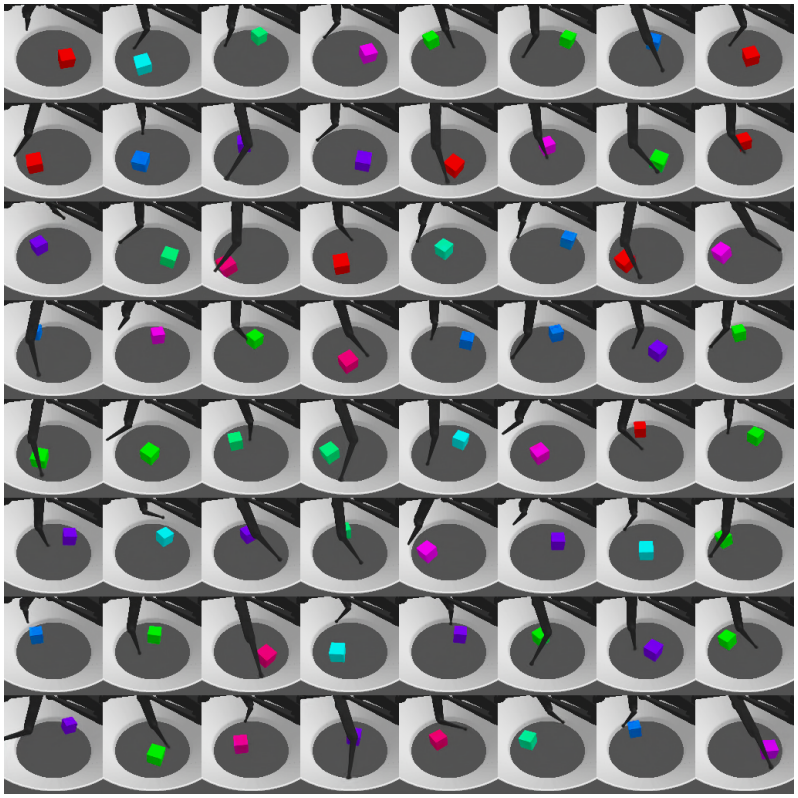


Figure D.4: Samples generated by a trained model. This model was selected based on the ELBO.

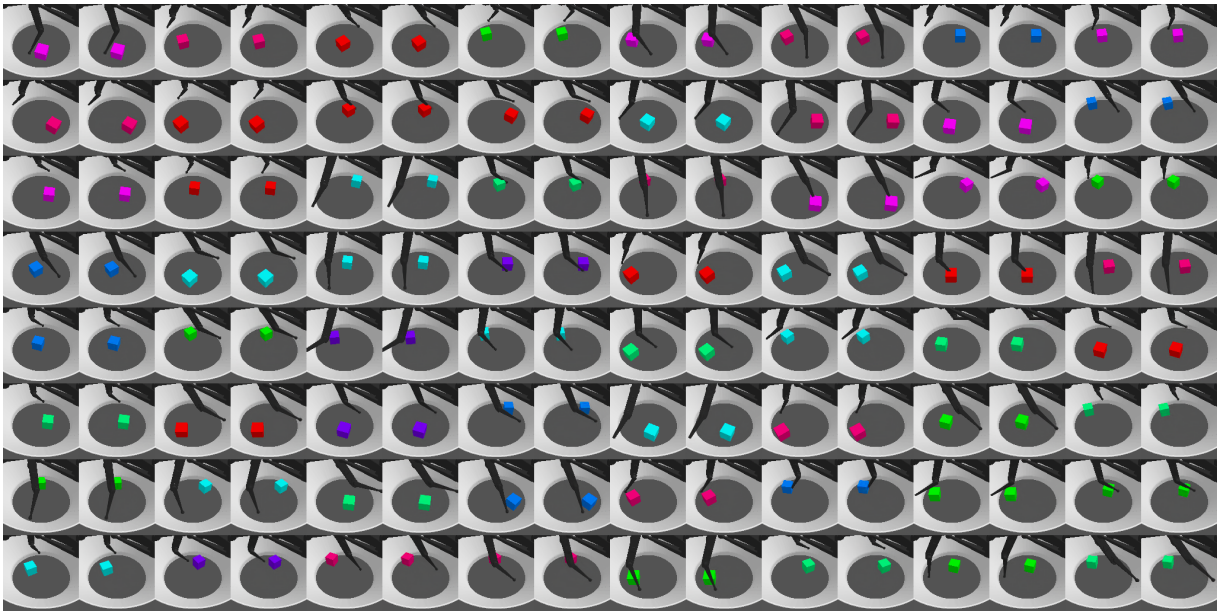
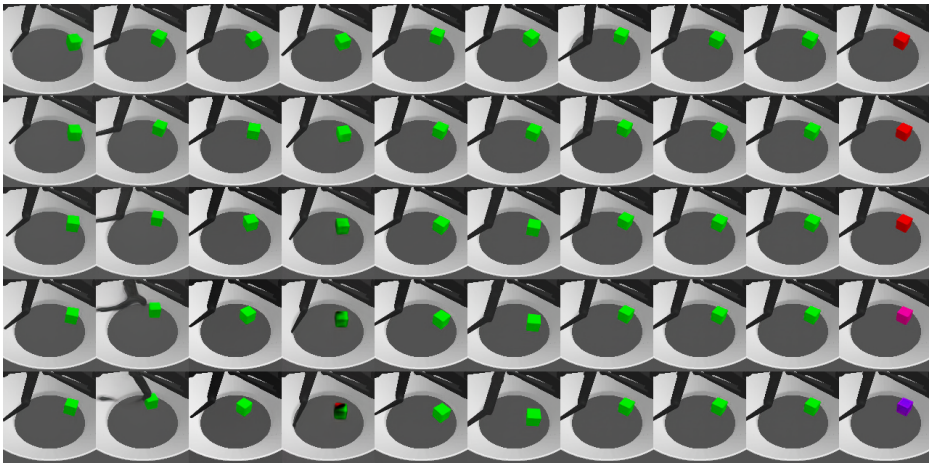


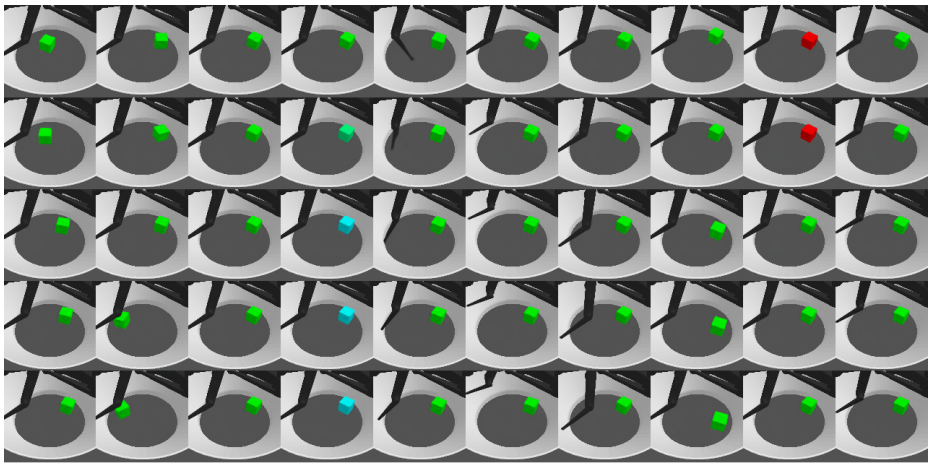
Figure D.5: Input reconstructions by a trained model. This model was selected based on the ELBO. Image inputs are on odd columns, reconstructions on even columns.

D.2.3 Latent Traversals

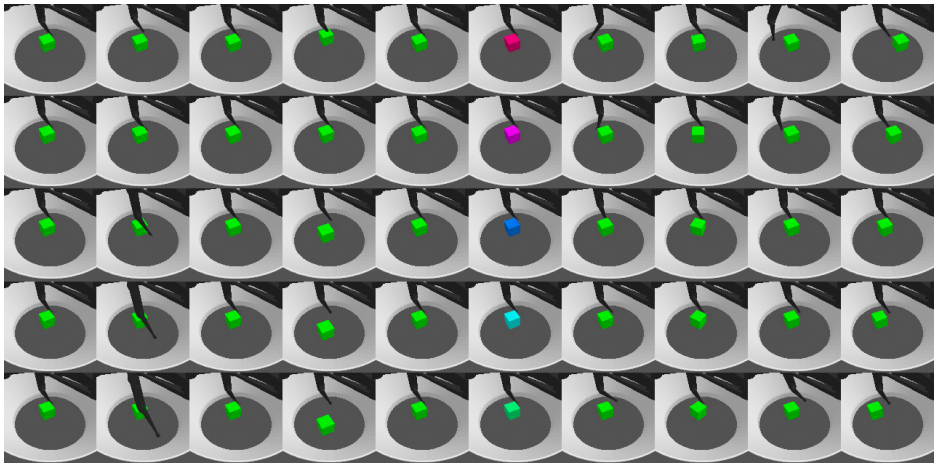
Figure D.6 show latent traversals for model with different DCI scores.



(a) Low disentanglement



(b) Medium disentanglement



(c) High disentanglement

Figure D.6: Latent traversals for a model with low DCI score (0.15) in (a), medium DCI score (0.5) in (b), and high DCI score (1.0) in (c).

D.2.4 Unsupervised Metrics and Disentanglement

Figure D.7 shows scatter plots of unsupervised metrics (ELBO and reconstruction loss) vs disentanglement (MIG, DCI) for 1,080 trained models.

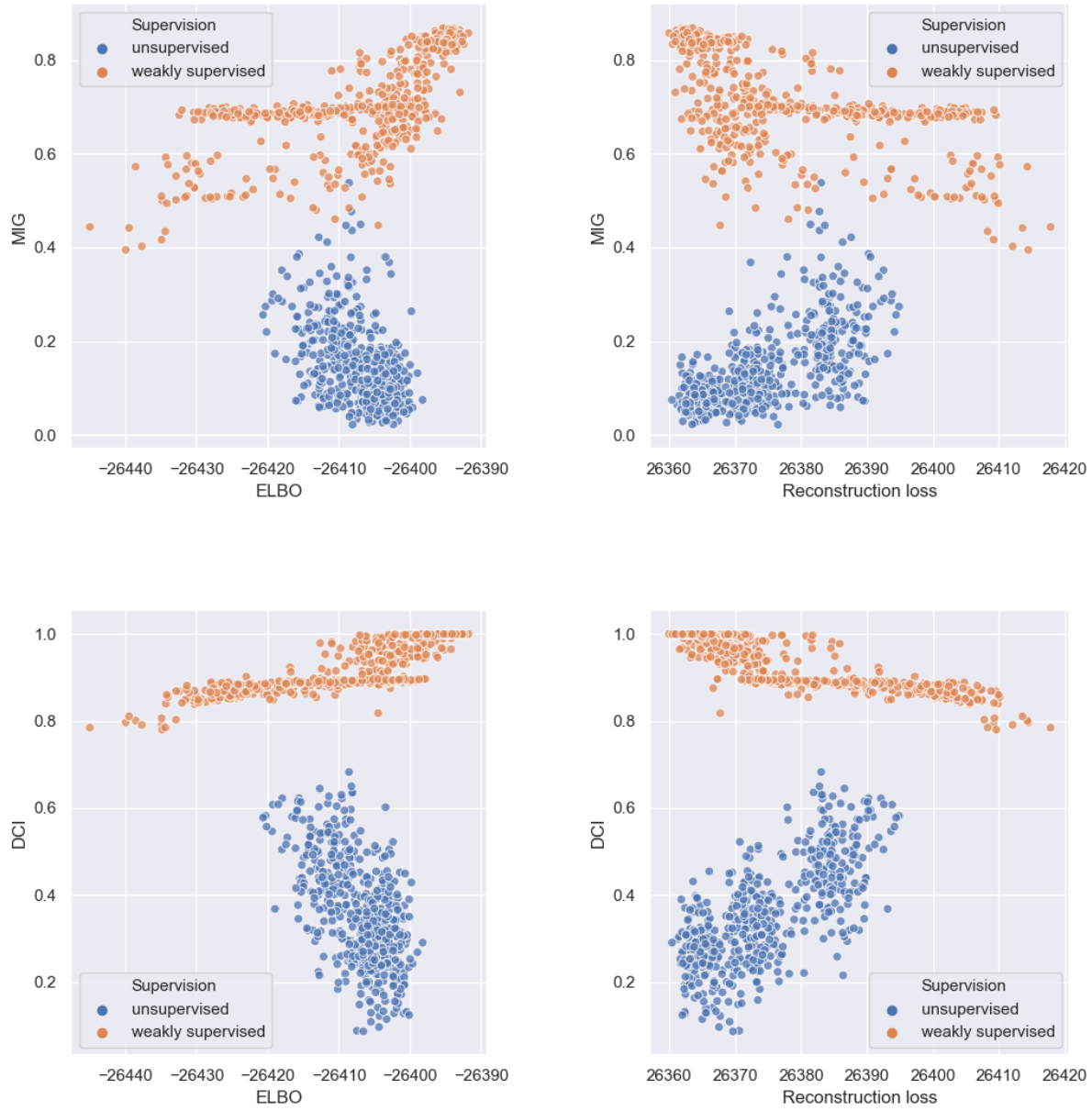


Figure D.7: Scatter plots of unsupervised metrics (left: ELBO; right: reconstruction loss) vs disentanglement (top: MIG; bottom: DCI) for 1,080 trained models, color-coded according to supervision. Each point represents a trained model.

D.2.5 Out-of-Distribution Transfer

Figure D.8 shows the transfer metric in OOD2-A and OOD2-B settings, decomposed according to the factor of variation and presence of input noise. When noise is added to the input during training, the inferred cube position error is relatively low (the scores are the mean absolute error, and they are normalized to [0, 1]). This is particularly useful in the OOD2-B setting (real world) where the joint state is anyway considered known, while object position has to be inferred with tracking methods.

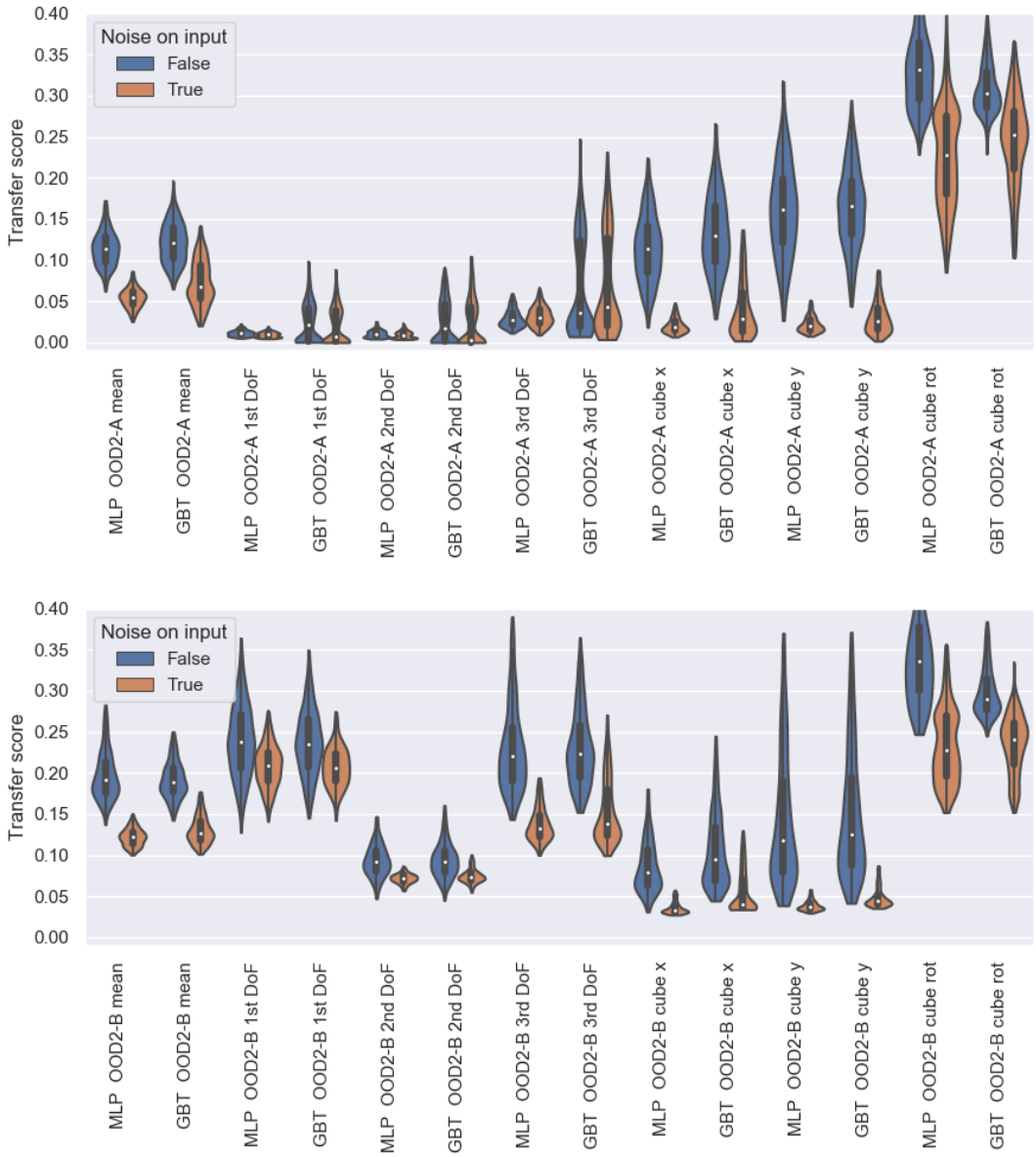
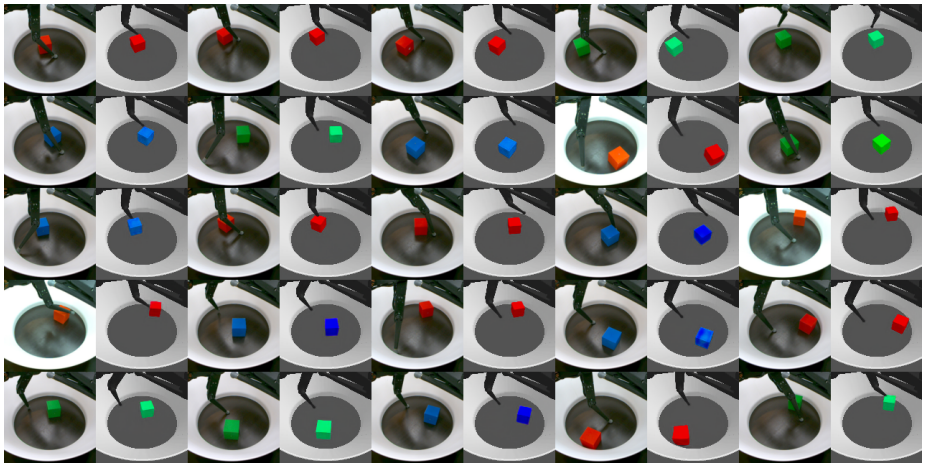


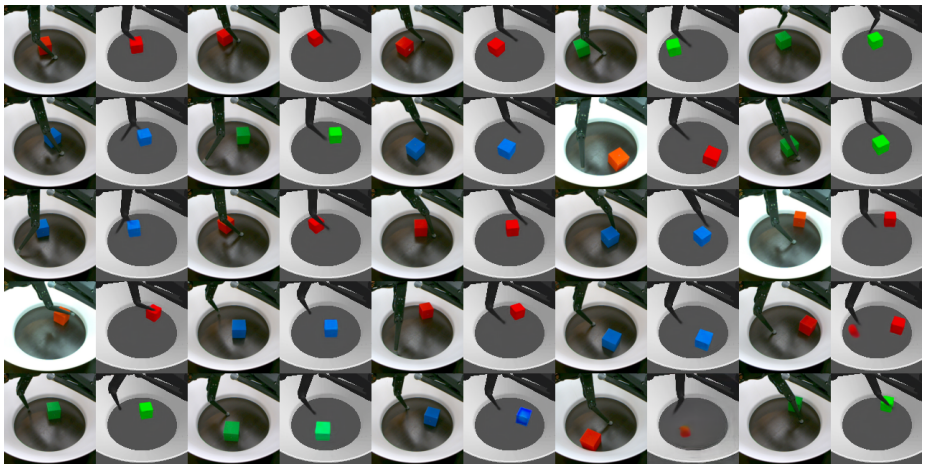
Figure D.8: Transfer metric in OOD2-A (top) and OOD2-B (bottom) settings, decomposed according to the factor of variation and presence of input noise.

D.2.6 Out-of-Distribution Reconstructions

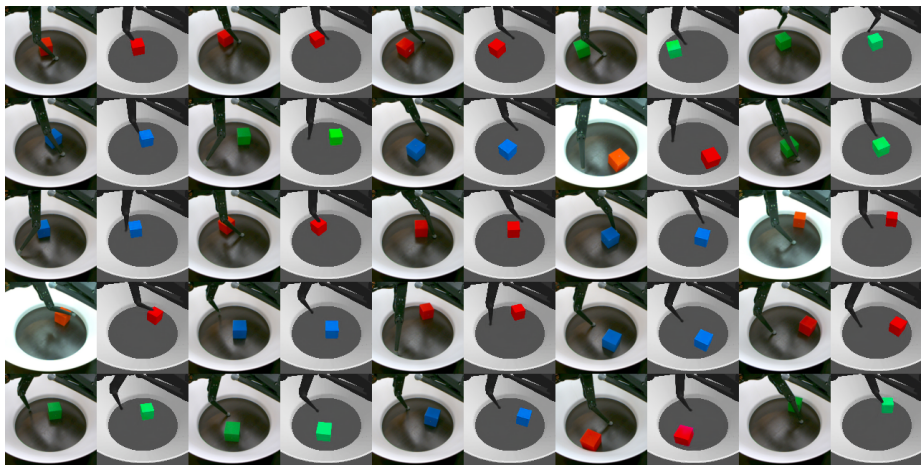
Figure D.9 and Figure D.10 show reconstructions of simulated and real-world images for models with different DCI scores.



(a) Low disentanglement

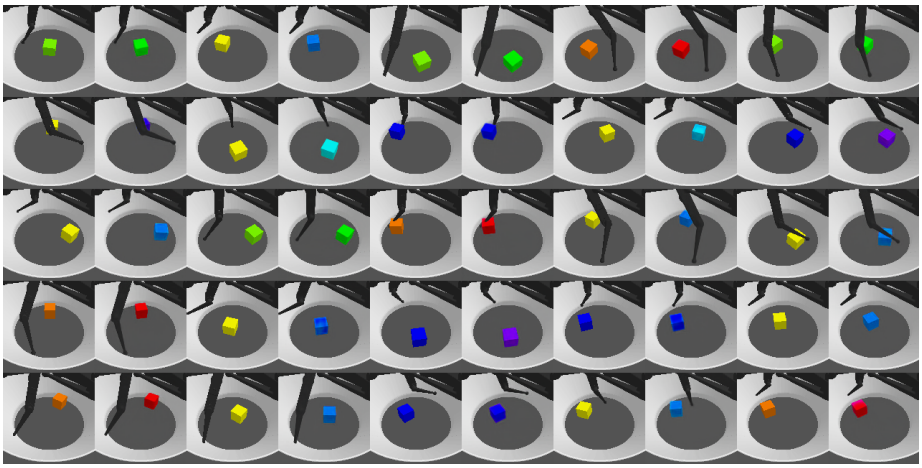


(b) Medium disentanglement

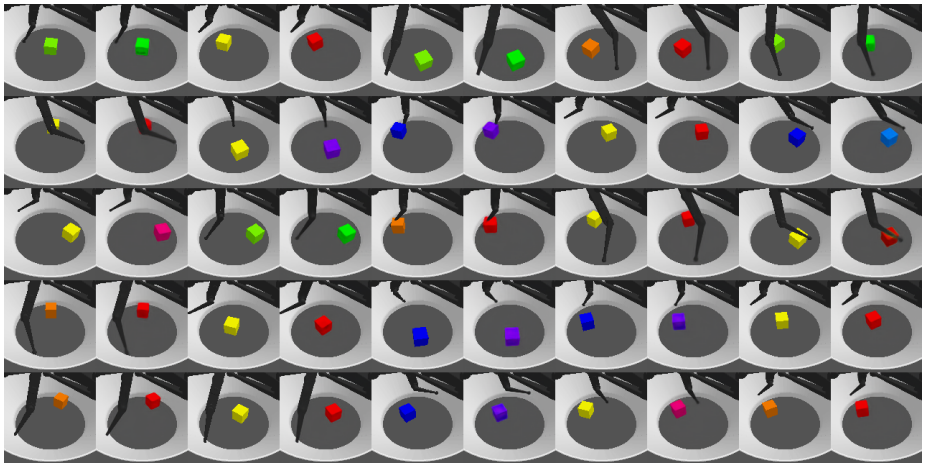


(c) High disentanglement

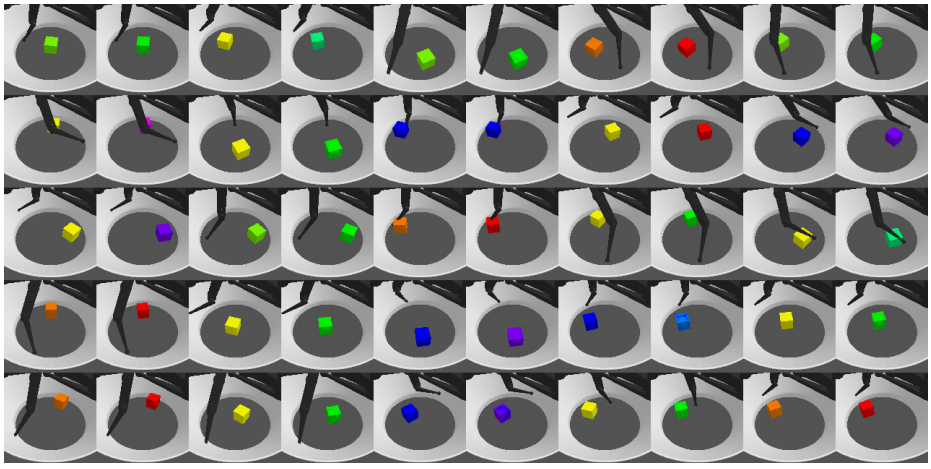
Figure D.9: Reconstructions of real-world images (OOD2-B) for a model with low DCI score (0.15) in (a), medium DCI score (0.5) in (b), and high DCI score (1.0) in (c).



(a) Low disentanglement



(b) Medium disentanglement



(c) High disentanglement

Figure D.10: Reconstructions of simulated images with encoder out-of-distribution colors (OOD2-A) for a model with low DCI score (0.15) in (a), medium DCI score (0.5) in (b), and high DCI score (1.0) in (c).

Appendix to Chapter 5 | E

E.1 Implementation Details

Task definitions and reward structure. We derive both tasks, *object reaching* and *pushing*, from the CausalWorld environments introduced by **Ahmed** et al. [2]. We pre-train representations on the dataset introduced by **Dittadi** et al. [3], and allow only one finger to move in our RL experiments. We introduce the *object reaching* environment that involves an unmovable cube. We used reward structures similar to those in **Ahmed** et al. [2]:

- *object reaching*: $r_t = -750\,[d(g_t, e_t) - d(g_{t-1}, e_{t-1})]$
- *pushing*: $r_t = -750\,[d(o_t, e_t) - d(o_{t-1}, e_{t-1})] - 250\,[d(o_t, g_t) - d(o_{t-1}, g_{t-1})] + \rho_t$

where t denotes the time step, $\rho_t \in [0, 1]$ is the fractional overlap with the goal cube at time t, $e_t \in \mathbf{R}^3$ is the end-effector position, $o_t \in \mathbf{R}^3$ the cube position, $g_t \in \mathbf{R}^3$ the goal position, and $d(\cdot, \cdot)$ denotes the Euclidean distance. The cube in *object reaching* is fixed, i.e. $o_t = g_t$ for all t. The time limit is 2 seconds in *object reaching* and 4 seconds in *pushing*.

Success metrics. Besides the accumulated reward along episodes, that is determined by the reward function, we also report two reward-independent normalized success definitions for better interpretability: In *object reaching*, the success metric indicates progress from the initial end effector position to the optimal distance from the center of the cube. It is 0 if the final distance is greater than or equal to the initial distance, and 1 if the end effector is touching the center of a face of the cube. In *pushing*, the success metric is defined as the volumetric overlap of the cube with the goal cube, and the task can be visually considered solved with a score around 80%. We observed that accumulated reward and success are very strongly correlated, thus allowing us to use one or the other for measuring performance.

Training and evaluation details. During training, the goal position is randomly sampled at every episode. Similarly, the object color is sampled from the 4 specified training colors from $\mathcal{D}_1$ that correspond to the OOD1-B split from [3].

For each policy evaluation (in-distribution and out-of-distribution variants), we average the accumulated reward and final success over 200 episodes with randomly sampled cube positions and the respective object color in both tasks.

SAC implementation. Our implementation of SAC builds upon the `stable-baselines` package [394]. We use the same SAC hyperparameters used for pushing in [2]. When using L1 regularization, we add to the loss function the L1 norm of the first layers of all MLPs, scaled by a coefficient α. We gradually increase regularization by linearly annealing α from 0 to $5 \cdot 10^{-7}$ over 200,000 time steps in *object reaching*, and from 0 to $6 \cdot 10^{-8}$ over 3,000,000 time steps in *pushing*.

E.2 Additional Results

E.2.1 Training environment

Figure 5.2 in Section 5.4.1 shows correlations of unsupervised and supervised metrics with in-distribution reward for *object reaching* and *pushing*, only in the case without regularization. In Figure E.1 we also show these results in the case with regularization, as well as when adjusting for MLP informativeness.

E.2.2 Out-of-distribution generalization in simulation

In Section 5.4.2 we discussed rank-correlations of OOD rewards with unsupervised, informativeness and generalization scores on *object reaching* without regularization. In Figure E.2 we also show these results for the case with regularization and on *pushing*, as well as when adjusting for MLP informativeness. Without regularization, we observe on *pushing* very similar correlations along all metrics as we observed on *object reaching*, confirming our conclusions on this more complex task. When using regularization, rank correlations are very similar across both tasks. Interestingly, the correlation between GS-OOD2 scores and OOD2 generalization of the policy is even stronger when using L1 regularization. In contrast to our observations without regularization, we find that the correlation between GS-OOD1 and OOD1 generalization of the policy vanishes when adjusting for the MLP metric.

Disentangled representations

As discussed in Section 5.4.2 for *object reaching* without regularization, we observe in Figure E.2 a weak correlation between some disentanglement metrics and OOD1 reward, which however vanishes when adjusting for MLP informativeness. In agreement with [3], we observe no significant correlation between disentanglement and OOD2 generalization, for both tasks, with and without regularization. From Figure E.3 we see that in some cases, especially without regularization, a very high DCI score seems to lead to better performance on average. However, this behavior is not significant (within error bars), as opposed to the results shown in simpler downstream tasks by **Dittadi** et al. [3]. Furthermore, this trend is likely due to representation informativeness, since the correlations with disentanglement disappear when adjusting for the MLP score, as discussed above.

Regularization

As seen in Figure E.3, regularization generally has a positive effect on OOD1 and OOD2 generalization, which is particularly prominent in the OOD1 setting. On the other hand, it leads to lower training rewards both in *object reaching* and in *pushing*. In the latter, the performance drop is particularly significant, while in *object reaching* it is negligible.

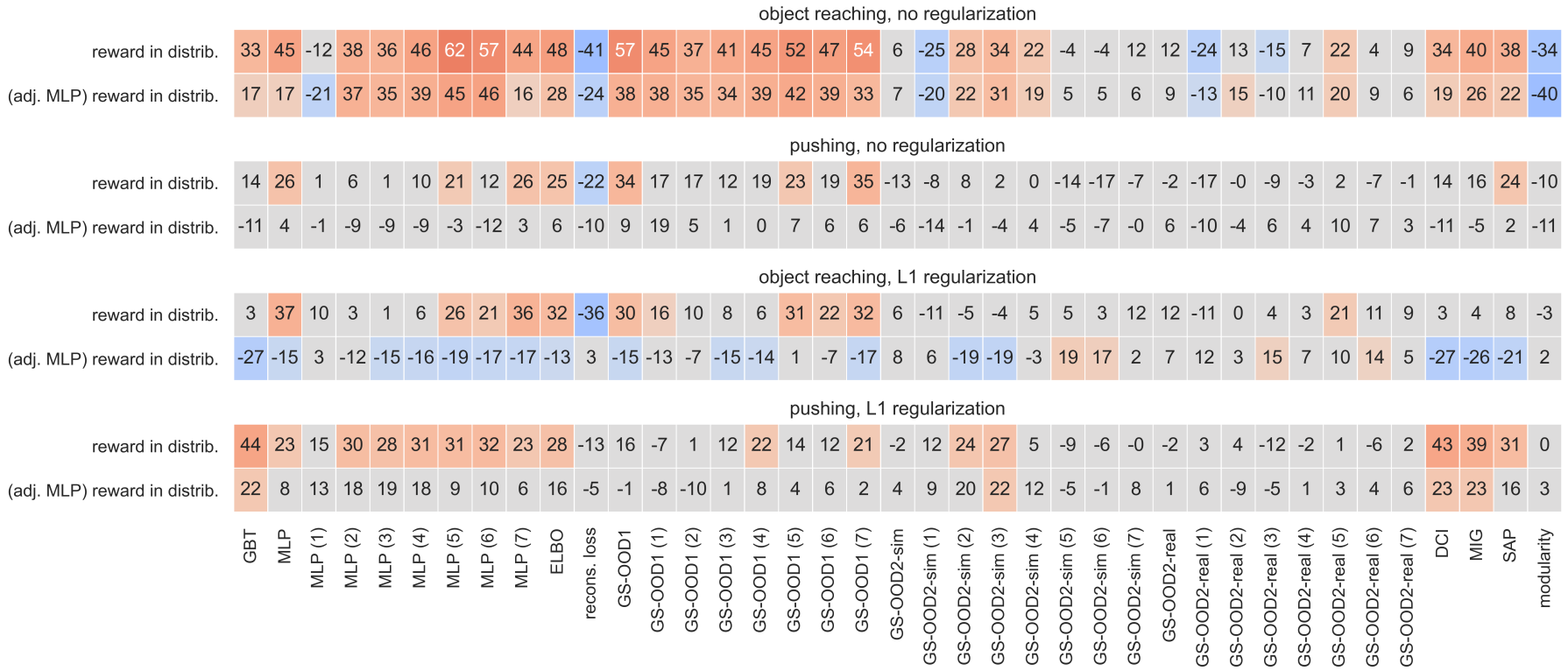


Figure E.1: Rank correlations between metrics and in-distribution reward, with and without adjusting for informativeness. Correlations are color-coded as described in Figure 5.2.

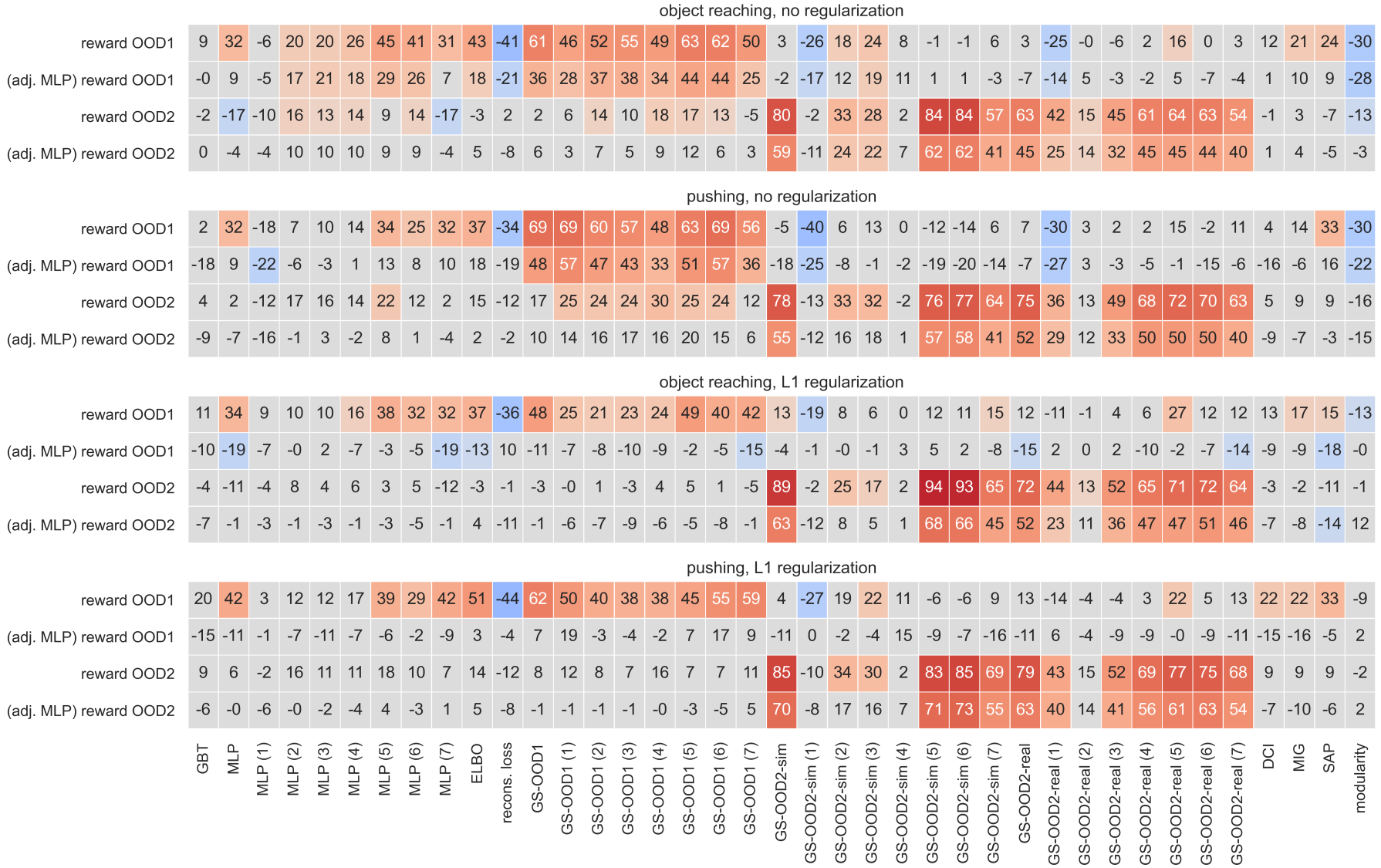


Figure E.2: Rank correlations between metrics and OOD reward, with and without adjusting for informativeness. Correlations are color-coded as described in Figure 5.2.

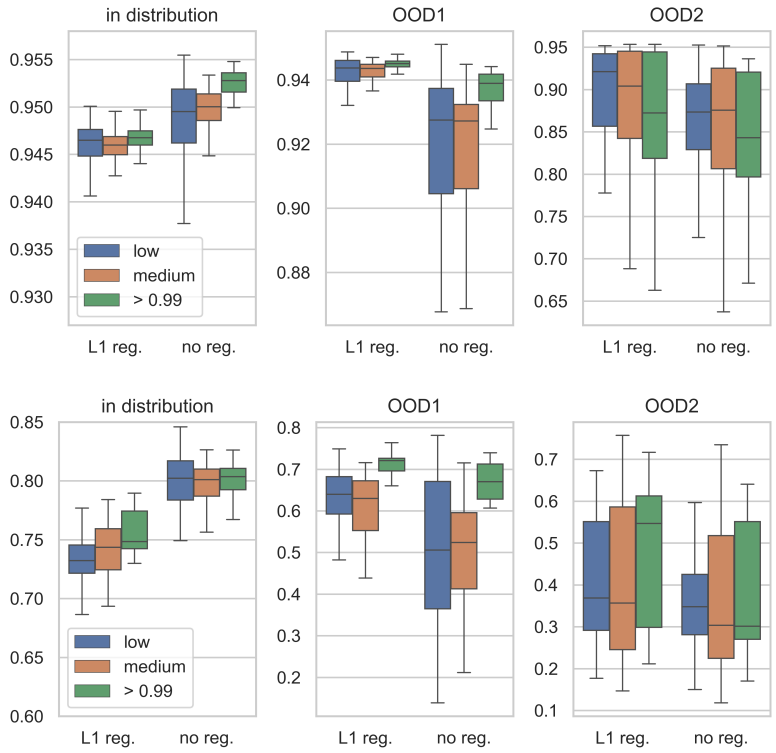


Figure E.3: Fractional success on *object reaching* (**top**) and *pushing* (**bottom**), split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. Results for *object reaching* are also reported in Figure 5.6 in Section 5.4.2.

Sample efficiency

In addition to the analysis reported in the main paper, we investigate how representation properties affect sample efficiency. Specifically, we store checkpoints of our policies at $t \in \{20k, 50k, 100k, 400k\}$ for *object reaching* and $t \in \{200k, 500k, 1M, 3M\}$ for *pushing*. We then evaluate policies at these time step on the same three environments as before: (1) on the cube colors from training; (2) on the OOD1 cube colors; and (3) on the OOD2-sim cube colors. Results are summarized in Figure E.4 for *object reaching* and Figure E.5 for *pushing*.

On *object reaching* (Figure E.4), we observe very similar trends with and without regularization: Unsupervised metrics (ELBO and reconstruction loss) display a correlation with the training reward, as do the supervised informativeness metrics (GBT and MLP). This is strongest on early timesteps, meaning these scores could be important for sample efficiency. Similarly, we observe a correlation with the disentanglement scores DCI, MIG and SAP. With the help of the additional evaluation of rewards adjusted for MLP informativeness, we can attribute this correlation again to this common confounder. Crucially, we see that the generalization scores (GS) are correlated with generalization of the corresponding policies under OOD1 and OOD2 shifts for all recorded time steps, confirming the results in the main text.

On *pushing* (Figure E.5), many correlations at early checkpoints are significantly reduced, especially with regularization. This behavior might be due to the more complicated nature of the task, which involves learning to reach the cube first, and then push it to the goal. Correlations are primarily seen towards the end of training, with similar spurious correlations with disentanglement as elaborated above. Importantly, correlations between generalization scores (GS) and policy generalization under

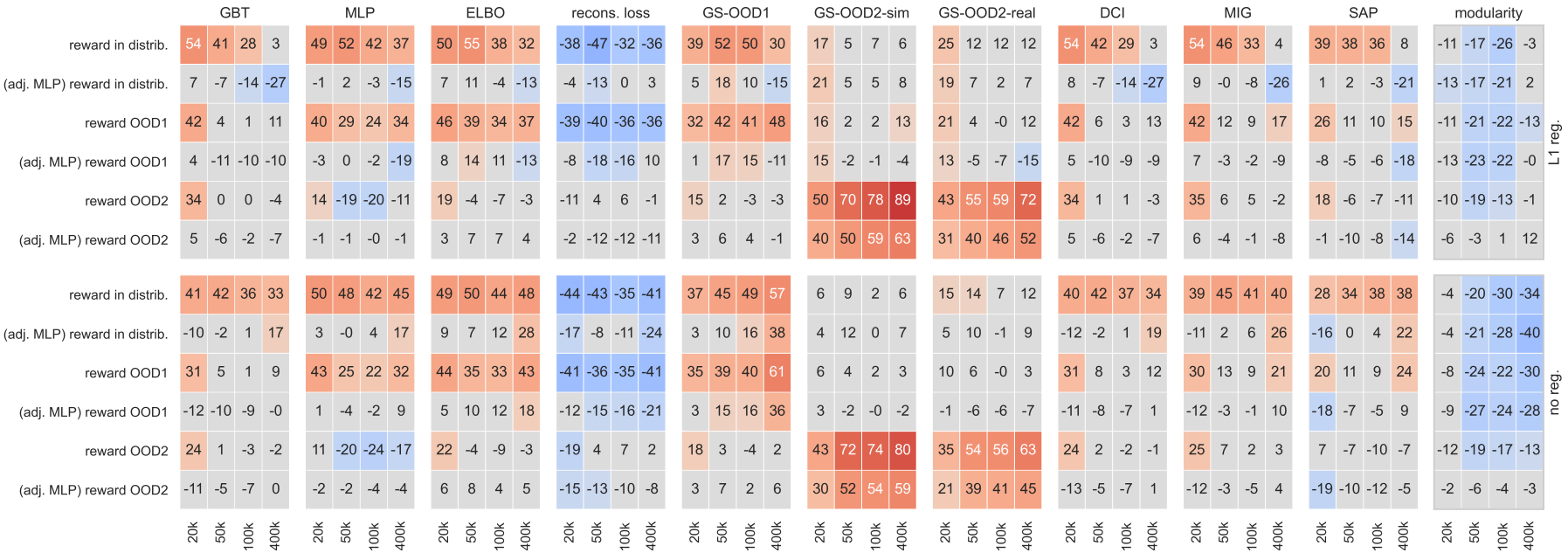


Figure E.4: Sample efficiency analysis for *object reaching*. Rank correlations of rewards with relevant metrics along multiple time steps. Correlations are color-coded as described in Figure 5.2.

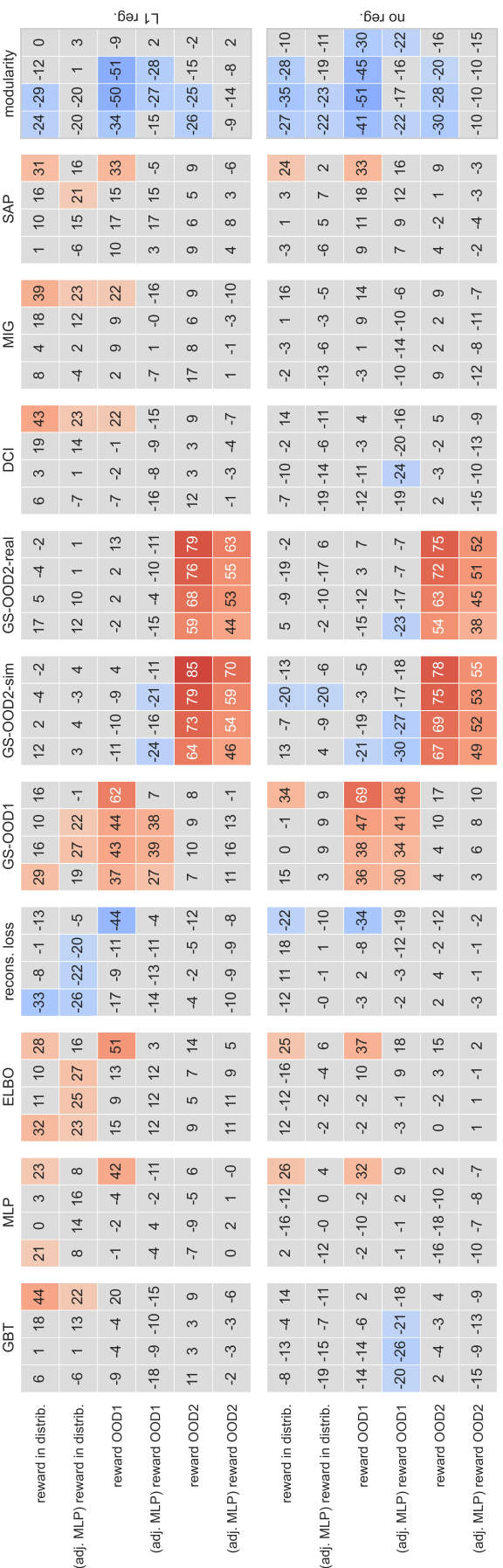


Figure E.5: Sample efficiency analysis for *pushing*. Rank correlations of rewards with relevant metrics along multiple time steps. Correlations are color-coded as described in Figure 5.2.

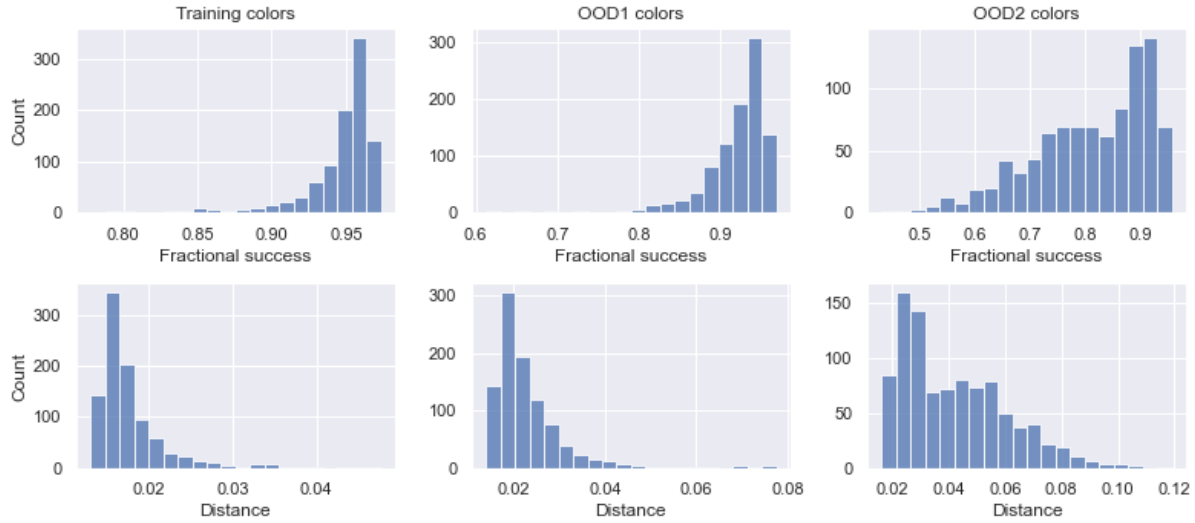


Figure E.6: Testing policies for *object reaching* under the same in-distribution, OOD1, and OOD2 evaluation protocols regarding object color in simulation, but replacing the cube with a sphere, which was never used in training.

the same distribution shifts remain strong and statistically significant, corroborating the analysis in the main text.

Generalization to a novel shape

As mentioned in Section 5.4.2, on the *object reaching* task, we also test generalization w.r.t. a novel object shape by replacing the cube with an unmovable sphere. This corresponds to a strong OOD2-type shift, since shape was never varied when training the representations. We then evaluate a subset of 960 trained policies as before, with the same color splits. Surprisingly, the policies appear to handle the novel shape as we see from the histograms in Figure E.6 in terms of success and final distance. In fact, when the sphere has the same colors that the cube had during policy training, *all* policies get closer than 5 cm to the sphere on average, with a mean success metric of about 95%. On sphere colors from the OOD1 split, more than 98.5% move the finger closer than this threshold, and on the strongest distribution shift (OOD2-sim colors and cube replaced by sphere) almost 70% surpass that threshold with an average success metric above 80%.

E.2.3 Deploying policies to the real world

In Figure E.7 we show three representative episodes of testing a reaching policy on the real robot for the strong OOD shift with a novel sphere object shape instead of the cube from training. We present the respective videos in the project page. There we also present videos of additional real-world episodes on pushing and reaching cubes of different colors.

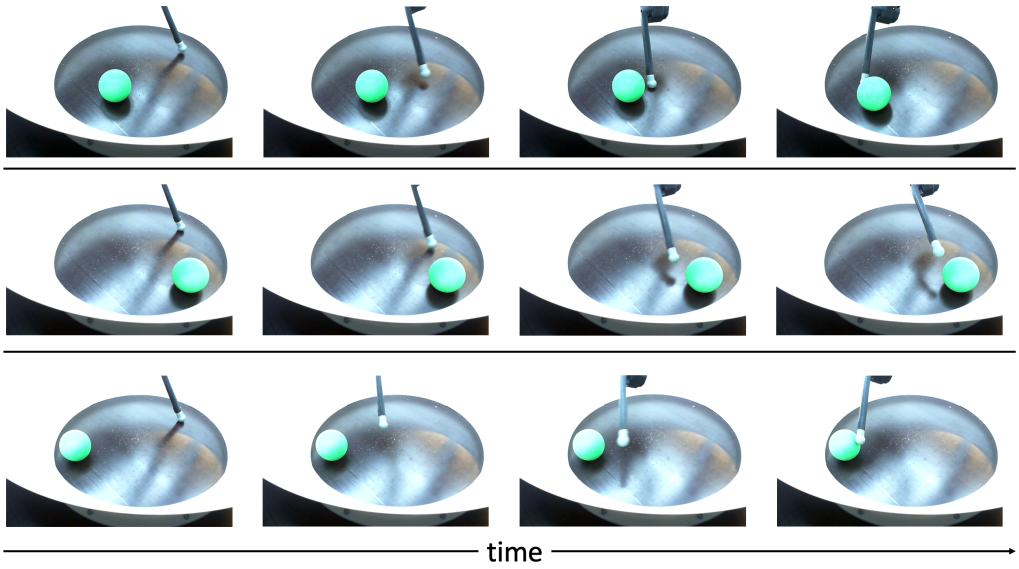


Figure E.7: Transferring policies for *object reaching* to the real robot setup without any fine-tuning on a green sphere (unseen shape *and* color). Correlations are color-coded as described in Figure 5.2.

Appendix to Chapter 6 F

F.1 Proof of Theorem

Proof. Let $f \in \{f_\theta^S, \hat{f}^S\}$. We define

$$\mathcal{C}_k^y = \left\{ (\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y} : y = \bar{y}, k = \operatorname*{argmin}_{i \in [|\mathcal{K}|]} d((k_\beta \circ z_\alpha)(\bar{x}), \mathcal{K}_i) \right\}.$$

We use the notation of $z = (x, y)$ and $\tilde{x}^\epsilon = g_\epsilon(x)$. We first write the expected error as the sum of the conditional expected error:

$$\mathbb{E}_z[\ell(f(\tilde{x}^\epsilon), y)] = \sum_{k,y} \mathbb{E}_x[\ell(f(\tilde{x}^\epsilon), y) | z \in \mathcal{C}_k^y] \Pr(z \in \mathcal{C}_k^y) = \sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_{k,y}^\epsilon), y)] \Pr(z \in \mathcal{C}_k^y),$$

where $\tilde{x}_{k,y}^\epsilon = g_\epsilon(x_{k,y})$ and $x_{k,y}$ is the random variable x conditioned on $z \in \mathcal{C}_k^y$. Using this, we decompose the generalization error into two terms:

$$\begin{aligned} \mathbb{E}_z[\ell(f(\tilde{x}^\epsilon), y)] - \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) = \sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_{k,y}^\epsilon), y)] \left(\Pr(z \in \mathcal{C}_k^y) - \frac{|\mathcal{I}_k^y|}{n} \right) \\ + \left(\sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_{k,y}^\epsilon), y)] \frac{|\mathcal{I}_k^y|}{n} - \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) \right). \end{aligned} \tag{F.1}$$

Since $\frac{1}{n} \sum_{i=1}^n \ell(h, r_i) = \frac{1}{n} \sum_y \sum_{k \in I_\mathcal{K}^y} \sum_{i \in \mathcal{I}_k^y} \ell(f(x_i), y_i)$, the second term in the right-hand side of (F.1) is further simplified as

$$\begin{aligned} &\sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_{k,y}^\epsilon), y)] \frac{|\mathcal{I}_k^y|}{n} - \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) \\ &= \frac{1}{n} \sum_y \sum_{k \in I_\mathcal{K}^y} |\mathcal{I}_k^y| \left(\mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_{k,y}^\epsilon), y)] - \frac{1}{|\mathcal{I}_k^y|} \sum_{i \in \mathcal{I}_k^y} \ell(f(x_i), y_i) \right). \end{aligned}$$

where $I_\mathcal{K}^y = \{k \in [|\mathcal{K}|] : |\mathcal{I}_k^y| \geq 1\}$. Substituting these into equation (F.1) yields

$$\begin{aligned} \mathbb{E}_z[\ell(f(\tilde{x}^\epsilon), y)] - \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) = \sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_k^\epsilon), y)] \left(\Pr(z \in \mathcal{C}_k^y) - \frac{|\mathcal{I}_k^y|}{n} \right) \\ + \frac{1}{n} \sum_y \sum_{k \in I_\mathcal{K}^y} |\mathcal{I}_k^y| \left(\mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_{k,y}^\epsilon), y)] - \frac{1}{|\mathcal{I}_k^y|} \sum_{i \in \mathcal{I}_k^y} \ell(f(x_i), y_i) \right) \end{aligned} \tag{F.2}$$

By using Lemma 1 of [395], we have that for any $\delta > 0$, with probability at least $1 - \delta$,

$$\begin{aligned}
&\sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_k^\epsilon), y)] \left(\Pr(z \in \mathcal{C}_k^y) - \frac{|\mathcal{I}_k^y|}{n} \right) \\
&\leq \left(\sum_{k,y} \mathbb{E}_{x_{k,y}}[\ell(f(\tilde{x}_k^\epsilon), y)] \sqrt{\Pr(z \in \mathcal{C}_k^y)} \right) \sqrt{\frac{2\ln(|\mathcal{Y}||\mathcal{K}|/\delta)}{n}} \\
&\leq M \left(\sum_{k,y} \sqrt{\Pr(z \in \mathcal{C}_k^y)} \right) \sqrt{\frac{2\ln(|\mathcal{Y}||\mathcal{K}|/\delta)}{n}}.
\end{aligned} \tag{F.3}$$

Here, we have that $\ln(|\mathcal{Y}||\mathcal{K}|/\delta) = \ln(e^{\ln|\mathcal{Y}||\mathcal{K}|}/\delta) \leq \ln(e^\zeta/\delta) \leq \ln((e/\delta)^\zeta) = \zeta \ln(e/\delta)$ where $\zeta = \max(1, \ln|\mathcal{Y}||\mathcal{K}|)$, since $\delta \in (0,1)$ and $\zeta \geq 1$. Moreover, note that for any (f, h, M) such that $M > 0$ and $B \geq 0$ for all X, we have that $\mathbb{P}(f(X) \geq M) \geq \mathbb{P}(f(X) > M) \geq \mathbb{P}(Bf(X) + h(X) > BM + h(X))$, where the probability is with respect to the randomness of X. Thus, by combining (F.2) and (F.3), we have that for any $\delta > 0$, with probability at least $1 - \delta$, the following holds:

$$\begin{aligned}
&\mathbb{E}_{z,\epsilon}[\ell(f(g_\epsilon(x)), y)] - \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i) \\
&= \mathbb{E}_\epsilon \left[\mathbb{E}_z[\ell(f(g_\epsilon(x)), y)] - \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i) \right] \\
&\leq \frac{1}{n}\sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} |\mathcal{I}_k^y| \left(\mathbb{E}_{z,\epsilon}[\ell(f(g_\epsilon(x)), y)|z \in \mathcal{C}_k^y] - \frac{1}{|\mathcal{I}_k^y|}\sum_{i\in\mathcal{I}_k^y} \ell(f(x_i), y_i) \right) + c\sqrt{\frac{2\ln(e/\delta)}{n}} \\
&= \frac{1}{n}\sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} |\mathcal{I}_k^y| \left(\mathbb{E}_z[\ell(f(x), y)|z \in \mathcal{C}_k^y] - \frac{1}{|\mathcal{I}_k^y|}\sum_{i\in\mathcal{I}_k^y} \ell(f(x_i), y_i) \right) \\
&\quad + \frac{1}{n}\sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} |\mathcal{I}_k^y| \mathbb{E}_z \left[\mathbb{E}_\epsilon[\ell(f(g_\epsilon(x)), y)] - \ell(f(x), y)|z \in \mathcal{C}_k^y \right] \\
&\quad + c\sqrt{\frac{2\ln(e/\delta)}{n}}
\end{aligned} \tag{F.4}$$

We now bound each term in the right-hand side of equation (F.4) for both cases of $f = \hat{f}^S$ and $f = f_\theta^S$. We first consider the case of $f = \hat{f}^S$. For the first term in the right-hand side of equation (F.4), we invoke Lemma 4 of [396] to obtain that for any $\delta > 0$, with probability at least $1 - \delta$,

$$\begin{aligned}
&\frac{1}{n}\sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} |\mathcal{I}_k^y| \left(\mathbb{E}_z[\ell(\hat{f}^S(x), y)|z \in \mathcal{C}_k^y] - \frac{1}{|\mathcal{I}_k^y|}\sum_{i\in\mathcal{I}_k^y} \ell(\hat{f}^S(x_i), y_i) \right) \\
&\leq \frac{1}{n}\sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} |\mathcal{I}_k^y| \left(2\mathcal{R}_{y,k}(\ell \circ \hat{\mathcal{F}}) + M\sqrt{\frac{\ln(|\mathcal{Y}||\mathcal{K}|/\delta)}{2|\mathcal{I}_k^y|}} \right) \\
&= 2\sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} |\mathcal{I}_k^y| \frac{\mathcal{R}_{y,k}(\ell \circ \hat{\mathcal{F}})}{n} + M\sqrt{\frac{\ln(|\mathcal{Y}||\mathcal{K}|/\delta)}{2n}} \sum_{y\in\mathcal{Y}}\sum_{k\in I_\mathcal{K}^y} \sqrt{\frac{|\mathcal{I}_k^y|}{n}}
\end{aligned}$$

where we used the fact that $\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}|\mathcal{I}_k^y| = n$. Moreover, we have that using the Cauchy–Schwarz inequality,

$$\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}\sqrt{\frac{|\mathcal{I}_k^y|}{n}} \leq \sqrt{\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}\frac{|\mathcal{I}_k^y|}{n}}\sqrt{\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}1} \leq \sqrt{|\mathcal{Y}||\mathcal{K}|}.$$

Thus, since $\ln(|\mathcal{Y}||\mathcal{K}|/\delta) \leq \zeta\ln(e/\delta)$ with $\zeta = \max(1, \ln|\mathcal{Y}||\mathcal{K}|)$ (see above), for any $\delta > 0$, with probability at least $1-\delta$,

$$\begin{aligned}&\frac{1}{n}\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}|\mathcal{I}_k^y|\left(\mathbb{E}_z[\ell(\hat{f}^S(x),y)|z\in\mathcal{C}_k^y] - \frac{1}{|\mathcal{I}_k^y|}\sum_{i\in\mathcal{I}_k^y}\ell(\hat{f}^S(x_i),y_i)\right) \\ &\leq 2\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}|\mathcal{I}_k^y|\frac{\mathcal{R}_{y,k}(\ell\circ\hat{\mathcal{F}})}{n} + c\sqrt{\frac{\ln(e/\delta)}{2n}}.\end{aligned} \tag{F.5}$$

For the case of $f = \hat{f}^S$, by combining equation (F.4) and (F.5) with union bound, it holds that any $\delta > 0$, with probability at least $1-\delta$,

$$\begin{aligned}&\mathbb{E}_{z,\epsilon}[\ell(\hat{f}^S(g_\epsilon(x)),y)] - \frac{1}{n}\sum_{i=1}^n\ell(\hat{f}^S(x_i),y_i) \\ &\leq 2\sum_{y\in\mathcal{Y}}\sum_{k\in I_{\mathcal{K}}^y}|\mathcal{I}_k^y|\frac{\mathcal{R}_{y,k}(\ell\circ\hat{\mathcal{F}})}{n} + c\sqrt{\frac{\ln(2e/\delta)}{2n}} + c\sqrt{\frac{2\ln(2e/\delta)}{n}} \\ &\quad + \frac{1}{n}\sum_{y,k}|\mathcal{I}_k^y|\mathbb{E}_{x\sim\mathcal{D}_{x|y},\epsilon}\left[\ell(\hat{f}^S(g_\epsilon(x)),y) - \ell(\hat{f}^S(x),y)|x\in\mathcal{C}_k\right].\end{aligned}$$

We now consider the case of $f = f_\theta^S$. For the first term in the right-hand side of equation (F.4),

$$\begin{aligned}&\mathbb{E}_z[\ell(f_\theta^S(x),y)|z\in\mathcal{C}_k^y] - \frac{1}{|\mathcal{I}_k^y|}\sum_{i\in\mathcal{I}_k^y}\ell(f_\theta^S(x_i),y_i) \\ &= \mathbb{E}_z[\ell((d_\delta\circ v_\gamma\circ k_\beta\circ z_\alpha)(x),y)|z\in\mathcal{C}_k^y] - \frac{1}{|\mathcal{I}_k^y|}\sum_{i\in\mathcal{I}_k^y}\ell((d_\delta\circ v_\gamma\circ k_\beta\circ z_\alpha)(x_i),y_i) \\ &= \ell((d_\delta\circ v_\gamma)(\mathcal{K}_k),y) - \ell((d_\delta\circ v_\gamma)(\mathcal{K}_k),y) = 0.\end{aligned} \tag{F.6}$$

For the case of $f = f_\theta^S$, by combining equation (F.4) and (F.6), it holds that any $\delta > 0$, with probability at least $1-\delta$,

$$\begin{aligned}&\mathbb{E}_{z,\epsilon}[\ell(f_\theta^S(g_\epsilon(x)),y)] - \frac{1}{n}\sum_{i=1}^n\ell(f_\theta^S(x_i),y_i) \\ &\leq c\sqrt{\frac{2\ln(2e/\delta)}{n}} + \frac{1}{n}\sum_{y,k}|\mathcal{I}_k^y|\mathbb{E}_{x\sim\mathcal{D}_{x|y},\epsilon}\left[\ell(f_\theta^S(g_\epsilon(x)),y) - \ell(f_\theta^S(x),y)|x\in\mathcal{C}_k\right]\end{aligned}$$

Moreover, for the case of $f = f_\theta^S$, if $x \in \mathcal{C}_k \Rightarrow g_\epsilon(x) \in \mathcal{C}_k$ with probability one, then

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{D}_{x|y},\epsilon}\left[\ell(f_\theta^S(g_\epsilon(x)), y) - \ell(f_\theta^S(x), y)|x \in \mathcal{C}_k\right] &= \ell((d_\delta \circ v_\gamma)(\mathcal{K}_k), y) - \ell((d_\delta \circ v_\gamma)(\mathcal{K}_k), y) \\ &= 0.\end{aligned}$$

□

F.2 Why do random down-projection matrices help?

In addition to the flexibility in reducing the term $\frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i)$ when having multiple codebooks, the benefit from random projections in terms of Theorem 6.3.1 is that this projection can reduce the term of $\mathcal{B}_{g_\epsilon}(f)$. For example, consider $z + v$ where z is an embedding and v is some perturbation of z. When computing a projection with a random matrix R by $R(z + v)$ and v is in the null space of R, then $R(z + v) = Rz$ and hence this perturbation has no effect. Since R is for down-sampling, there is a large null space for R. In other words, if the distribution shift results in v such that Rv is small, $\mathcal{B}_{g_\epsilon}(f)$ is small.

F.3 Implementation Details.

Below we summarize all relevant implementation details.*

Model Architecture. Our experiments build upon the following model architecture.

Encoder: We use 4 publicly available backbones:

1. ResNet50 backbone, pre-trained supervised on ImageNet [25, 397]: We used the weights from PyTorchs `torchvision.models` package, specifically the weights under identifier `ResNet50_Weights.IMAGENET1K_V2`†. The backbone yields an embedding dimension of $m_z = 2048$.
2. ViT-B/32 backbone, CLIP-pre-trained [242]: We used the ViT B/32 model that can be easily obtained via OpenAI's official github repository `clip` python package ‡. The backbone yields an embedding dimension of $m_z = 512$.
3. ResNet50w2 with twice the standard wideness, SwAV self-supervised pre-trained [240]: We use the official pytorch model that can be loaded with PyTorch using the command `torch.hub.load('facebookresearch/swav:main', 'resnet50w2')`. The backbone yields an embedding dimension of $m_z = 2048$.
4. ResNet50, DINO self-supervised pre-trained [233]: We use the official pytorch model that can be loaded using `torch.hub.load('facebookresearch/swav:main', 'resnet50w2')`. The backbone yields an embedding dimension of $m_z = 1024$.

* https://github.com/ftraeuble/experiments_discrete_key_value_bottleneck.
† https://pytorch.org/vision/stable/models.html
‡ https://github.com/openai/CLIP

5. As a fifth model, we present results using a ConvMixer pre-trained on a downsampled (32x32) Imagenet training set. As we wanted to directly compare against SDMLPs performance on the same class-incremental learning setting, we use their ConvMixer embeddings evaluated on the entire Imagenet and CIFAR10 dataset, which can be downloaded from the associated github repository.[§] The backbone yields an embedding dimension of $m_z = 256$.

To cover a broad range of feature embeddings, we extracted the spatial continuous representation after the second last layer of residual blocks in the case of the two self-supervised pre-trained backbones and applied adaptive average pooling. The inferred continuous-valued representations cover a broad range of embedding dimensionalities depending on the backbone. In all our experiments we obtained the C embedding heads for each key-value codebook by simply down-projecting this vector using C random projection matrices into C heads of dimension d_{key}.

Key-Value Bottleneck: The key-value bottleneck consists of 256 key-value codebooks that have each 4096 key-value pairs per codebook. Keys are of the same dimension as the embedding heads (d_{key}), and we chose the value codes to be of the same size as the classes to predict, i.e. $d_{value} = 10$.

Decoder: All C fetched value codes are element-wise average-pooled at the end and the resulting tensor is passed to a softmax layer that predicts the class label. For simplicity of exposition, we assume that the encoder representation does not have spatial dimensions; nevertheless, the extension to spatial feature maps is straightforward.

Initialization of Keys. To initialize the keys, we build upon the exponential moving average updates (EMA) implemented in the pip package `vector-quantize-pytorch`[¶] as introduced in [237, 238]. For a given sequence of $t = 1, ..., T$ mini-batches, we compute the moving averages of the codes (=key) positions $k_i^{(t)}$ and counts $N_i^{(t)}$ as follows for the c-th codebook

$$N_i^{(t)} := \gamma N_i^{(t-1)} + n_i^{(t)}(1-\gamma), \tag{F.7}$$

$$m_i^{(t)} := \gamma m_i^{(t-1)} + \sum_j^{n_i^{(t)}} E^c(x)_{i,j}^{(t)}(1-\gamma), \tag{F.8}$$

$$k_i^{(t)} := \frac{m_i^{(t)}}{N_i^{(t)}} \tag{F.9}$$

where $E^c(x)^{(t)}_{i,j=1,...,n_i^{(t)}}$ are the $n_i^{(t)}$ (head) embeddings of the set of observations in the mini-batch that attach to the i-th key. We use a decay factor of $\gamma = 0.95$. Moreover, we expire codes if they are not being adequately utilized. If the cluster size corresponding to a key (i.e., the moving average of the number of encodings that attach to a key) is below a threshold, we reinitialize the said key to a randomly selected encoding. This threshold scales as: $0.1 \cdot \text{batch-size} \cdot h \cdot w / \text{num-pairs}$ with h and w the spatial feature map dimensions in case of spatial pooling after the bottleneck. Keys are initialized at random encoder feature positions from the first batch. Our experiments apply spatial pooling before the bottleneck such that $h = w = 1$.

[§] `https://github.com/anon8371/AnonPaper1`

[¶] `https://github.com/lucidrains/vector-quantize-pytorch`

Training. After initializing the keys for 10 epochs on the key-initialization dataset (1 epoch on the Imagenet datasets), we train the model on the class-incremental CIFAR10 task with the SGD PyTorch optimizer without any weight decay or momentum and a learning rate lr = 0.3 for the bottleneck and lr = 0.001 for the linear probe. We used a label smoothing parameter of 0.1 though we did not find its value to be sensitive to the result. We use no learning rate schedule. We used a batch size of 256 during key initialization and continual learning.

F.4 Additional Experiments

F.4.1 Granularity during class-incremental learning.

In addition to the class-incremental learning scenario discussed in the main text, we experimented with additional class-incremental curricula of different split granularities. Specifically, we also tested the same model's performance (with the ConvMixer backbone) not only on 5 phases with 2 classes each but also on the more extreme case of 10 phases with a single class each and two phases with 5 classes each. We show these results in Figure F.1, further supporting our finding from the main experiment that the proposed model is robust against various class-incremental curricula and approaches the same performance at the end of training.

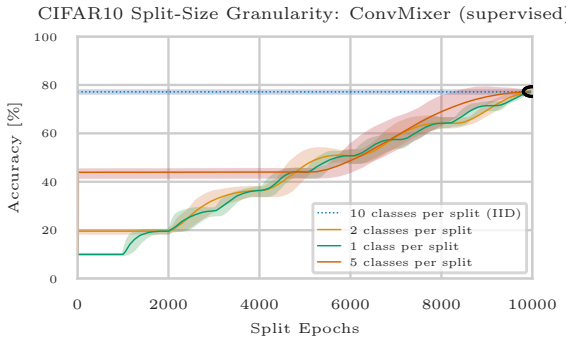


Figure F.1: Analysing the effect of class-granularity during class incremental learning. In line with our theory, we observe *across all possible class-split sizes* the appealing learning property that the bottleneck yields the same performance at the end of training.

F.4.2 Additional experiments on ConvMixer backbone.

The authors in [271] report two further methods with higher performance but their "SDMLP+EWC" results does not apply as it requires task boundaries and the FlyModel from Shen, Dasgupta, and Navlakha [274] was not trained on the batched 2000 epochs streaming setup. However, we also attempted to implement the FlyModel that proposes a solution for continual learning inspired by the neural circuit in the fruit fly olfactory system. Similar to us, the FlyModel also relies on a frozen backbone but then projects this embedding via sparse binary projections (which are likewise fixed) into an extremely high-dimensional space reflecting more than 10,000 so-called "Kenyon cells". It then instantiates num-class "MBON cells" that are associative layers, fed from this high-dimensional

To the best of our knowledge, all software and assets we use are open source and under MIT, Apache or Creative Commons Licenses.

representation. The FlyModel is trained such that learning a particular class updates the "MBON" associative layer of this class and was primarily tested under less challenging task-incremental settings by the authors. As mentioned before, the model reported in [271] was not trained for 2000 epochs per split. Instead, it was only updated over a single epoch simply assuming no further catastrophic forgetting in the batched streaming setting we are investigating (the FlyModel is not trained using back-propagation). In order to make the comparison as fair as possible, we trained the FlyModel for 2000 epochs per split building upon the implementation in [271]. Following Bricken et al. [271], we applied the following parameters that were reported to work best for them on this task and this backbone, specifically a learning rate of 0.005, 10,000 Kenyon cells and the number of projection neuron connections to be 32. We also set $\alpha = 0$ to minimize catastrophic forgetting as described by Shen, Dasgupta, and Navlakha [274]. After rerunning this model for 5 random seeds each we obtained the results presented in Figure F.2. As we can see, the model learns new classes at the beginning of a new split but quickly drops to some lower-level performance until the end of the phase. Eventually, the model cannot integrate any new information and remains at a very low level, which indicates strong catastrophic forgetting in the batched streaming setting.

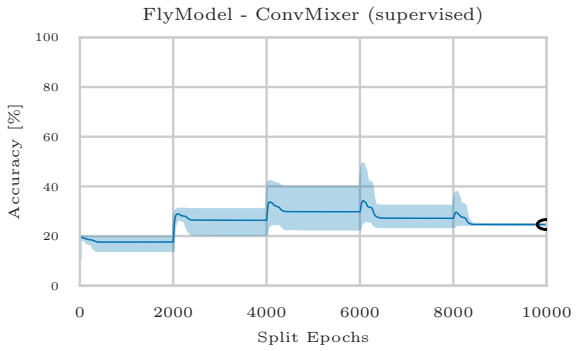


Figure F.2: Applying the FlyModel from [274] to the CIFAR10 learning setup with each split being repeated 2000 epochs. The model with hyperparameters from [271] learns new classes at the beginning of a new split but quickly drops to some lower-level performance until the end of the phase. Eventually, the model cannot integrate any new information and remains at a very low level.

F.4.3 Additional baseline: Ridge regression on frozen backbone features

We have also tested a further baseline method by directly performing ridge regression with the frozen feature encoder. Specifically, we were rerunning the CIFAR10 experiment on all five backbones using two different heads (Linear Probe and as well as the MLP head from Figure 6.3). In contrast to these baselines already shown in the main paper, we now applied weight decay on the head parameters across four orders of magnitude with $w_d \in [1e-2, 1e-4, 1e-6]$. We repeated three runs with different random seeds each. In total, we trained 90 additional models for this experiment. As illustrated in Figure F.4 in the case of the Linear Probe and Figure F.5 in the case of the MLP head, the learned networks were unable to effectively integrate novel class information using this additional form of ridge regularization.

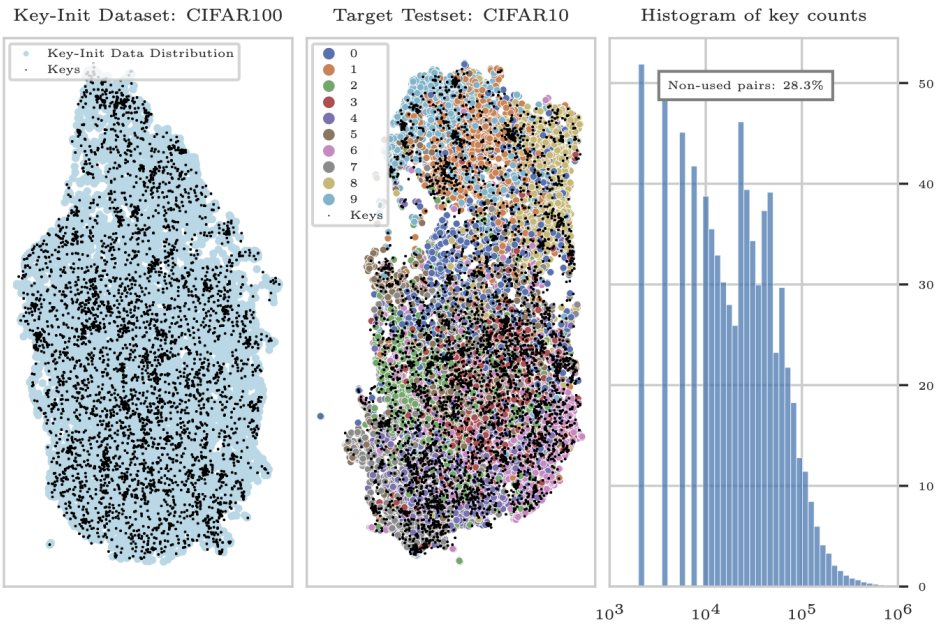


Figure F.3: Analysing the **key codes** on the **ResNet50 backbone.** Left Plot: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle Plot. The key code distribution on the target test dataset. Keys exhibit increased clustering, which reflects the smaller feature diversity present in CIFAR10. Right Plot: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many different input samples, while the bottleneck retains some unused capacity.

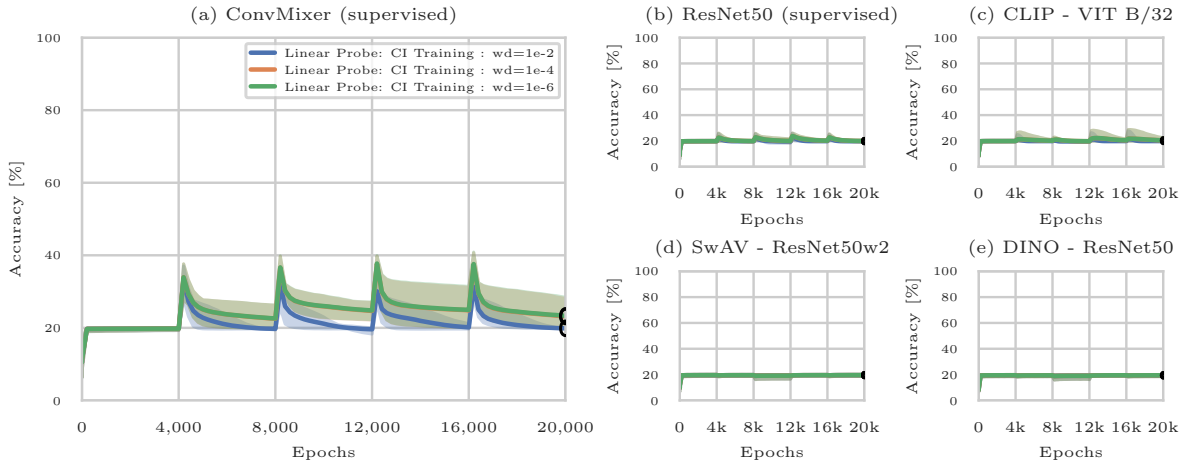


Figure F.4: Additional baseline experiments: Rerunning the CIFAR10 experiment on all five backbones using the same Linear Probe as in Figure 6.3. In contrast to these baselines already shown in the main paper, we now applied weight decay on the head parameters across four orders of magnitude with $w_d \in [10^{-2}, 10^{-4}, 10^{-6}]$. The learned models are unable to effectively integrate novel class information using this additional form of ridge regularization.

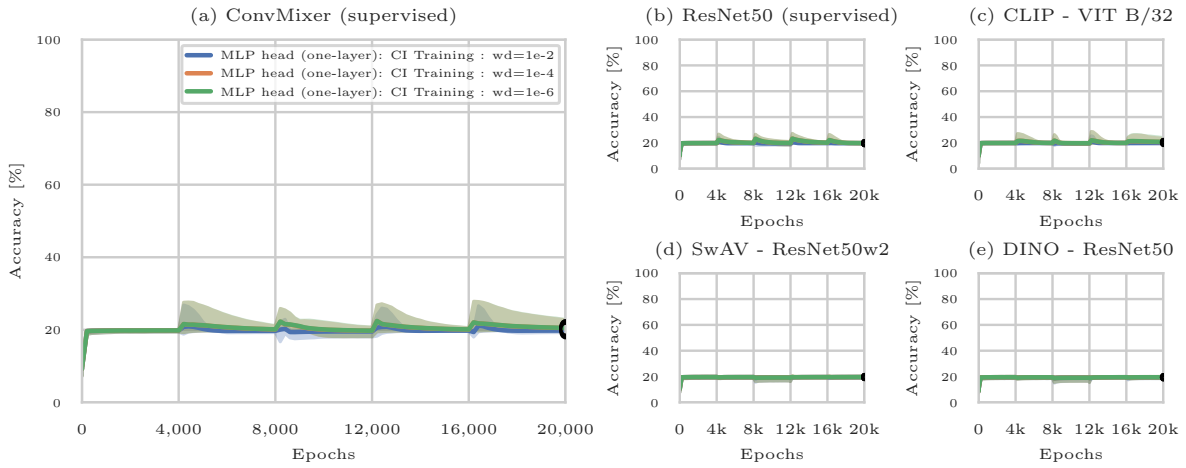


Figure F.5: Additional baseline experiments: Rerunning the CIFAR10 experiment on all five backbones using the same MLP head as in Figure 6.3. In contrast to these baselines already shown in the main paper, we now applied weight decay on the head parameters across four orders of magnitude with $w_d \in [10^{-2}, 10^{-4}, 10^{-6}]$. The learned models are unable to effectively integrate novel class information using this additional form of ridge regularization.

F.4.4 Class-incremental learning on larger datasets.

To further test the future scalability of our method, we have run additional experiments on two larger datasets with many more classes. Importantly, we were using the very same architecture as for the CIFAR10 experiments:

1. CIFAR-100 (see Figure F.6): Using 50 splits, each consisting of 2 classes, we tested our model on the ResNet-50 and CLIP-Backbone architectures. Our model achieved final accuracies of 54.7% and 64% respectively, compared to the 43% achieved by SDMLP on the same task [271].
2. ImageNet (see Figure F.7): We evaluated our model on 500 splits, each comprising 2 classes, using the CLIP encoder. The final accuracy reached 49.9% for the i.i.d. setting and 49.0% for the class-incremental setting. To the best of our knowledge, no other method has been tested on such a large-scale class-incremental learning setting, encompassing 1000 classes and 500 splits.

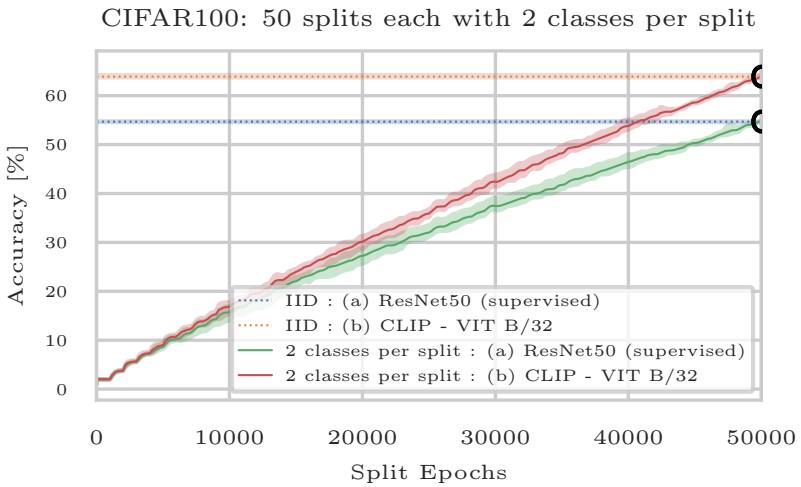


Figure F.6: Additional class-incremental learning experiment on CIFAR100: Using 50 splits, each consisting of 2 classes, we tested our model on the ResNet-50 and CLIP-Backbone architectures. Our model achieved final accuracies of 54.7% and 64% respectively, compared to the 43% achieved by SDMLP on the same task [271]. Keys were initialized on the CIFAR10 dataset for this experiment.

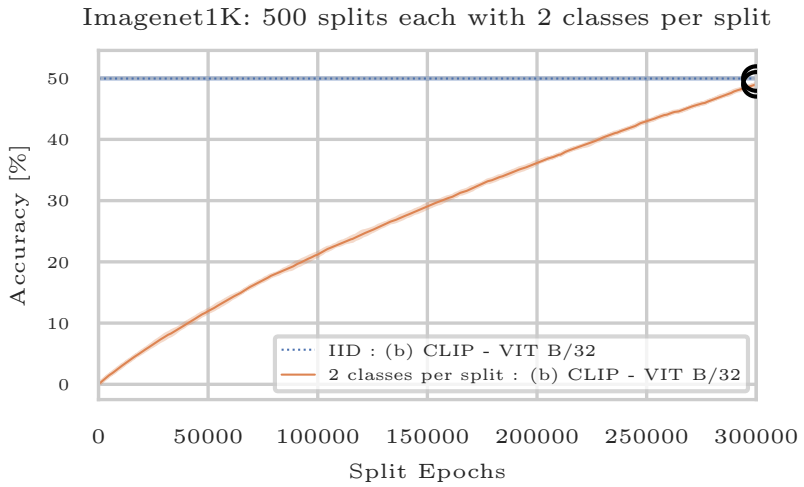


Figure F.7: Additional class-incremental learning experiment on ImageNet: We evaluated our model on 500 splits, each comprising 2 classes, using the CLIP encoder. The final accuracy reached 49.9% for the i.i.d. setting and 49.0% for the class-incremental setting. To the best of our knowledge, no other method has been tested on such a large-scale class-incremental learning setting, encompassing 1000 classes and 500 splits. Keys were initialized on the CIFAR100 dataset for this experiment.

F.5 Additional Analysis

F.5.1 Robustness via discrete bottleneck.

The discretization of the bottleneck confers additional robustness, which is a significant attribute of our model. To illustrate this property, we have conducted additional experiments to assess the models' robustness. Specifically, we evaluated the performance of the models analyzed in Figure F.3, Figure F.9, Figure F.10 and Figure F.11, which employ different backbones, when subjected to additive Gaussian noise spanning three orders of magnitude. We applied the noise either to the original input pixels or the backbone embeddings. Our results which are shown in Figure F.8, indicate that the model's performance remains largely unaffected by the noise and only starts to deteriorate substantially when the noise level exceeds $\alpha = 0.1$, at which point the image becomes nearly unrecognizable upon visual inspection. From a theoretical perspective, the term $\mathcal{B}_g$ in Theorem 1 is capturing the effect of input noise on the encoding process across the entire training dataset. If we now let f_k be the restriction of the function f to C_k for some C_k that is defined in Section 6.3, then the Lipschitz constant of f_k is zero for all k. This means that the Lipschitz constant is zero locally in each domain C_k.

F.5.2 Value Codes

Finally, we want to provide additional analysis on the similarity of the learned values when training i.i.d. versus class-incrementally. To answer how much the value codes differ depending on the training order (everything else being the same), we computed the relative mean absolute difference between values learned i.i.d. and values learned class-incrementally in the case of the five analyzed

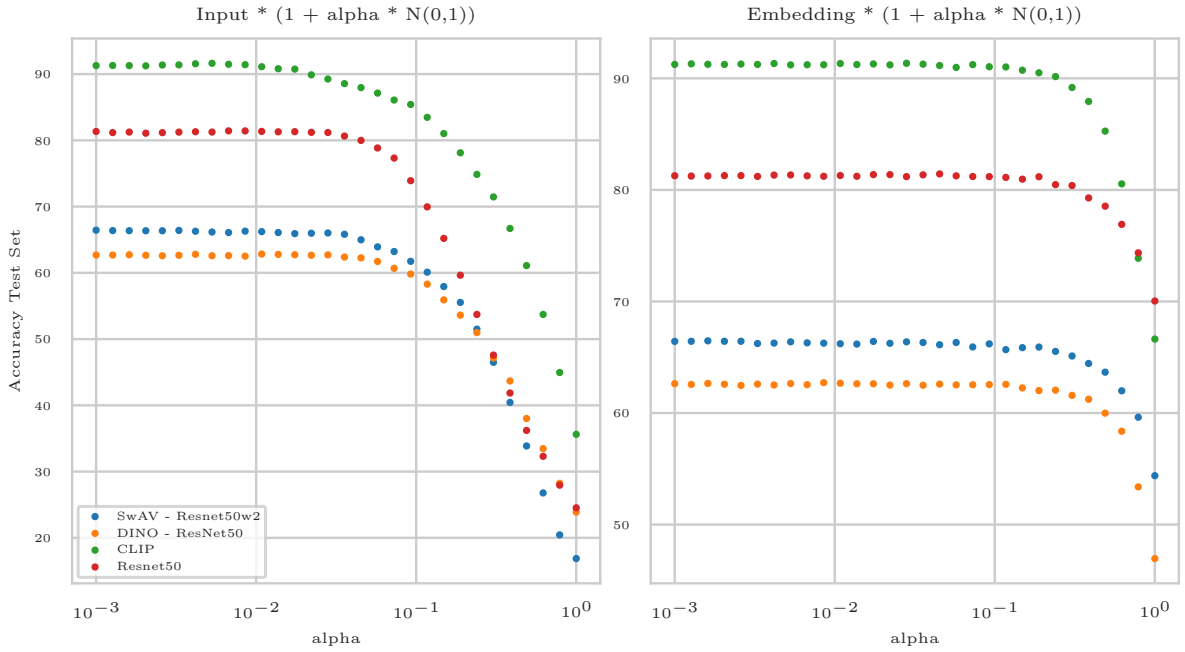


Figure F.8: We have conducted additional experiments to assess the models' robustness. Specifically, we evaluated the performance of the models analyzed in Figures 8-11, which employ different backbones, when subjected to additive Gaussian noise spanning three orders of magnitude. We applied the noise either to the original input pixels (left) or the backbone embeddings (right). Results indicate that the model's performance remains largely unaffected by the noise and only starts to deteriorate substantially when the noise level exceeds $\alpha = 0.1$, at which point the image becomes nearly unrecognizable upon visual inspection. From a theoretical per

models in the previous section, formally

$$\frac{\sum |values_{CI} - values_{IID}|}{\sum |values_{CI}|}. \tag{F.10}$$

As we can see from Table F.1, the analysis suggests that the learned values are indeed fairly similar, albeit not identical, explaining the almost identical final performance.

F.5.3 Key Codes

In this section, we analyze key codes used in the proposed model to better understand how they cover the embedding manifold of the data. We investigate three questions:

1. Do key codes broadly cover the embedding manifold with respect to the key initialization dataset?
2. How does key initialization on another dataset affect coverage on the embedding manifold under the target dataset?
3. Can we quantify the amount of key-value pair sharing among training samples?

To answer these questions, we inspect the key codes for each backbone discussed in Section 6.5, analyzing the model per backbone with the same fixed random seed trained under the class-incremental training setting. Results are shown in Figures F.3 and F.9 to F.12.

We begin by inspecting the key code space of the first codebook under the supervised ResNet50 backbone in Figure F.3 (left plot). The keys of this model were initialized on the unlabelled CIFAR100

Table F.1: Relative mean absolute distance between values trained i.i.d. and class-incremental for the five different backbones from the base experiment.

Backbone	Rel. mean abs. distance
Resnet50	2.19%
CLIP	2.09%
DINO	2.2%
SwAV	2.2%
ConvMixer	3.07%

dataset. When applying the UMAP dimensionality reduction on the input heads of 10k CIFAR100 training samples, we can see that the 4096 key codes of this codebook (black dots) broadly cover the entire data manifold (light blue dots). This verifies that the EMA method is working as expected. When repeating the same procedure for the test set of the target distribution, here CIFAR10, we observe that the fixed keys from before are still covering the entire data domain of CIFAR10, see Figure F.3 (middle plot). Compared to the distribution on the key initialization dataset, the coverage is more sparse with several concentrations of multiple clusters of keys. We hypothesize that these keys might have specialized on certain features present in CIFAR100 but not present in CIFAR10. At the same time, we observe keys across all CIFAR10 class domains (indicated by color) even though the same class labels are not present in CIFAR100. We observe this behaviour across all backbones: Interestingly, we qualitatively observe a slightly reduced clustering in the case of the lower-level SSL feature embeddings in Figure F.10 and Figure F.11 over the others. In our ablation experiments, we only observed a slight drop in performance when initializing the keys on the much higher-resolution ImageNet dataset, which can be partially explained with the analysis under the ConvMixer model, that was initialized on a 32x32 downsampled Imagenet variant (see Figure F.12). First, the EMA approach still yields a broad coverage of the much more diverse Imagenet data manifold without any change in EMA parameters. Second, the keys are covering the CIFAR10 domain sparsely but there are still at least a few keys present close to each domain. From our discussed results we can thereby conclude that the bottleneck can still effectively learn to generalize along all classes even though the keys might not be optimally distributed. Finally, we tracked the usage of all key-value pairs across all codebooks over the course of the training class incrementally on the target dataset; see Figure F.3 (right plot). Note that the y-axis is scaling in counts of thousands. In the case of the ResNet50 backbone, we observe that the majority of key-value pairs were fetched between 10,000 and 150,000 times over the course of training. As we repeat each episode for 2000 epochs, each key that is snapped to by a particular training image contributes a utilization count of 2000 to this key. Interestingly, there is also a long tail of a few keys that are used very often and about 28% of keys that are never used, which suggests that the bottleneck did not distil the information during train time in all available key-value codes. This behaviour can be seen similarly among all other backbones, too.

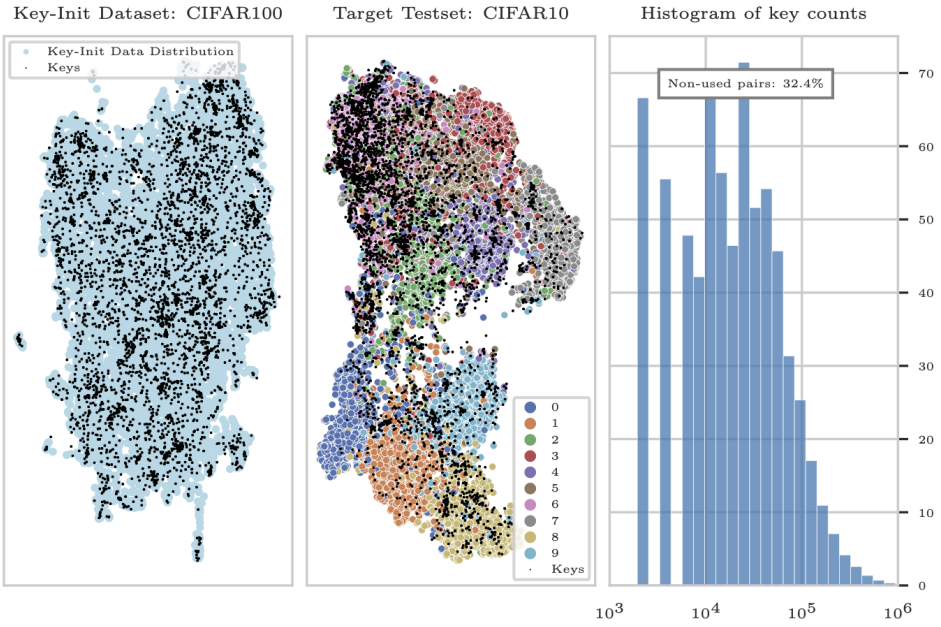


Figure F.9: Analysing the **key codes** on the **CLIP-pre-trained ViT backbone**. Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Although the coverage appears more clustered, the keys still cover the majority of the target test set, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many different input samples, while the bottleneck retains some unused capacity.

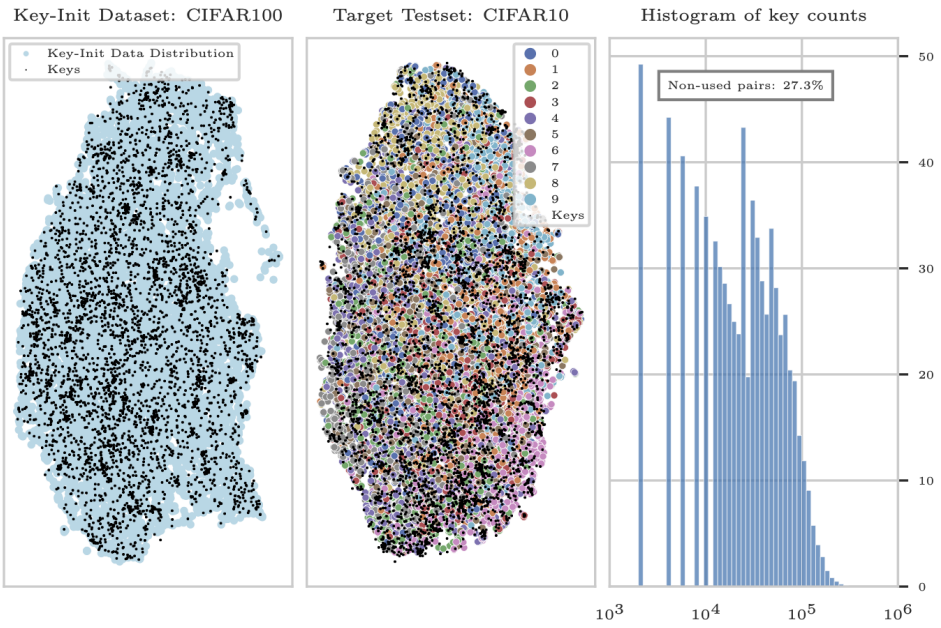


Figure F.10: Analysing the **key codes** on the **DINO-pre-trained backbone**. Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Although the coverage appears more clustered, the keys still cover the majority of the target test set, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many different input samples, while the bottleneck retains some unused capacity.

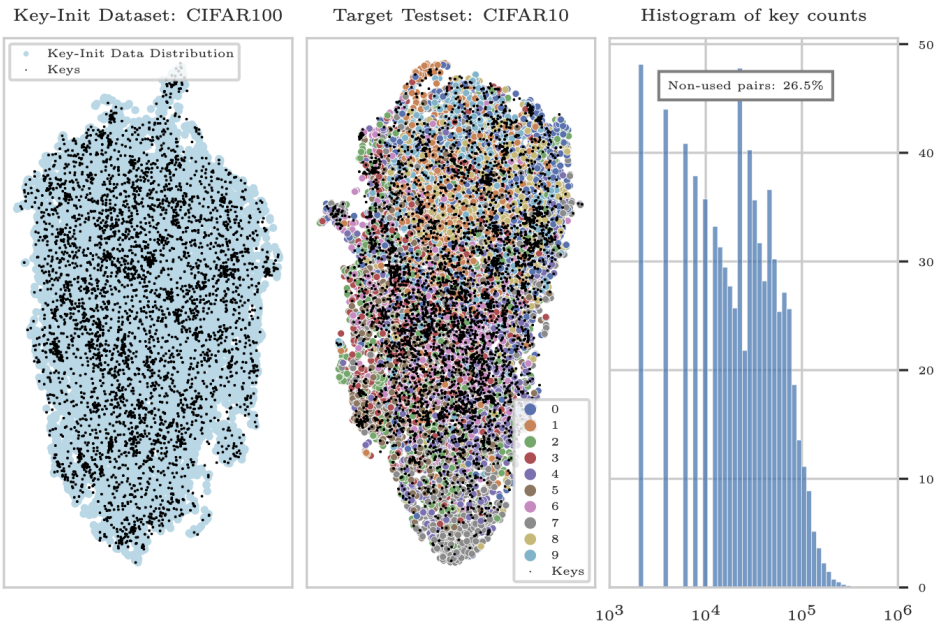


Figure F.11: Analysing the **key codes** on the **SwAV-pre-trained backbone.** Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Keys exhibit increased clustering, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many input samples, while the bottleneck retains some unused capacity.

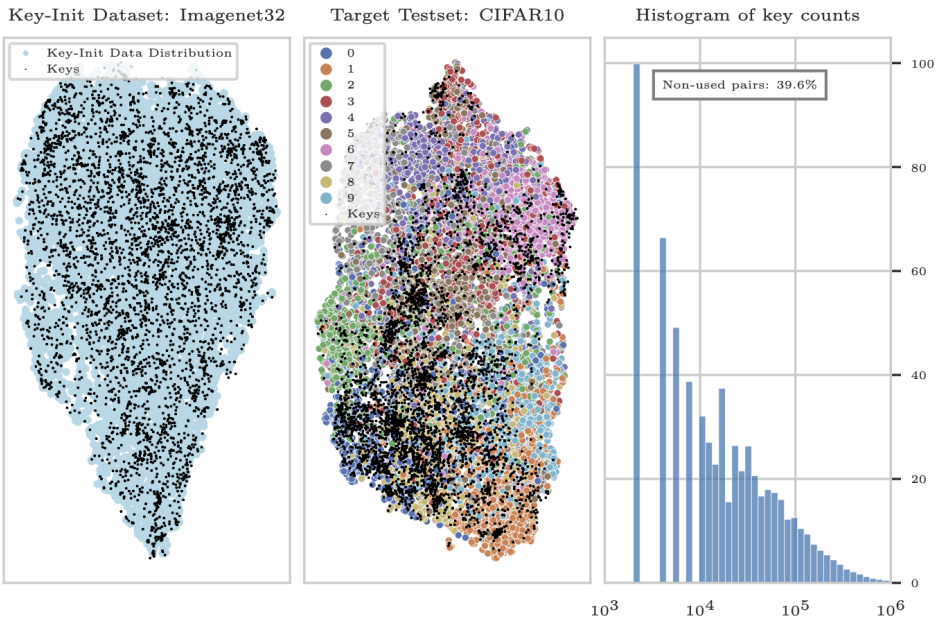


Figure F.12: Analysing the **key codes** on the **ConvMixer backbone** from [271]. Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Keys exhibit increased clustering, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many input samples, while the bottleneck retains some unused capacity.

Appendix to Chapter 7 G

G.1 Assumption of Conditionally-Independent Classifiers

In the case of two given classifiers our assumption implies that $\forall n, \forall (t, t') : I(\hat{Y}_n^t; \hat{Y}_n^{t'} | Y_n) = 0$. To quantify deviations of Assumption 1, one could therefore measure this conditional mutual information, but note that this in itself is challenging on finite data and would again inherit model assumptions through the choice of the estimator. Since pairwise independence does not imply mutual independence, even having these quantities might not be sufficient for a long classifier sequence. An alternative approach could thus be to measure the Maximum Mean Discrepancy (MMD) between the LHS and RHS of Assumption 7.3.1, which can be viewed as a conditional version of the d-dimensional Hilbert Schmidt Independence Criterion (dHSIC), see [398].in

Importantly, all of the above approaches aim to quantify and subsequently incorporate correlations between classifier predictions, and thus rely on having large amounts of labelled data to properly estimate them which is almost never the case in practice. Since our goal was to devise an approach that can be useful in practice, when the number of classes, observations, and models is typically very large, we believe that the model choice behind Assumption 7.3.1 is reasonable; it enables both tractable inference and parameter estimation.

G.2 Backward Compatibility Scores

For the reported BTC and BEC scores we follow the definition proposed in [304] with h_1 and h_2 being the predictors at time step 1 and time step 2. Translated to our setup, we associate h_1 with our estimated and stored prediction at time step 1 whereas h_2 corresponds to the updated prediction dataset at time step 2:

Backward Trust Compatibility (BTC) score. "The ratio of points in a held-out test set (e.g., D_{test}) that predicted correctly among all points h_1 had already predicted correctly." [304]

$$\text{BTC} = \frac{\sum_{i=1}^{|D|} \mathbf{1}[h_1(x_i) = y_i, h_2(x_i) = y_i]}{\sum_{i=1}^{|D|} \mathbf{1}[h_1(x_i) = y_i]} \tag{G.1}$$

Backward Error Compatibility (BEC) score. "The proportion of points in a held-out test set that h_2 predicted incorrectly, out of which h_1 also predicted incorrectly, thus capturing the probability that a mistake made by h_2 is not new." [304]

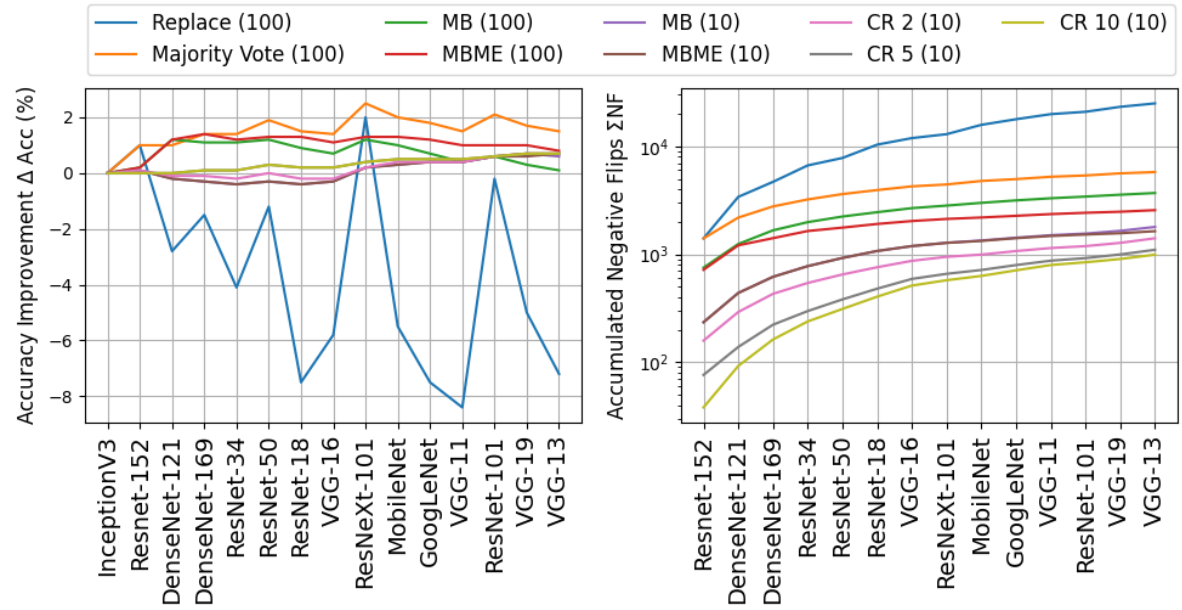
$$\text{BEC} = \frac{\sum_{i=1}^{|D|} \mathbf{1}[h_1(x_i) \neq y_i, h_2(x_i) \neq y_i]}{\sum_{i=1}^{|D|} \mathbf{1}[h_2(x_i) \neq y_i]} \tag{G.2}$$

G.3 Robustness to Random and Adversarial Model Sequences

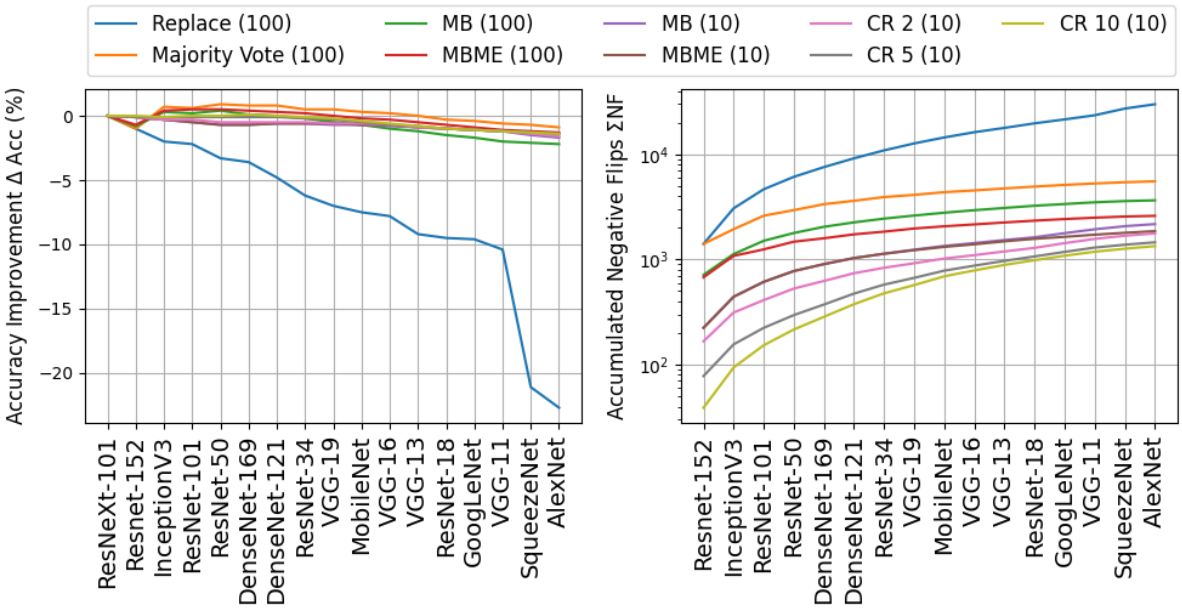
As already eluded to in Section 7.6.4, in our main experiments we have assumed that the models C^t are *improving* over time. In general, this assumption may be justified by having observed superior performance of a new model for the source domain on which they were trained, prior to deciding to use it for re-evaluation on the target data set. However, such information may not always be available.

To check for robustness against violations of the assumption of improving classifiers, we also consider scenarios where C^t arrive in a *random* or *adversarial* (i.e., strictly deteriorating) order with respect to their accuracy. To avoid unrealistically large fluctuations in the accuracy of new classifiers, we removed AlexNet and SqueezeNet as the most poorly performing models from the *random* model sequence for this set of experiments. Results on ImageNet in the form of the temporal evolution of accuracy improvement and accumulated negative flips across different strategies and budgets are shown in Figure G.1.

For the random ordering in Figure G.1 (a), we find that our methods—unlike, e.g., the replace strategy—achieve strict increases in *final* accuracy while introducing much fewer negative flips, e.g., almost an order of magnitude fewer for our CR strategies, compared to the Majority Vote baseline. Even in the adversarial case in Figure G.1 (b), our methods improve accuracy during the initial steps and introduce much fewer negative flips over the entire history. These findings suggest robustness of our approach to situations where an ordering by performance of C^t may not be available.



(a) Random order



(b) Adversarial order

Figure G.1: Temporal evolution of accuracy improvement and accumulated negative flips on ImageNet for two scenarios where the models C^t do not exhibit improving performance, but instead arrive in (a) random or (b) adversarial order. As can be seen, our methods are robust in both these cases while introducing much fewer negative flips than the non-probabilistic baselines.

G.4 Role of Confusion Matrix Estimates

For the results of our main experiments reported in Section 7.6.2 and Section 7.6.3, we estimated only the diagonal elements of the confusion matrices and set the off-diagonal elements to a constant, c.f. Section 7.3.1. To investigate the role of the estimation procedure for the confusion matrices π^t, we also report results for the case where we estimate the *full* confusion matrix of each classifier (i.e., including the off-diagonal elements). To avoid off-diagonal elements which are estimated to be zero due to the large number of classes and the limited size of the validation set split, we use a Laplace-smoothed version of the maximum likelihood estimate, i.e., we add a one to each count and normalise accordingly, as suggested in Section 7.4.

Results of this comparison between different confusion matrix estimators for ImageNet and ObjectNet are shown in Figure G.2 as well as the detailed evolution of key metrics in Figure G.3 and Figure G.4. (The corresponding results for CIFAR-10 are shown in Figure G.6 and are discussed in more detail in G.5.) As is apparent from the comparison on ImageNet and ObjectNet in Figure G.2, we indeed observe generally improved performance from estimating full confusion matrices, as speculated in Section 7.4.

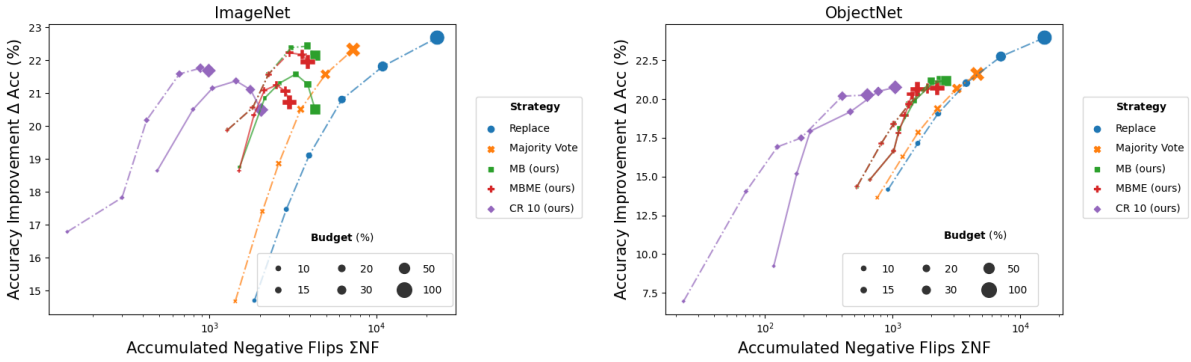
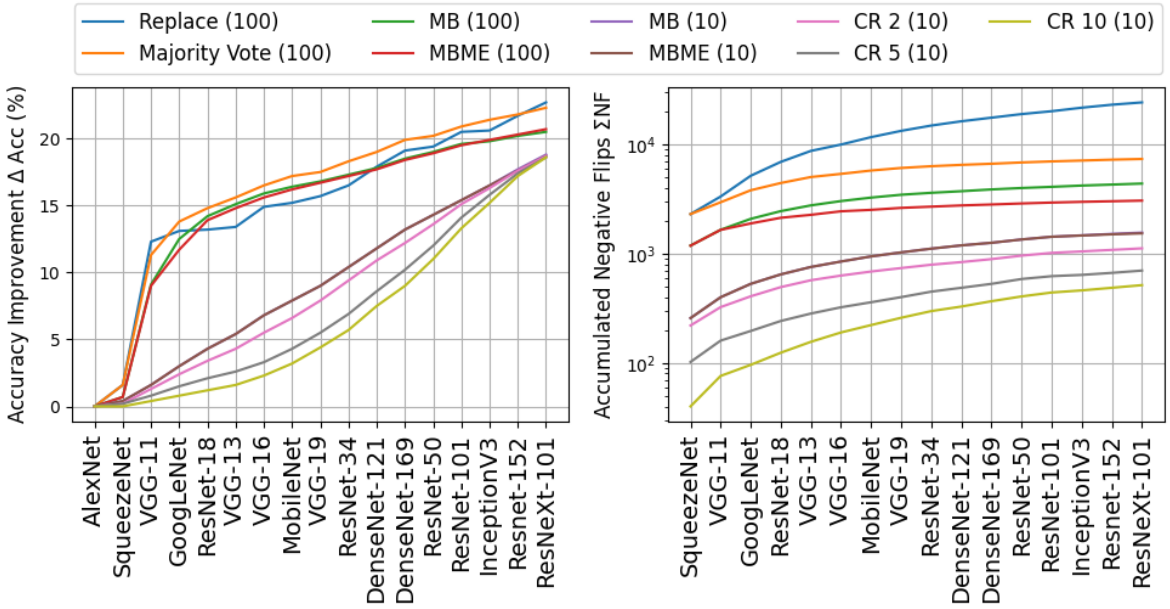


Figure G.2: Investigation into the role of different confusion matrix estimates on ImageNet (left) and ObjectNet (right). Dashed lines in the scatter represent results for when the full confusion matrices (i.e., including the off-diagonals) are estimated from smoothed counts on a split of the validation set, and solid lines represent results under the less accurate confusion matrix estimates where only diagonal elements (i.e., the class-specific accuracies) are estimated, as in the experiments of the main paper.

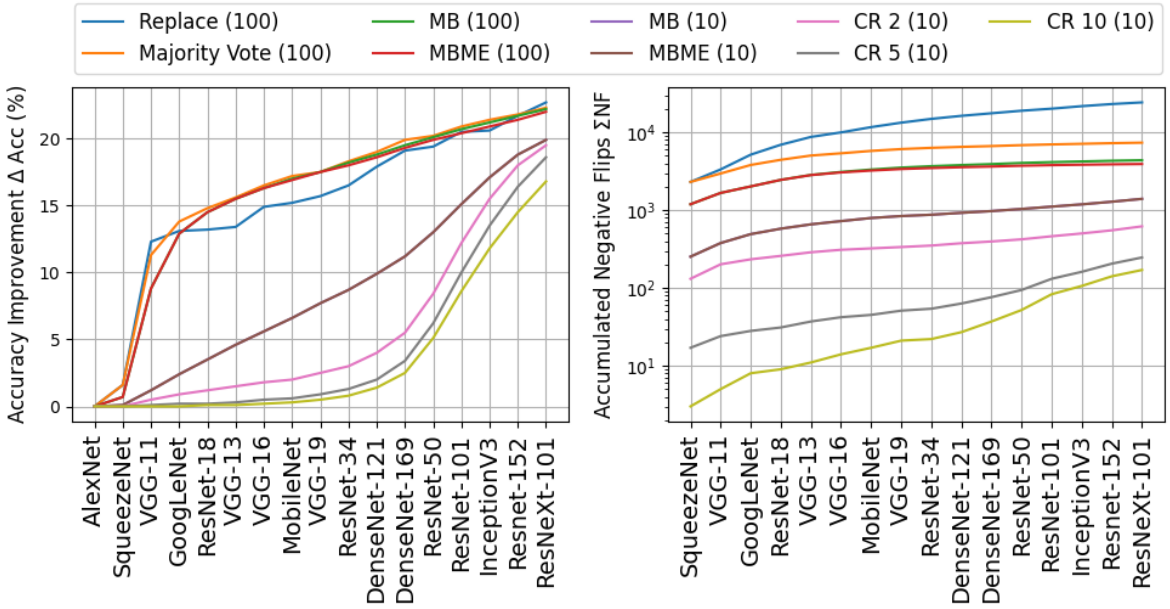
In terms of final accuracy, we observe a different behaviour across different budgets. For large budgets, estimating full confusion matrices leads to substantial accuracy gains on ImageNet and to roughly equal or slightly improved accuracy on ObjectNet. For small budgets, on the other hand, we find similar or smaller accuracy gains for full confusion matrix estimates; importantly, this effect is most pronounced for the more conservative CR prediction-update strategies. In terms of negative flips, we observe slight to major reductions across all budgets and strategies with full confusion matrix estimates compared to only estimating the diagonals, i.e., the class-specific accuracies.

We believe that this behaviour is intuitive and can be explained as follows. Estimating the full confusion matrices with smoothed counts generally yields smaller diagonal elements and larger off-diagonal elements. This, in turn, means that our posterior beliefs (in particular, the ratio between two consecutive MAP values) change less drastically after new re-evaluations, so that the more conservative CR strategies update fewer labels and thus experience smaller and slower accuracy gains when only few samples can be re-evaluated. At the same time, the resulting posterior estimates are likely more accurate which is consistent with larger gains for large budgets and the

reduced number of negative flips. In summary, better confusion matrix estimates result in a further reduction in the number of negative flips across the board. For small budgets and conservative prediction-update strategies, however, this comes with a trade-off in overall accuracy gain.

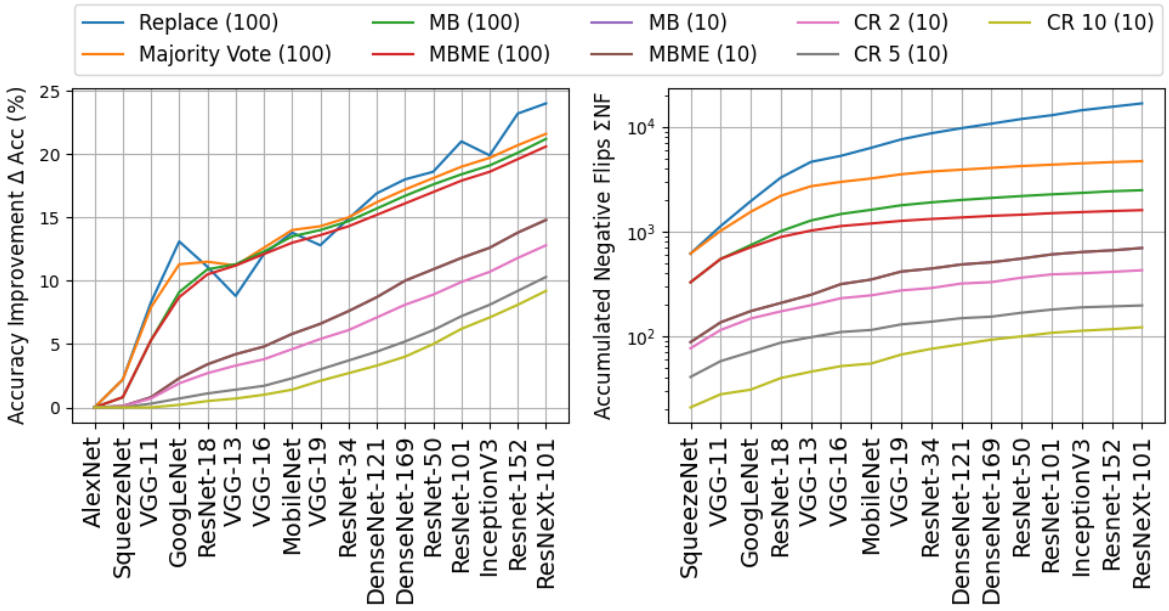


(a) ImageNet: Diagonal Confusion Matrix

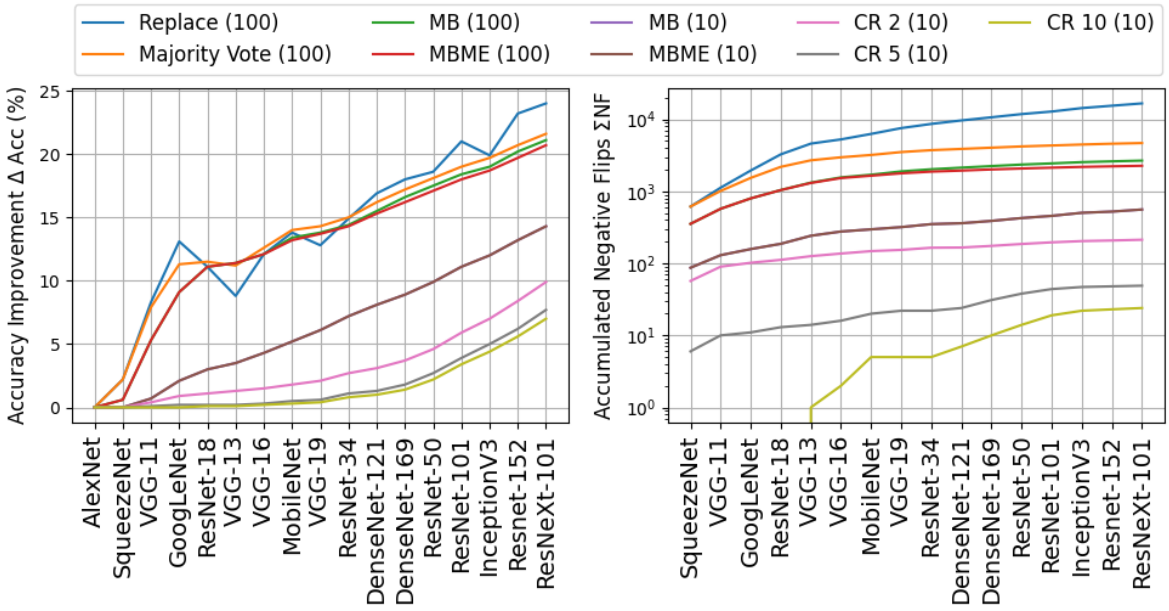


(b) ImageNet: Full Confusion Matrix

Figure G.3: Investigation into the role of different confusion matrix estimates on the evolution of key metrics on ImageNet. Only Diagonal Confusion Matrix Estimation (top) and Full Confusion Matrix Estimation (bottom).



(a) ObjectNet: Diagonal Confusion Matrix



(b) ObjectNet: Full Confusion Matrix

Figure G.4: Investigation into the role of different confusion matrix estimates on the evolution of key metrics on ObjectNet. Only Diagonal Confusion Matrix Estimation (top) and Full Confusion Matrix Estimation (bottom).

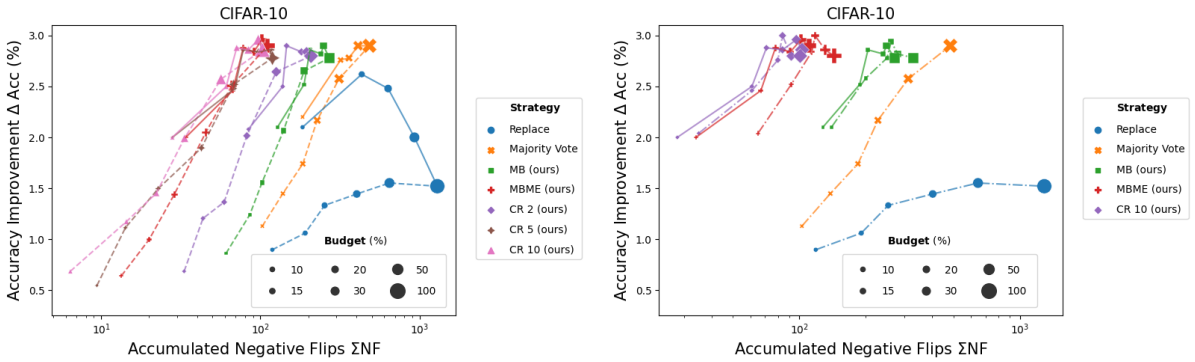


Figure G.5: Summary of prediction-updates on CIFAR-10. Left: Dashed lines correspond to the ablation using a random selection strategy. Right: Similar to Figure G.2, dashed lines represent results with full confusion matrices estimated from smoothed counts and solid lines represent using a less accurate confusion matrix estimate where only diagonals are estimated from data. We observe very similar behaviour to that described for ImageNet and ObjectNet in Section 7.6.2 and Section 7.6.3, respectively. The most interesting difference compared to our experiments on ImageNet and ObjectNet, is that, on CIFAR-10, our methods achieve substantially larger accuracy gains compared with the baselines, even for small budgets.

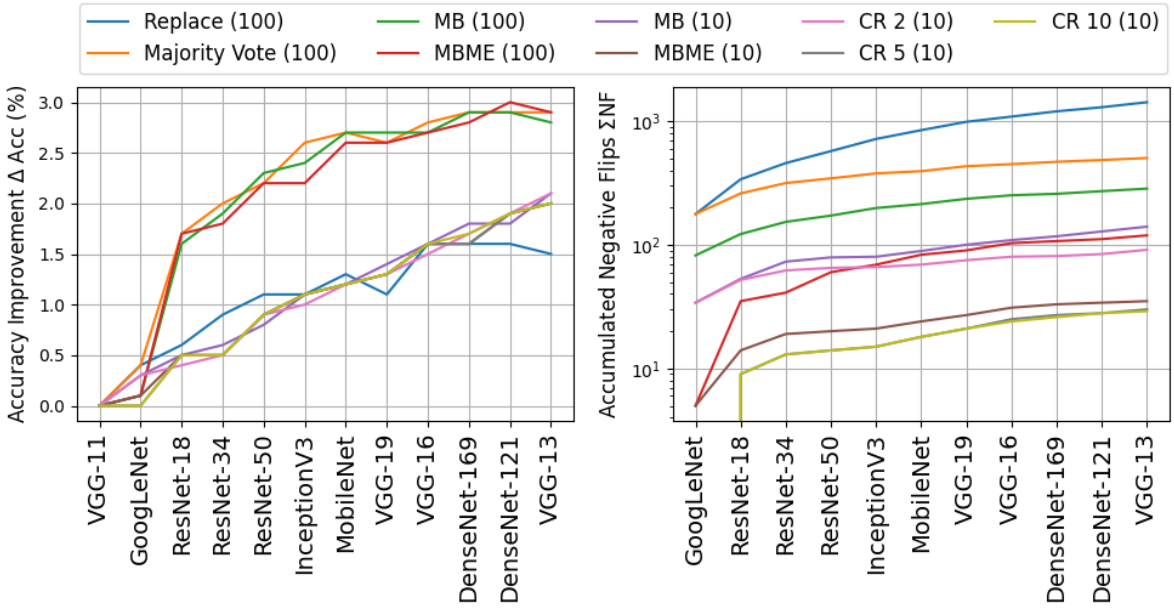
G.5 Experiments on CIFAR-10

CIFAR-10 is an image classification dataset comprising images of objects from 10 distinct classes (airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, trucks). It contains a training set of 50,000 images and a test set of size 10,000. It is thus a rather simple classification dataset compared to ImageNet and ObjectNet but has been a key driver for advancing ML models and computer vision before the introduction of ImageNet. Using the CIFAR-10 test set as our target dataset, we can study the Prediction Updates Problem in an i.i.d. setting with a small number of classes.

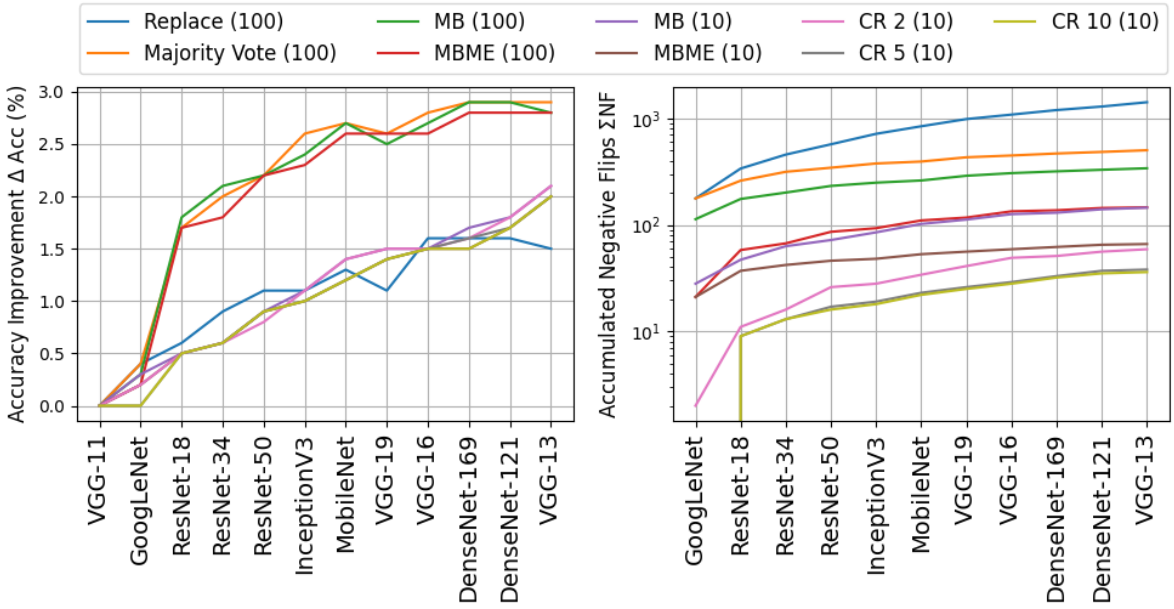
Similar to our experiments on ImageNet and ObjectNet reported in Section 7.6 where we used models which had been pre-trained on ImageNet,* for our experiments on CIFAR-10, we use models which have instead been pre-trained on CIFAR-10, available from an open source github repository.† This repository contains pre-trained models for a subset of the same architectures listed in Section 7.6.1, excluding AlexNet, SqueezeNet, ResNet-101, ResNeXt; the released VGG pre-trained models represent variants with batch normalization. We order these CIFAR-10 pre-trained models according to their validation accuracies reported in the above repository. Compared to ImageNet and ObjectNet, pre-trained models on CIFAR-10 exhibit a much higher level of accuracy (between 93% and 95%) with much smaller differences between the best and worst model. As a consequence, our experiment on CIFAR-10 emulates a scenario in which incoming new classifiers exhibit smaller performance gains. This might reflect a practical prediction-updates setting with a higher frequency of incoming new models.

The results of our experiments on CIFAR-10 are summarised in Figure G.6 and Figure G.5. Our method shows very similar trends w.r.t. all update strategies and metrics as we have worked out on ImageNet and ObjectNet. Interestingly, there is one novel characteristic when it comes to the accuracy gains: due to the presumably less steep increase in accuracy from one model to its successor and fewer class categories, we can form more accurate posterior beliefs which results in accuracy gains for our CR 10 with smallest budget that outperforms the backfilling Replace (100) baseline. The overall accuracy gain similarly remains competitive with all Majority Vote variants

* ImageNet pre-trained models: link
† CIFAR-10 pre-trained models: link



(a) Diagonal confusion matrix estimate



(b) Full confusion matrix estimate

Figure G.6: Summary of prediction-updates on CIFAR-10. We observe very similar behaviour to that described for ImageNet and ObjectNet in Section 7.6.2 and Section 7.6.3, respectively. The most interesting difference compared to our experiments on ImageNet and ObjectNet, is that, on CIFAR-10, our methods achieve substantially larger accuracy gains compared with the baselines, even for small budgets.

while exhibiting only a fraction of the negative flips. Again, BTC and BEC strongly outperform all baselines along every update step for all our methods as can be seen in Figure G.7

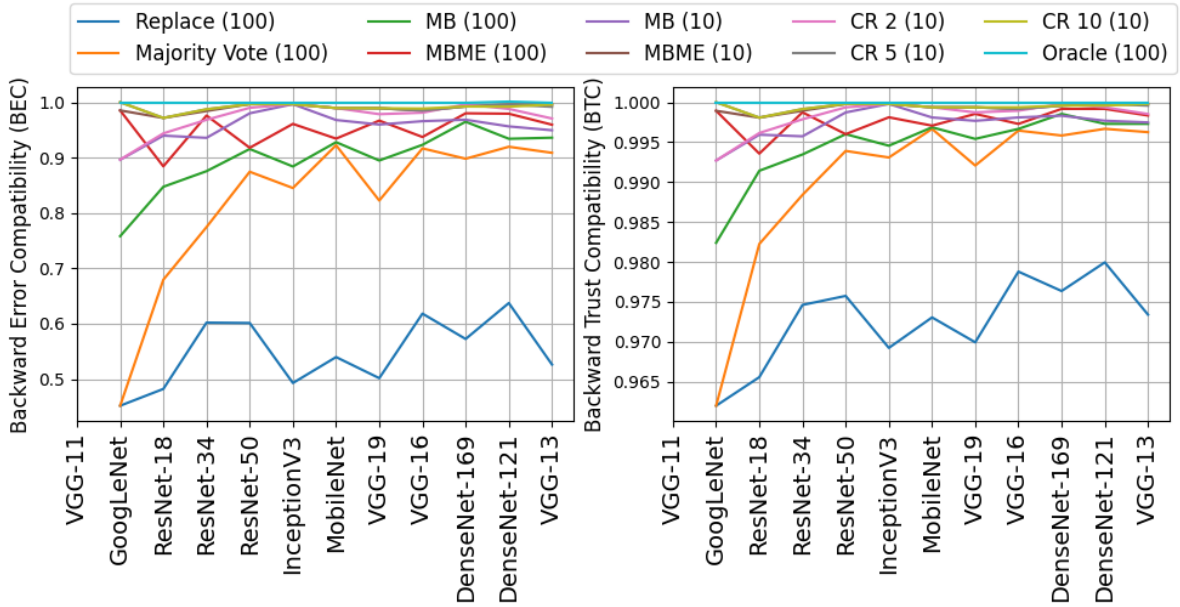


Figure G.7: BTC and BEC evolution on CIFAR10 with utilizing diagonal confusion matrix elements only.

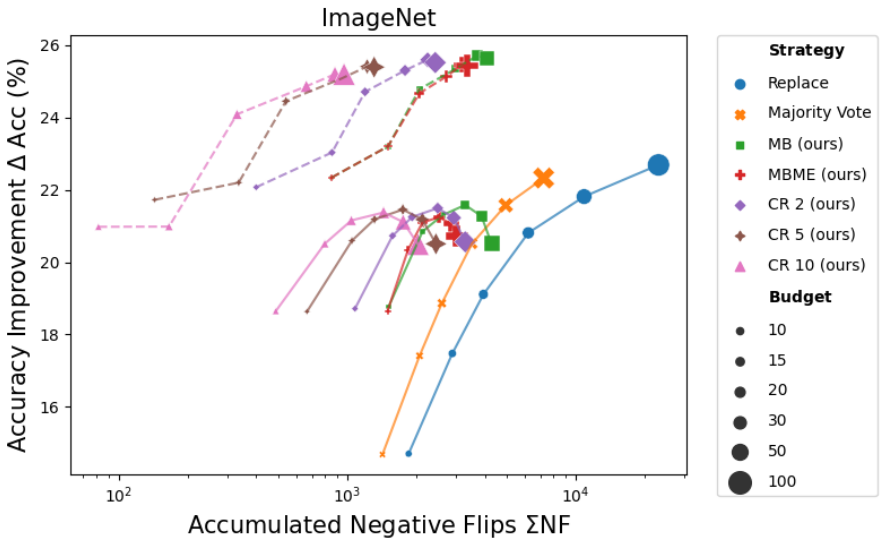


Figure G.8: Further investigation into the peaking phenomenon observed for prediction-updates on ImageNet in Figure 7.3 (right). Solid lines are as in Figure 7.3 (right), i.e., using a 50:50 split of the validation set, using one half to estimate the diagonal elements of the confusion matrices and the other half as our target evaluation set; dashed lines show the results of our ablation experiment where we use the full validation set to both estimate the exact statistics of the confusion matrix and to evaluate the different methods. Since the evaluation set for the latter is twice as large, we scaled the reported negative flips by 0.5 for ease of comparison.

G.6 On Peaking Accuracies

In Figure 7.3 (right, solid lines), we observed a peak in Δ**Acc** at 30% budget for all our strategies when performing prediction-updates on ImageNet. This suggests that our update rules are no longer beneficial on this dataset when re-evaluating more than the top 30% of samples with highest posterior entropy. From our ablation experiments on different confusion matrix estimates shown in Figure G.2 (left), we observe that this peaking phenomenon is much less pronounced and occurs only at larger budgets of $B \geq 50\%$ when estimating the full confusion matrix based on the half of the validation set on which we do not evaluate. We therefore conjecture that this behaviour may be related to inaccurate estimates of the posterior resulting from our approximation of the unknown confusion matrices (and possibly the assumption of conditionally independent classifiers).

To further investigate this hypothesis, we performed an ablation where we use the *full* validation set both to estimate the confusion matrix (with smoothing) and for evaluation—i.e., we do not use a 50-50 split of the validation set as in Figure G.2 and all experiment in the main paper—thus matching almost exactly[‡] the confusion counts statistics of the evaluation set. As a result, we can expect our estimates of the posterior distribution to be much more accurate in this case.

The results are shown in Figure G.8 in dashed lines, compared to the peaking behaviour from our default method in solid lines for reference. Apart from substantial gains in overall accuracy, we find that the peaking phenomenon disappears almost entirely,with only a very slight drop in accuracy remaining between budgets of 50% and 100% for some strategies. This seems to confirm our hypothesis that the main source of the peaking behaviour are approximation errors in our posterior estimates resulting from inaccurate confusion matrices.

G.7 Role of the Selection Strategy

We also conduct an ablation study on the role of the selection strategy for samples to be re-evaluated as was being mentioned in Section 7.6.4. In particular we conduct a comparison between our posterior entropy selection and the random selection (without replacement) baseline for all our methods across a range of budgets—see Figure G.9, Figure G.10 and Figure G.11 for these ablation results on all three datasets CIFAR-10, ImageNet and ObjectNet. Along all datasets, we find that random selection leads to substantially smaller accuracy gains, but also results in fewer negative flips. We can explain these fewer negative flips by the fact that our random selection more often chooses "easy" samples for re-evaluation. Under a random selection criteria, there is no preference in re-evaluating more "controversial" samples w.r.t previous predictions. This effect becomes particularly pronounced for small budgets. Our finding suggests that the selection strategy is indeed a relevant component, with room for improvement as discussed in Section 7.4. This is also highlighting the potential control capability of this component regarding our three desiderata.

[‡] up to the one count smoothing effect.

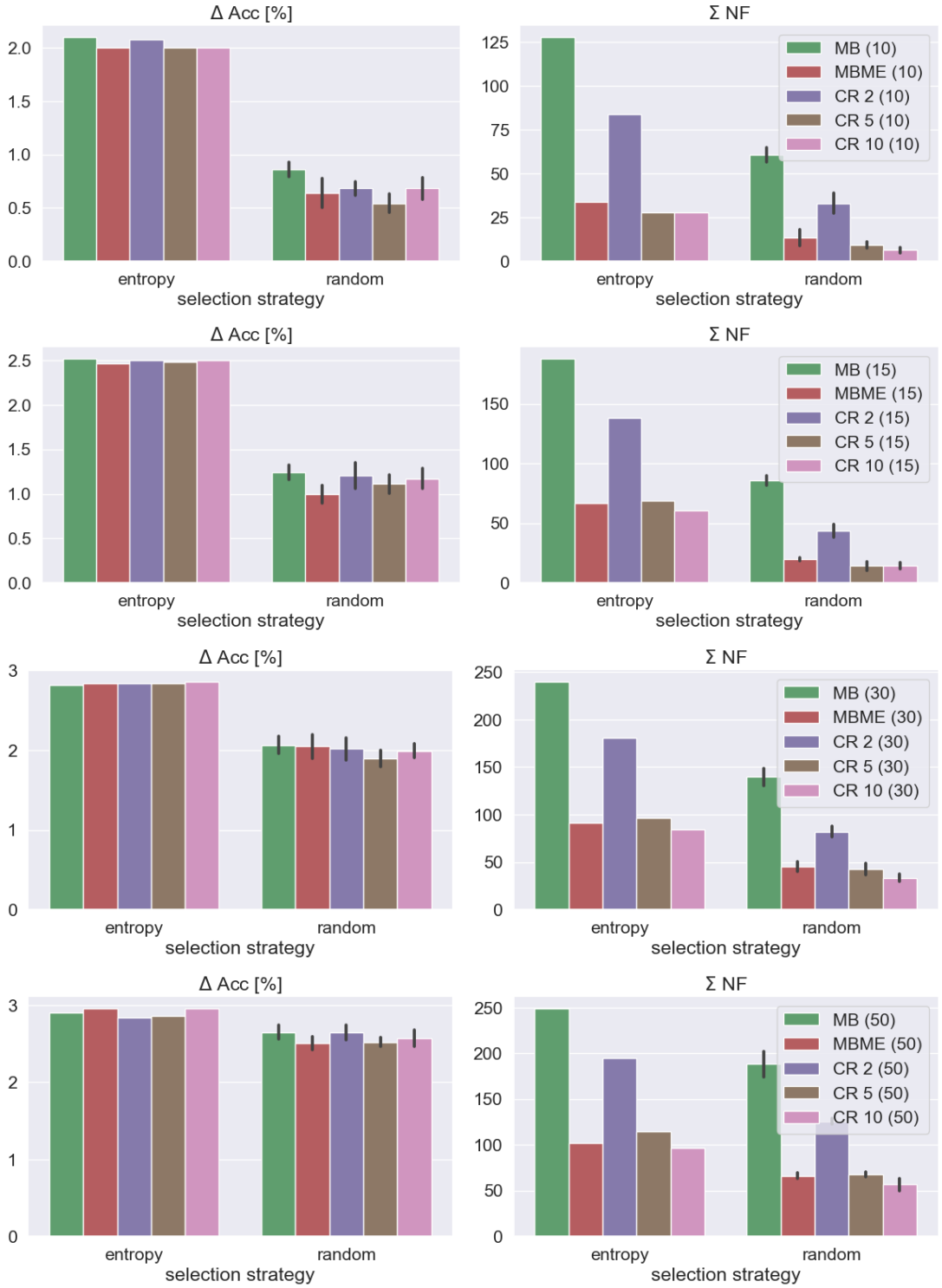


Figure G.9: Ablation results for comparing random (dashed lines in scatter plots) and entropy-ranked (solid lines in scatter plots) selection strategies on CIFAR-10. Different rows of the bar plots correspond to, from top to bottom, B =10, 15, 30, 50%. Note that at 100% budgets, the two selection strategies are identical as all samples are re-evaluated at each step. We find that random selection leads to substantially smaller accuracy gains, but also fewer negative flips. This is intuitive as random selection more often chooses "easy" samples for re-evaluation. The effect is particularly pronounced for small budgets.

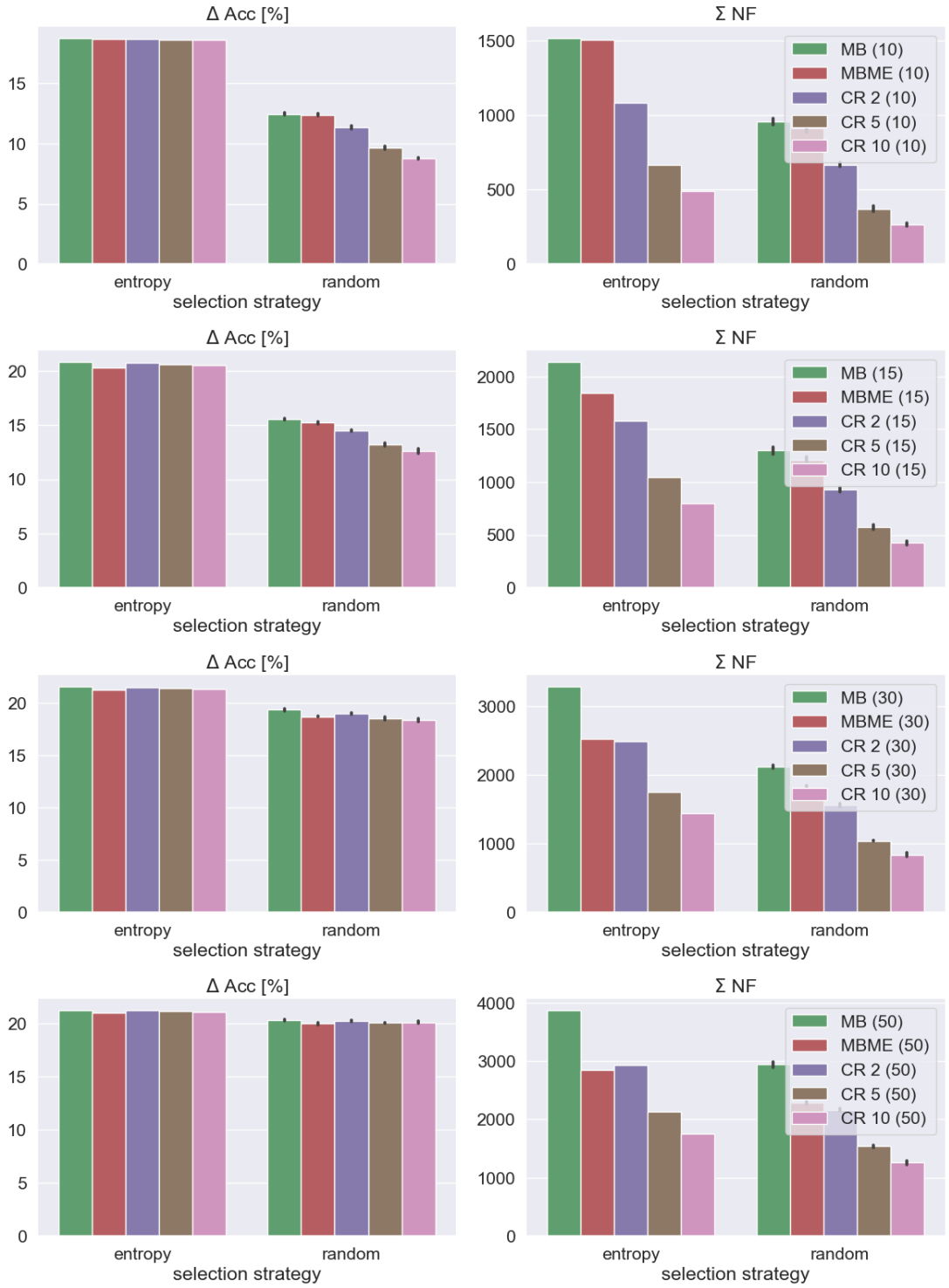


Figure G.10: Ablation results for comparing random (dashed lines in scatter plots) and entropy-ranked (solid lines in scatter plots) selection strategies on ImageNet. Different rows of the bar plots correspond to, from top to bottom, B =10, 15, 30, 50%. Note that at 100% budgets, the two selection strategies are identical as all samples are re-evaluated at each step. We find that random selection leads to substantially smaller accuracy gains, but also fewer negative flips. This is intuitive as random selection more often chooses "easy" samples for re-evaluation. The effect is particularly pronounced for small budgets.

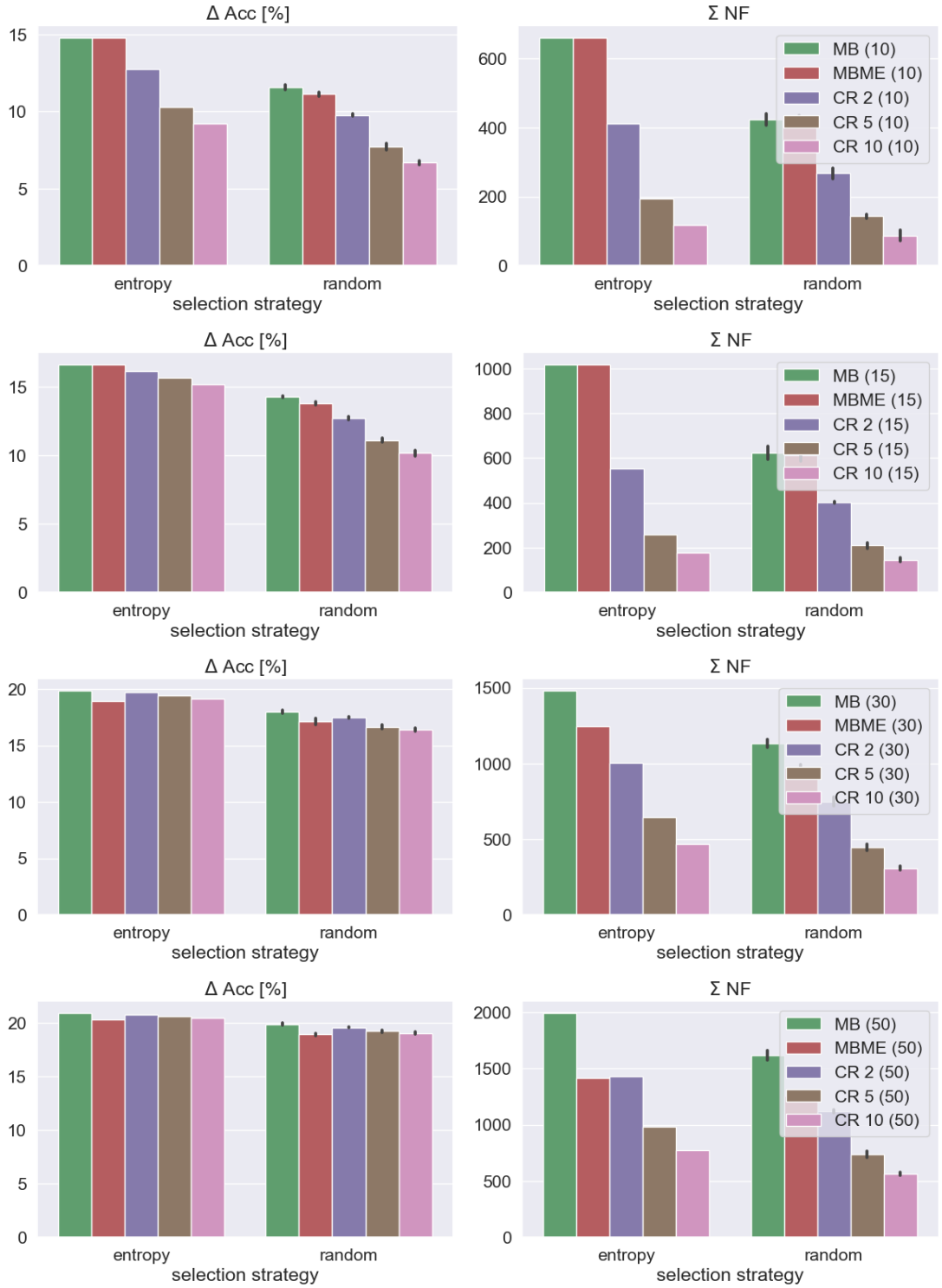


Figure G.11: Ablation results for comparing random (dashed lines in scatter plots) and entropy-ranked (solid lines in scatter plots) selection strategies on ObjectNet. Different rows of the bar plots correspond to, from top to bottom, B =10, 15, 30, 50%. Note that at 100% budgets, the two selection strategies are identical as all samples are re-evaluated at each step. We find that random selection leads to substantially smaller accuracy gains, but also fewer negative flips. This is intuitive as random selection more often chooses "easy" samples for re-evaluation. The effect is particularly pronounced for small budgets.

G.8 Evolution of the Stored Label Distribution

For additional insights which samples are getting selected for re-evaluation by our entropy selection criterion, we also show some characteristic count distribution of correctly and incorrectly stored predictions along various update steps. See Figure G.12 for this distribution after every second update step on ImageNet using the MB strategy with a compute budget of $B = 10$ (%).

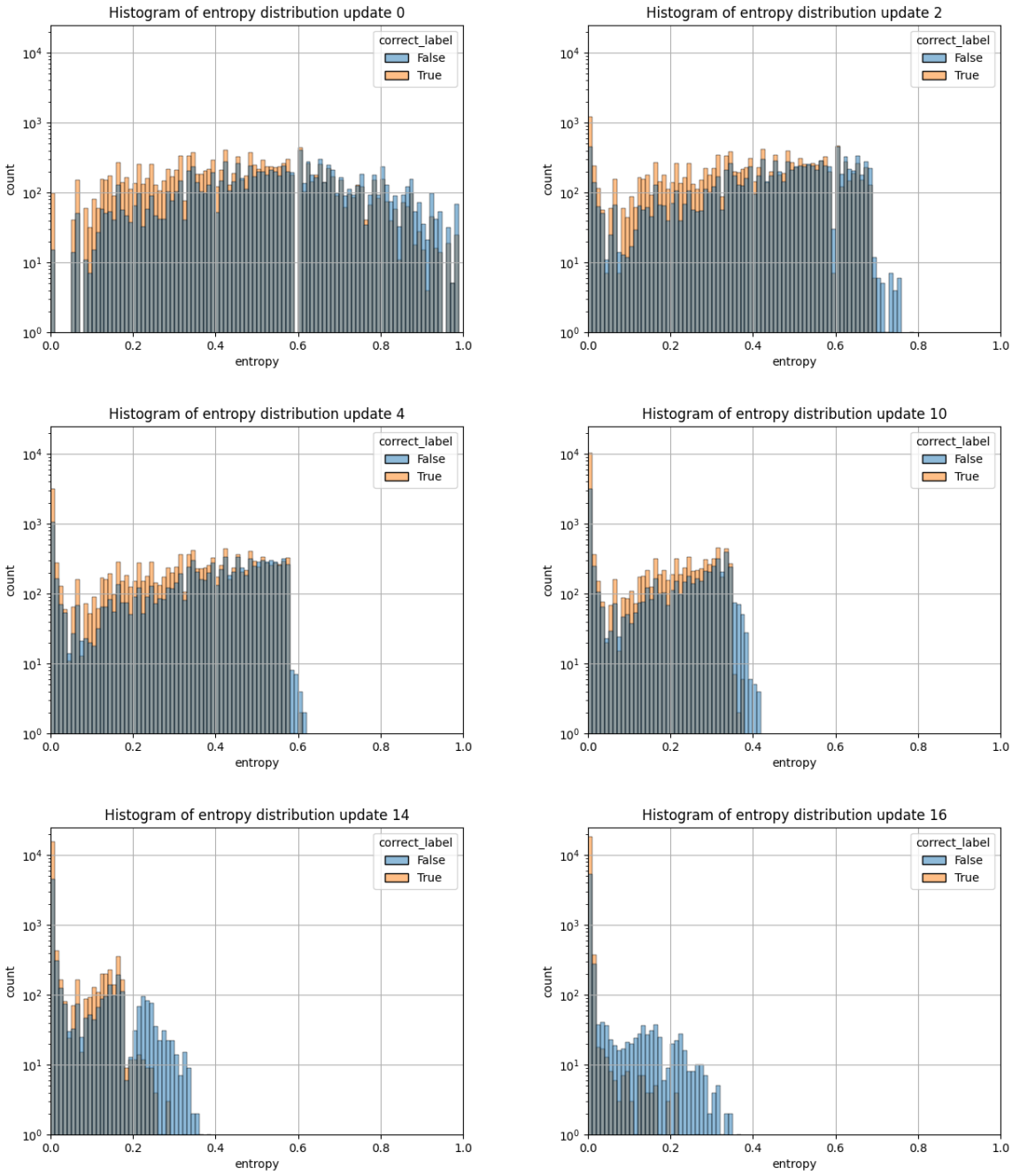


Figure G.12: Count distribution evolution of correctly and incorrectly stored predictions on ImageNet using the MB update strategy under a budget of B = 10. Samples with currently stored predictions being false are depicted with blue, and correctly stored predictions with orange. We observe that under this MB strategy, samples with highest entropy tend to store false predictions, thus benefitting from re-evaluation particularly.

G.9 Reducing re-evaluations matters at scale

To see that the re-evaluations of an example image using deep neural network based models clearly dominate computational cost as compared to our method backbone that involved computing the approximate posterior, label entropy and update strategy details, we measure the time for the plain model estimation vs the time for posterior update on the same Intel Xeon (Cascade Lake) CPU for the sake of comparison. We summarize our measurement in Table G.1 for all used ImageNet models, showing the multiples in compute time required by the model inference (per image on average) compared to the timing per image of our fully unoptimized method implementation, i.e. the posterior update backbone. For very large data sets and with new models generally increasing in size, reducing the inference budget B is of crucial importance, emphasizing the relevance of desideratum 3.

Table G.1: Time measurements of a single forward pass versus the mean average time to compute the posterior update (0.406 milliseconds) on the same computational ressources for comparability. For reference we extrapolated how long it would take to infer the predictions from 1B images.

Model	model re-evaluation [milliseconds]	ratio inference vs posterior update	inference for 1B images [days]
alexnet	29,6	73	343
squeezenet	32,6	80	378
vgg11	125,5	309	1452
googlenet	65,6	162	759
resnet18	31,5	78	364
vgg13	164,1	404	1899
vgg16	191,3	471	2215
mobilenet	23,1	57	267
vgg19	223,0	550	2582
resnet34	50,8	125	587
densenet121	74,4	183	862
densenet169	93,5	230	1082
resent50	70,6	174	818
resnet101	114,0	281	1319
inceptionv3	115,8	285	1340
resnet152	158,2	390	1831
resnext101 32x8d	206,6	509	2391

G.10 Result Tables of Main Experiments

For sake of completeness and reproducibility, we also provide a complete quantitative account of all the main experiments and ablations on all three datasets discussed throughout this paper in Table G.2, Table G.3 and Table G.4.

Table G.2: Full results of all experiments on **ImageNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote.

Strategy Budget %	**Acc** (%)	**ΔAcc** (%)	**Σ NF**	**NFR** (%)	**PF / NF**	**Avg. BTC**	**Avg. BEC**
100:Oracle	91.2 (91.2)	34.7 (34.7)	0 (0)	0.0 (0.0)	NaN (NaN)	100.0 (100.0)	100.0 (100.0)
R100:Replace	79.2 (79.2)	22.7 (22.7)	24214 (24214)	6.05 (6.05)	1.2 (1.2)	91.37 (91.37)	77.71 (77.71)
R100:Majority Vote	78.9 (78.9)	22.3 (22.3)	7352 (7352)	1.84 (1.84)	1.8 (1.8)	97.18 (97.18)	93.95 (93.95)
E100:MB	77.1 (78.7)	20.5 (22.2)	4378 (4373)	1.09 (1.09)	2.2 (2.3)	98.32 (98.32)	96.45 (96.4)
E100:MBME	77.3 (78.5)	20.7 (22.0)	3057 (3910)	0.76 (0.98)	2.7 (2.4)	98.78 (98.48)	97.69 (96.88)
E100:CR 2	77.1 (78.5)	20.6 (22.0)	3368 (2342)	0.84 (0.59)	2.5 (3.3)	98.72 (99.12)	97.19 (98.0)
E100:CR 5	77.1 (78.4)	20.5 (21.8)	2520 (1300)	0.63 (0.33)	3.0 (5.2)	99.06 (99.52)	97.82 (98.86)
E100:CR 10	77.0 (78.2)	20.5 (21.7)	2112 (1023)	0.53 (0.26)	3.4 (6.3)	99.22 (99.62)	98.15 (99.11)
R50:Replace	78.4 (78.4)	21.8 (21.8)	11478.8 (11478.8)	2.87 (2.87)	1.5 (1.5)	95.82 (95.82)	89.88 (89.88)
R50:Majority Vote	78.1 (78.1)	21.6 (21.6)	5048.2 (5048.2)	1.26 (1.26)	2.1 (2.1)	98.07 (98.07)	95.92 (95.92)
R50:MB	76.9 (77.7)	20.4 (21.1)	3022.8 (3052.8)	0.76 (0.76)	2.7 (2.7)	98.83 (98.82)	97.62 (97.57)
R50:MBME	76.5 (77.5)	20.0 (20.9)	2325.4 (2917.6)	0.58 (0.73)	3.1 (2.8)	99.08 (98.87)	98.27 (97.72)
R50:CR 2	76.8 (77.4)	20.3 (20.8)	2233.6 (1572.8)	0.56 (0.39)	3.3 (4.3)	99.14 (99.4)	98.22 (98.74)
R50:CR 5	76.6 (77.0)	20.1 (20.4)	1603.8 (840.6)	0.4 (0.21)	4.1 (7.1)	99.39 (99.68)	98.69 (99.32)
R50:CR 10	76.7 (76.7)	20.1 (20.2)	1314.6 (659.2)	0.33 (0.16)	4.8 (8.6)	99.51 (99.75)	98.92 (99.46)
E50:Replace	78.9 (79.6)	22.4 (23.1)	15732 (15503)	3.93 (3.88)	1.4 (1.4)	94.41 (94.53)	85.37 (85.23)
E50:Majority Vote	78.7 (79.1)	22.2 (22.6)	6318 (6420)	1.58 (1.6)	1.9 (1.9)	97.6 (97.58)	94.74 (94.55)
E50:MB	77.8 (79.0)	21.3 (22.4)	3969 (3941)	0.99 (0.99)	2.3 (2.4)	98.48 (98.5)	96.78 (96.71)
E50:MBME	77.6 (78.7)	21.1 (22.2)	2904 (3636)	0.73 (0.91)	2.8 (2.5)	98.85 (98.6)	97.79 (97.05)
E50:CR 2	77.8 (78.8)	21.2 (22.2)	3014 (2128)	0.75 (0.53)	2.8 (3.6)	98.86 (99.21)	97.5 (98.17)
E50:CR 5	77.7 (78.6)	21.2 (22.0)	2214 (1168)	0.55 (0.29)	3.4 (5.7)	99.18 (99.57)	98.09 (98.96)
E50:CR 10	77.7 (78.3)	21.1 (21.8)	1832 (913)	0.46 (0.23)	3.9 (7.0)	99.33 (99.67)	98.4 (99.19)
R30:Replace	77.4 (77.4)	20.8 (20.8)	6546.4 (6546.4)	1.64 (1.64)	1.8 (1.8)	97.56 (97.56)	94.53 (94.53)
R30:Majority Vote	77.1 (77.1)	20.5 (20.5)	3616.4 (3616.4)	0.9 (0.9)	2.4 (2.4)	98.6 (98.6)	97.18 (97.18)
R30:MB	75.9 (76.3)	19.4 (19.7)	2186.2 (2200.4)	0.55 (0.55)	3.2 (3.2)	99.14 (99.14)	98.35 (98.34)
R30:MBME	75.3 (76.2)	18.7 (19.6)	1859.6 (2183.8)	0.46 (0.55)	3.5 (3.2)	99.26 (99.14)	98.65 (98.36)
R30:CR 2	75.5 (75.5)	19.0 (18.9)	1607.0 (1092.4)	0.4 (0.27)	4.0 (5.3)	99.37 (99.57)	98.8 (99.19)
R30:CR 5	75.1 (74.7)	18.5 (18.2)	1087.2 (564.8)	0.27 (0.14)	5.3 (9.1)	99.58 (99.78)	99.17 (99.57)
R30:CR 10	74.9 (74.2)	18.4 (17.7)	875.4 (434.2)	0.22 (0.11)	6.2 (11.2)	99.66 (99.83)	99.33 (99.68)
E30:Replace	78.5 (79.3)	22.0 (22.8)	9708 (8900)	2.43 (2.23)	1.6 (1.6)	96.53 (96.82)	91.01 (91.69)
E30:Majority Vote	78.5 (79.2)	22.0 (22.6)	5232 (4989)	1.31 (1.25)	2.1 (2.1)	98.03 (98.12)	95.63 (95.79)
E30:MB	78.1 (78.9)	21.6 (22.4)	3375 (3180)	0.84 (0.8)	2.6 (2.8)	98.71 (98.79)	97.25 (97.37)
E30:MBME	77.8 (78.8)	21.2 (22.2)	2577 (3071)	0.64 (0.77)	3.1 (2.8)	98.98 (98.83)	98.04 (97.49)
E30:CR 2	78.0 (78.7)	21.5 (22.1)	2578 (1664)	0.64 (0.42)	3.1 (4.3)	99.02 (99.37)	97.86 (98.58)
E30:CR 5	78.0 (78.3)	21.5 (21.8)	1831 (890)	0.46 (0.22)	3.9 (7.1)	99.32 (99.67)	98.43 (99.21)
E30:CR 10	77.9 (78.1)	21.4 (21.6)	1517 (707)	0.38 (0.18)	4.5 (8.6)	99.44 (99.74)	98.67 (99.37)
R20:Replace	75.7 (75.7)	19.1 (19.1)	4171.0 (4171.0)	1.04 (1.04)	2.1 (2.1)	98.41 (98.41)	96.71 (96.71)
R20:Majority Vote	75.4 (75.4)	18.9 (18.9)	2690.2 (2690.2)	0.67 (0.67)	2.8 (2.8)	98.95 (98.95)	97.99 (97.99)
R20:MB	74.0 (74.2)	17.5 (17.6)	1662.8 (1655.4)	0.42 (0.41)	3.6 (3.7)	99.34 (99.34)	98.81 (98.81)
R20:MBME	73.5 (74.1)	16.9 (17.6)	1480.6 (1654.0)	0.37 (0.41)	3.9 (3.7)	99.4 (99.34)	98.97 (98.82)
R20:CR 2	73.2 (72.5)	16.7 (15.9)	1205.6 (786.2)	0.3 (0.2)	4.5 (6.1)	99.52 (99.68)	99.15 (99.45)
R20:CR 5	72.3 (71.3)	15.8 (14.7)	768.0 (388.8)	0.19 (0.1)	6.1 (10.5)	99.69 (99.84)	99.46 (99.73)
R20:CR 10	71.9 (70.5)	15.4 (14.0)	603.8 (280.2)	0.15 (0.07)	7.4 (13.5)	99.76 (99.89)	99.57 (99.81)
E20:Replace	78.5 (79.0)	22.0 (22.5)	6191 (5430)	1.55 (1.36)	1.9 (2.0)	97.74 (98.02)	94.47 (95.16)
E20:Majority Vote	78.3 (78.7)	21.8 (22.1)	4295 (3788)	1.07 (0.95)	2.3 (2.5)	98.37 (98.56)	96.46 (96.89)
E20:MB	77.9 (78.1)	21.3 (21.6)	2700 (2371)	0.68 (0.59)	3.0 (3.3)	98.96 (99.09)	97.86 (98.11)
E20:MBME	77.6 (78.1)	21.1 (21.6)	2183 (2371)	0.55 (0.59)	3.4 (3.3)	99.13 (99.09)	98.39 (98.11)
E20:CR 2	77.8 (77.8)	21.3 (21.3)	1999 (1197)	0.5 (0.3)	3.7 (5.4)	99.23 (99.54)	98.41 (99.04)
E20:CR 5	77.7 (77.4)	21.2 (20.8)	1383 (599)	0.35 (0.15)	4.8 (9.7)	99.47 (99.77)	98.87 (99.49)
E20:CR 10	77.7 (76.7)	21.2 (20.2)	1101 (461)	0.28 (0.12)	5.8 (11.9)	99.58 (99.83)	99.08 (99.62)
R10:Replace	71.3 (71.3)	14.7 (14.7)	1958.4 (1958.4)	0.49 (0.49)	2.9 (2.9)	99.22 (99.22)	98.63 (98.63)
R10:Majority Vote	71.2 (71.2)	14.7 (14.7)	1481.4 (1481.4)	0.37 (0.37)	3.5 (3.5)	99.4 (99.4)	98.98 (98.98)
R10:MB	69.0 (69.0)	12.5 (12.4)	991.4 (963.6)	0.25 (0.24)	4.1 (4.2)	99.59 (99.6)	99.35 (99.36)
R10:MBME	68.9 (69.0)	12.4 (12.4)	944.8 (976.8)	0.24 (0.24)	4.3 (4.2)	99.61 (99.6)	99.38 (99.35)
R10:CR 2	67.9 (66.0)	11.4 (9.5)	703.6 (432.2)	0.18 (0.11)	5.0 (6.5)	99.71 (99.82)	99.54 (99.73)
R10:CR 5	66.2 (64.2)	9.6 (7.6)	393.4 (147.0)	0.1 (0.04)	7.1 (14.0)	99.84 (99.94)	99.75 (99.91)
R10:CR 10	65.3 (63.5)	8.8 (7.0)	282.6 (104.8)	0.07 (0.03)	8.7 (17.6)	99.88 (99.96)	99.82 (99.93)
E10:Replace	76.1 (77.3)	19.5 (20.8)	2468 (2320)	0.62 (0.58)	3.0 (3.2)	99.04 (99.11)	98.12 (98.12)
E10:Majority Vote	75.9 (77.0)	19.3 (20.5)	2417 (2235)	0.6 (0.56)	3.0 (3.3)	99.06 (99.14)	98.18 (98.22)
E10:MB	75.3 (76.4)	18.8 (19.9)	1557 (1394)	0.39 (0.35)	4.0 (4.6)	99.38 (99.44)	98.89 (98.98)
E10:MBME	75.2 (76.4)	18.7 (19.9)	1533 (1394)	0.38 (0.35)	4.0 (4.6)	99.38 (99.44)	98.92 (98.98)
E10:CR 2	75.3 (76.1)	18.7 (19.5)	1118 (618)	0.28 (0.15)	5.2 (8.9)	99.55 (99.75)	99.22 (99.56)
E10:CR 5	75.2 (75.2)	18.6 (18.6)	700 (246)	0.18 (0.06)	7.7 (19.9)	99.72 (99.9)	99.51 (99.81)
E10:CR 10	75.2 (73.3)	18.6 (16.8)	515 (170)	0.13 (0.04)	10.1 (25.7)	99.79 (99.93)	99.64 (99.87)

Table G.3: Full results of all experiments on **ObjectNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote.

Strategy (Budget %)	**Acc** (%)	**ΔAcc** (%)	**Σ NF**	**NFR** (%)	**PF / NF**	**Avg. BTC**	**Avg. BEC**
100:Oracle	50.5 (50.5)	42.6 (42.6)	0 (0)	0.0 (0.0)	NaN (NaN)	100.0 (100.0)	99.99 (99.99)
R100:Replace	31.9 (31.9)	24.0 (24.0)	16669 (16669)	5.62 (5.62)	1.3 (1.3)	72.65 (72.65)	92.61 (92.61)
R100:Majority Vote	29.6 (29.6)	21.6 (21.6)	4690 (4690)	1.58 (1.58)	1.9 (1.9)	89.99 (89.99)	98.02 (98.02)
E100:MB	29.1 (29.1)	21.2 (21.1)	2477 (2675)	0.83 (0.9)	2.6 (2.5)	94.46 (94.04)	98.96 (98.88)
E100:MBME	28.6 (28.7)	20.6 (20.7)	1599 (2267)	0.54 (0.76)	3.4 (2.7)	95.86 (94.63)	99.34 (99.06)
E100:CR 2	29.0 (28.8)	21.0 (20.9)	1876 (1611)	0.63 (0.54)	3.1 (3.4)	95.92 (96.54)	99.21 (99.32)
E100:CR 5	28.8 (28.5)	20.8 (20.6)	1372 (853)	0.46 (0.29)	3.8 (5.5)	97.18 (98.28)	99.41 (99.63)
E100:CR 10	28.7 (28.2)	20.8 (20.3)	1084 (660)	0.37 (0.22)	4.6 (6.7)	97.82 (98.68)	99.54 (99.72)
R50:Replace	30.7 (30.7)	22.7 (22.7)	7583.8 (7583.8)	2.56 (2.56)	1.6 (1.6)	86.63 (86.63)	96.69 (96.69)
R50:Majority Vote	28.6 (28.6)	20.7 (20.7)	3281.6 (3281.6)	1.11 (1.11)	2.2 (2.2)	93.01 (93.01)	98.62 (98.62)
R50:MB	27.8 (27.7)	19.9 (19.8)	1676.6 (1780.4)	0.56 (0.6)	3.2 (3.1)	96.13 (95.9)	99.3 (99.26)
R50:MBME	26.9 (27.4)	18.9 (19.4)	1280.4 (1682.0)	0.43 (0.57)	3.7 (3.1)	96.77 (96.04)	99.48 (99.31)
R50:CR 2	27.5 (27.3)	19.6 (19.4)	1165.4 (923.0)	0.39 (0.31)	4.1 (4.9)	97.3 (97.86)	99.52 (99.62)
R50:CR 5	27.2 (26.6)	19.2 (18.7)	780.2 (469.6)	0.26 (0.16)	5.6 (8.4)	98.26 (98.98)	99.67 (99.8)
R50:CR 10	27.0 (26.2)	19.1 (18.3)	599.8 (353.2)	0.2 (0.12)	6.9 (10.6)	98.7 (99.22)	99.75 (99.85)
E50:Replace	31.0 (31.3)	23.1 (23.4)	7882 (7730)	2.66 (2.6)	1.5 (1.6)	86.26 (86.78)	96.53 (96.59)
E50:Majority Vote	29.5 (29.6)	21.5 (21.7)	3895 (3759)	1.31 (1.27)	2.0 (2.1)	91.73 (92.2)	98.36 (98.41)
E50:MB	28.9 (29.1)	20.9 (21.2)	2079 (2081)	0.7 (0.7)	2.9 (2.9)	95.23 (95.34)	99.13 (99.13)
E50:MBME	28.2 (28.6)	20.3 (20.7)	1452 (1944)	0.49 (0.66)	3.6 (3.0)	96.18 (95.52)	99.41 (99.19)
E50:CR 2	28.7 (28.7)	20.8 (20.8)	1506 (1136)	0.51 (0.38)	3.6 (4.4)	96.64 (97.54)	99.37 (99.52)
E50:CR 5	28.5 (28.4)	20.6 (20.4)	1049 (569)	0.35 (0.19)	4.6 (7.7)	97.83 (98.88)	99.55 (99.75)
E50:CR 10	28.4 (28.1)	20.5 (20.2)	828 (430)	0.28 (0.14)	5.6 (9.7)	98.35 (99.16)	99.64 (99.81)
R30:Replace	29.0 (29.0)	21.0 (21.0)	4070.6 (4070.6)	1.37 (1.37)	2.0 (2.0)	92.19 (92.19)	98.26 (98.26)
R30:Majority Vote	27.3 (27.3)	19.4 (19.4)	2346.6 (2346.6)	0.79 (0.79)	2.5 (2.5)	94.91 (94.91)	99.02 (99.02)
R30:MB	25.9 (25.9)	18.0 (17.9)	1185.8 (1218.6)	0.4 (0.41)	3.8 (3.7)	97.12 (97.01)	99.52 (99.5)
R30:MBME	25.1 (25.7)	17.2 (17.8)	1008.4 (1208.0)	0.34 (0.41)	4.2 (3.7)	97.41 (97.03)	99.59 (99.51)
R30:CR 2	25.4 (24.9)	17.5 (16.9)	782.2 (601.2)	0.26 (0.2)	5.2 (6.2)	98.05 (98.46)	99.68 (99.76)
R30:CR 5	24.6 (23.9)	16.6 (16.0)	476.2 (275.4)	0.16 (0.09)	7.5 (11.8)	98.8 (99.32)	99.81 (99.89)
R30:CR 10	24.4 (23.5)	16.4 (15.5)	331.6 (190.8)	0.11 (0.06)	10.2 (16.1)	99.17 (99.53)	99.86 (99.92)
E30:Replace	29.0 (29.7)	21.0 (21.7)	4316 (4093)	1.45 (1.38)	1.9 (2.0)	91.75 (92.39)	98.14 (98.23)
E30:Majority Vote	28.2 (28.6)	20.3 (20.6)	2970 (2703)	1.0 (0.91)	2.3 (2.4)	93.54 (94.23)	98.76 (98.87)
E30:MB	27.8 (27.6)	19.9 (19.7)	1565 (1402)	0.53 (0.47)	3.4 (3.6)	96.24 (96.66)	99.35 (99.42)
E30:MBME	26.9 (27.6)	18.9 (19.7)	1280 (1402)	0.43 (0.47)	3.7 (3.6)	96.64 (96.66)	99.48 (99.42)
E30:CR 2	27.7 (26.9)	19.7 (19.0)	1074 (675)	0.36 (0.23)	4.4 (6.2)	97.42 (98.32)	99.55 (99.72)
E30:CR 5	27.4 (26.0)	19.4 (18.1)	689 (312)	0.23 (0.11)	6.2 (11.8)	98.43 (99.31)	99.71 (99.87)
E30:CR 10	27.1 (25.4)	19.2 (17.5)	504 (214)	0.17 (0.07)	8.1 (16.2)	98.91 (99.53)	99.79 (99.91)
R20:Replace	27.0 (27.0)	19.1 (19.1)	2453.8 (2453.8)	0.83 (0.83)	2.4 (2.4)	94.82 (94.82)	98.97 (98.97)
R20:Majority Vote	25.8 (25.8)	17.8 (17.8)	1658.8 (1658.8)	0.56 (0.56)	3.0 (3.0)	96.17 (96.17)	99.32 (99.32)
R20:MB	23.8 (23.7)	15.8 (15.7)	887.2 (892.2)	0.3 (0.3)	4.3 (4.3)	97.74 (97.69)	99.64 (99.64)
R20:MBME	23.5 (23.9)	15.6 (15.9)	793.8 (866.2)	0.27 (0.29)	4.6 (4.4)	97.93 (97.77)	99.68 (99.65)
R20:CR 2	22.6 (21.8)	14.7 (13.8)	530.0 (392.2)	0.18 (0.13)	6.2 (7.6)	98.57 (98.89)	99.79 (99.84)
R20:CR 5	21.5 (20.4)	13.5 (12.5)	304.2 (171.2)	0.1 (0.06)	9.3 (14.5)	99.14 (99.52)	99.88 (99.93)
R20:CR 10	20.9 (19.9)	12.9 (11.9)	215.6 (121.4)	0.07 (0.04)	12.1 (19.2)	99.4 (99.65)	99.91 (99.95)
E20:Replace	26.9 (27.5)	19.0 (19.5)	2588 (2480)	0.87 (0.84)	2.4 (2.5)	94.54 (95.12)	98.91 (98.94)
E20:Majority Vote	26.2 (26.6)	18.3 (18.7)	2390 (2252)	0.81 (0.76)	2.4 (2.5)	94.8 (95.41)	99.0 (99.05)
E20:MB	26.0 (26.3)	18.1 (18.4)	1169 (1075)	0.39 (0.36)	3.9 (4.2)	97.01 (97.35)	99.53 (99.56)
E20:MBME	25.8 (26.3)	17.8 (18.4)	1131 (1075)	0.38 (0.36)	3.9 (4.2)	97.06 (97.35)	99.54 (99.56)
E20:CR 2	26.0 (26.2)	18.1 (18.2)	670 (490)	0.23 (0.17)	6.0 (7.9)	98.14 (98.71)	99.73 (99.8)
E20:CR 5	26.0 (25.6)	18.0 (17.7)	369 (193)	0.12 (0.07)	10.1 (18.0)	98.97 (99.55)	99.85 (99.92)
E20:CR 10	25.9 (24.9)	17.9 (16.9)	248 (134)	0.08 (0.05)	14.4 (24.4)	99.34 (99.69)	99.9 (99.94)
R10:Replace	22.1 (22.1)	14.2 (14.2)	996.6 (996.6)	0.34 (0.34)	3.6 (3.6)	97.49 (97.49)	99.6 (99.6)
R10:Majority Vote	21.6 (21.6)	13.6 (13.6)	808.8 (808.8)	0.27 (0.27)	4.1 (4.1)	97.86 (97.86)	99.68 (99.68)
R10:MB	19.5 (19.3)	11.6 (11.4)	450.2 (442.4)	0.15 (0.15)	5.8 (5.8)	98.7 (98.71)	99.82 (99.83)
R10:MBME	19.1 (19.1)	11.1 (11.2)	446.4 (449.0)	0.15 (0.15)	5.6 (5.6)	98.69 (98.69)	99.83 (99.82)
R10:CR 2	17.7 (16.1)	9.8 (8.1)	287.2 (196.6)	0.1 (0.07)	7.3 (8.7)	99.11 (99.34)	99.89 (99.92)
R10:CR 5	15.6 (14.1)	7.7 (6.2)	151.6 (74.6)	0.05 (0.03)	10.4 (16.4)	99.5 (99.75)	99.94 (99.97)
R10:CR 10	14.6 (13.5)	6.7 (5.5)	94.6 (46.0)	0.03 (0.02)	14.1 (23.4)	99.68 (99.84)	99.96 (99.98)
E10:Replace	23.7 (25.7)	15.7 (17.8)	996 (849)	0.34 (0.29)	3.9 (4.9)	97.5 (97.9)	99.6 (99.65)
E10:Majority Vote	23.7 (25.7)	15.7 (17.7)	996 (849)	0.34 (0.29)	3.9 (4.9)	97.5 (97.9)	99.6 (99.65)
E10:MB	22.7 (22.3)	14.8 (14.3)	696 (560)	0.23 (0.19)	4.9 (5.8)	98.08 (98.39)	99.72 (99.78)
E10:MBME	22.7 (22.3)	14.8 (14.3)	696 (560)	0.23 (0.19)	4.9 (5.8)	98.08 (98.39)	99.72 (99.78)
E10:CR 2	20.7 (17.8)	12.8 (9.9)	427 (213)	0.14 (0.07)	6.6 (9.6)	98.68 (99.21)	99.83 (99.92)
E10:CR 5	18.2 (15.7)	10.3 (7.7)	197 (49)	0.07 (0.02)	10.7 (30.3)	99.32 (99.82)	99.92 (99.98)
E10:CR 10	17.2 (14.9)	9.2 (7.0)	122 (24)	0.04 (0.01)	15.0 (54.9)	99.57 (99.91)	99.95 (99.99)

Table G.4: Full results of all experiments on **CIFAR10** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote.

Strategy (Budget %)	**Acc** (%)	**ΔAcc** (%)	**Σ NF**	**NFR** (%)	**PF / NF**	**Avg. BTC**	**Avg. BEC**
100:Oracle	99.3 (99.3)	6.1 (6.1)	0 (0)	0.0 (0.0)	NaN (NaN)	100.0 (100.0)	100.0 (100.0)
R100:Replace	94.7 (94.7)	1.5 (1.5)	1418 (1418)	2.58 (2.58)	1.1 (1.1)	97.26 (97.26)	54.79 (54.79)
R100:Majority Vote	96.1 (96.1)	2.9 (2.9)	502 (502)	0.91 (0.91)	1.3 (1.3)	99.03 (99.03)	81.94 (81.94)
E100:MB	95.9 (95.9)	2.8 (2.8)	284 (340)	0.52 (0.62)	1.5 (1.4)	99.45 (99.34)	89.64 (87.71)
E100:MBME	96.1 (96.0)	2.9 (2.8)	119 (146)	0.22 (0.27)	2.2 (2.0)	99.77 (99.72)	95.28 (94.47)
E100:CR 2	96.0 (96.0)	2.8 (2.8)	219 (170)	0.4 (0.31)	1.6 (1.8)	99.58 (99.67)	91.69 (93.28)
E100:CR 5	95.9 (95.9)	2.8 (2.8)	132 (126)	0.24 (0.23)	2.1 (2.1)	99.75 (99.76)	94.65 (94.98)
E100:CR 10	96.0 (96.0)	2.9 (2.8)	112 (103)	0.2 (0.19)	2.3 (2.4)	99.79 (99.8)	95.49 (95.92)
R50:Replace	94.7 (94.7)	1.6 (1.6)	711.4 (711.4)	1.29 (1.29)	1.1 (1.1)	98.62 (98.62)	77.8 (77.8)
R50:Majority Vote	95.7 (95.7)	2.6 (2.6)	327.4 (327.4)	0.6 (0.6)	1.4 (1.4)	99.37 (99.37)	89.06 (89.06)
R50:MB	95.8 (95.8)	2.7 (2.6)	196.4 (227.8)	0.36 (0.41)	1.7 (1.6)	99.62 (99.56)	93.41 (92.47)
R50:MBME	95.7 (95.6)	2.5 (2.5)	69.8 (90.6)	0.13 (0.16)	2.8 (2.4)	99.87 (99.83)	97.58 (96.93)
R50:CR 2	95.8 (95.8)	2.6 (2.6)	133.8 (104.2)	0.24 (0.19)	2.0 (2.3)	99.74 (99.8)	95.32 (96.24)
R50:CR 5	95.7 (95.7)	2.5 (2.5)	73.2 (69.0)	0.13 (0.13)	2.7 (2.8)	99.86 (99.87)	97.4 (97.59)
R50:CR 10	95.7 (95.7)	2.6 (2.5)	61.0 (60.6)	0.11 (0.11)	3.1 (3.1)	99.88 (99.88)	97.83 (97.85)
E50:Replace	95.2 (95.2)	2.0 (2.0)	1019 (1012)	1.85 (1.84)	1.1 (1.1)	98.04 (98.05)	64.97 (64.69)
E50:Majority Vote	96.1 (96.0)	2.9 (2.9)	428 (419)	0.78 (0.76)	1.3 (1.3)	99.18 (99.19)	84.48 (84.84)
E50:MB	96.1 (96.0)	2.9 (2.8)	260 (289)	0.47 (0.53)	1.6 (1.5)	99.5 (99.44)	90.63 (89.8)
E50:MBME	96.1 (96.0)	3.0 (2.9)	105 (134)	0.19 (0.24)	2.4 (2.1)	99.8 (99.74)	96.04 (95.09)
E50:CR 2	96.0 (96.0)	2.8 (2.8)	206 (149)	0.37 (0.27)	1.7 (1.9)	99.6 (99.71)	92.4 (94.27)
E50:CR 5	96.0 (95.9)	2.9 (2.8)	123 (112)	0.22 (0.2)	2.2 (2.2)	99.76 (99.79)	95.18 (95.67)
E50:CR 10	96.1 (96.0)	3.0 (2.8)	99 (95)	0.18 (0.17)	2.5 (2.5)	99.81 (99.82)	96.21 (96.35)
R30:Replace	94.6 (94.6)	1.4 (1.4)	439.6 (439.6)	0.8 (0.8)	1.2 (1.2)	99.15 (99.15)	86.74 (86.74)
R30:Majority Vote	95.3 (95.3)	2.2 (2.2)	238.4 (238.4)	0.43 (0.43)	1.5 (1.5)	99.54 (99.54)	92.54 (92.54)
R30:MB	95.2 (95.3)	2.1 (2.1)	146.2 (163.2)	0.27 (0.3)	1.7 (1.7)	99.72 (99.68)	95.46 (94.91)
R30:MBME	95.2 (95.2)	2.0 (2.1)	47.0 (64.2)	0.09 (0.12)	3.2 (2.6)	99.91 (99.88)	98.47 (97.92)
R30:CR 2	95.2 (95.2)	2.0 (2.0)	88.8 (64.6)	0.16 (0.12)	2.1 (2.6)	99.83 (99.88)	97.16 (97.92)
R30:CR 5	95.1 (95.0)	1.9 (1.9)	45.6 (47.0)	0.08 (0.09)	3.1 (3.0)	99.91 (99.91)	98.54 (98.49)
R30:CR 10	95.2 (95.1)	2.0 (2.0)	35.4 (35.4)	0.06 (0.06)	3.8 (3.8)	99.93 (99.93)	98.85 (98.85)
E30:Replace	95.6 (95.4)	2.5 (2.2)	688 (689)	1.25 (1.25)	1.2 (1.2)	98.68 (98.68)	75.18 (75.12)
E30:Majority Vote	95.9 (95.9)	2.8 (2.8)	380 (395)	0.69 (0.72)	1.4 (1.4)	99.27 (99.24)	86.44 (85.83)
E30:MB	96.0 (96.1)	2.8 (2.9)	253 (277)	0.46 (0.5)	1.6 (1.5)	99.51 (99.47)	91.14 (90.39)
E30:MBME	96.0 (96.2)	2.8 (3.0)	96 (123)	0.17 (0.22)	2.5 (2.2)	99.82 (99.76)	96.45 (95.68)
E30:CR 2	96.0 (96.2)	2.8 (3.0)	192 (147)	0.35 (0.27)	1.7 (2.0)	99.63 (99.72)	93.12 (94.5)
E30:CR 5	96.0 (96.2)	2.8 (3.0)	103 (98)	0.19 (0.18)	2.4 (2.6)	99.8 (99.81)	96.07 (96.36)
E30:CR 10	96.0 (96.2)	2.9 (3.0)	89 (86)	0.16 (0.16)	2.6 (2.7)	99.83 (99.83)	96.67 (96.85)
R20:Replace	94.5 (94.5)	1.3 (1.3)	274.6 (274.6)	0.5 (0.5)	1.2 (1.2)	99.47 (99.47)	91.94 (91.94)
R20:Majority Vote	94.9 (94.9)	1.7 (1.7)	195.2 (195.2)	0.35 (0.35)	1.4 (1.4)	99.62 (99.62)	94.06 (94.06)
R20:MB	94.7 (94.8)	1.6 (1.7)	107.0 (120.4)	0.19 (0.22)	1.7 (1.7)	99.79 (99.77)	96.86 (96.43)
R20:MBME	94.6 (94.6)	1.4 (1.4)	31.0 (45.4)	0.06 (0.08)	3.3 (2.6)	99.94 (99.91)	99.09 (98.67)
R20:CR 2	94.5 (94.5)	1.4 (1.4)	65.0 (48.6)	0.12 (0.09)	2.0 (2.4)	99.87 (99.91)	98.08 (98.55)
R20:CR 5	94.7 (94.7)	1.5 (1.5)	26.6 (27.4)	0.05 (0.05)	3.8 (3.7)	99.95 (99.95)	99.19 (99.16)
R20:CR 10	94.6 (94.6)	1.5 (1.4)	24.6 (24.4)	0.04 (0.04)	4.0 (3.9)	99.95 (99.95)	99.25 (99.26)
E20:Replace	95.8 (95.7)	2.6 (2.5)	462 (478)	0.84 (0.87)	1.3 (1.3)	99.11 (99.08)	83.35 (83.11)
E20:Majority Vote	95.9 (96.0)	2.8 (2.8)	336 (347)	0.61 (0.63)	1.4 (1.4)	99.35 (99.33)	88.2 (87.94)
E20:MB	96.0 (95.9)	2.9 (2.8)	217 (263)	0.39 (0.48)	1.7 (1.5)	99.58 (99.49)	92.45 (91.02)
E20:MBME	96.0 (96.0)	2.9 (2.8)	82 (115)	0.15 (0.21)	2.8 (2.2)	99.84 (99.78)	97.02 (96.04)
E20:CR 2	96.1 (95.9)	2.9 (2.8)	156 (130)	0.28 (0.24)	1.9 (2.1)	99.7 (99.75)	94.53 (95.27)
E20:CR 5	96.0 (95.9)	2.9 (2.8)	84 (90)	0.15 (0.16)	2.7 (2.5)	99.84 (99.83)	96.86 (96.71)
E20:CR 10	96.0 (95.9)	2.9 (2.8)	75 (80)	0.14 (0.15)	2.9 (2.7)	99.86 (99.85)	97.22 (97.11)
R10:Replace	94.1 (94.1)	0.9 (0.9)	129.2 (129.2)	0.23 (0.23)	1.3 (1.3)	99.75 (99.75)	96.31 (96.31)
R10:Majority Vote	94.3 (94.3)	1.1 (1.1)	109.4 (109.4)	0.2 (0.2)	1.5 (1.5)	99.79 (99.79)	96.86 (96.86)
R10:MB	94.0 (94.2)	0.9 (1.0)	64.0 (70.8)	0.12 (0.13)	1.7 (1.7)	99.88 (99.86)	98.23 (98.0)
R10:MBME	93.8 (93.8)	0.6 (0.6)	14.2 (21.4)	0.03 (0.04)	3.3 (2.4)	99.97 (99.96)	99.61 (99.41)
R10:CR 2	93.8 (93.8)	0.7 (0.7)	35.2 (23.8)	0.06 (0.04)	2.0 (2.4)	99.93 (99.95)	99.03 (99.34)
R10:CR 5	93.7 (93.7)	0.5 (0.6)	11.0 (11.0)	0.02 (0.02)	3.5 (3.6)	99.98 (99.98)	99.69 (99.69)
R10:CR 10	93.8 (93.8)	0.7 (0.7)	6.8 (7.0)	0.01 (0.01)	6.0 (5.9)	99.99 (99.99)	99.81 (99.8)
E10:Replace	95.3 (95.4)	2.1 (2.2)	206 (201)	0.37 (0.37)	1.5 (1.5)	99.6 (99.61)	93.45 (93.65)
E10:Majority Vote	95.4 (95.4)	2.2 (2.2)	200 (200)	0.36 (0.36)	1.6 (1.6)	99.61 (99.61)	93.67 (93.69)
E10:MB	95.3 (95.3)	2.1 (2.1)	140 (145)	0.25 (0.26)	1.8 (1.7)	99.73 (99.72)	95.59 (95.45)
E10:MBME	95.2 (95.2)	2.0 (2.0)	35 (66)	0.06 (0.12)	3.9 (2.5)	99.93 (99.87)	98.92 (98.02)
E10:CR 2	95.2 (95.2)	2.1 (2.1)	91 (59)	0.17 (0.11)	2.1 (2.7)	99.82 (99.89)	97.26 (98.11)
E10:CR 5	95.2 (95.2)	2.0 (2.0)	30 (38)	0.05 (0.07)	4.3 (3.7)	99.94 (99.93)	99.05 (98.79)
E10:CR 10	95.2 (95.2)	2.0 (2.0)	29 (36)	0.05 (0.07)	4.4 (3.8)	99.94 (99.93)	99.09 (98.86)

G.11 On Utilizing Uncalibrated Soft Labels

Finally, we tested our method on incroporating the uncalibrated soft labels provided as output in the pre-trained ImageNet models by repeating the ImageNet and ObjectNet experiment (in combination with full confusion matrix estimation). As already discussed in Section 7.4, deep neural networks are known to have unreliable uncertainty estimates [310–313] and we therefore do not expect these results to be representative of what our method could achieve if it has access to truly calibrated soft labels. We now utilize the full output softmax vector $\mathbf{p}^{sl} = C^t(\mathbf{x}_n)$ and refine our likelihood estimate in eq. (2) by multiplying the confusion matrix coefficients with the soft label vector, i.e. $\pi^t(\hat{y}_n^t, k) \rightarrow \sum_i p_i^{sl} \pi^t(i, k)$. We show the temporal evolution compared to the hard-label implementation using only diagonal confusion matrix estimates in Figure G.13 and a full account over the final performance metrics and all selection strategies and budgets in Table G.5, Table G.6 and Table G.7. Overall the results are slightly less conclusive and less consistent, most likely due to the known uncalibrated soft labels, emphasizing the strength of our method when only having deterministic labels. Specifically, on CIFAR-10, there are additional accuracy gains but often we accumulate more negative flips and BTC, BEC are typically slightly worse. On ImageNet, accuracy gains are sometimes worse without finding a clear pattern when this is the case, but negative flips are sometimes significantly lower. BTC, BEC seem to be better on the majority of update strategies and labels. On ObjectNet, we likewise find no clear patter in the accuracy gains. However, accumulated negative flips are generally much lower and both, BTC and BEC, even higher (very close to 1.0) but we want to emphasize that they were already very high for the ObjectNet experiments from the main paper.

Table G.5: Full results of all experiments on **CIFAR-10** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing **and incorporating soft labels**. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote.

Strategy Budget %	**Acc** (%)	**ΔAcc** (%)	**Σ NF**	**NFR** (%)	**PF / NF**	**Avg. BTC**	**Avg. BEC**
100:Oracle	99.3 (99.3)	6.1 (6.1)	0 (0)	0.0 (0.0)	NaN (NaN)	100.0 (100.0)	100.0 (100.0)
R100:Replace	94.7 (94.7)	1.5 (1.5)	1418 (1418)	2.58 (2.58)	1.1 (1.1)	97.26 (97.26)	54.79 (54.79)
R100:Majority Vote	96.1 (96.1)	2.9 (2.9)	502 (502)	0.91 (0.91)	1.3 (1.3)	99.03 (99.03)	81.94 (81.94)
E100:MB	95.9 (96.2)	2.8 (3.0)	284 (252)	0.52 (0.46)	1.5 (1.6)	99.45 (99.52)	89.64 (89.69)
E100:MBME	96.1 (96.1)	2.9 (3.0)	119 (173)	0.22 (0.31)	2.2 (1.9)	99.77 (99.67)	95.28 (92.98)
E100:CR 2	96.0 (96.1)	2.8 (3.0)	219 (170)	0.4 (0.31)	1.6 (1.9)	99.58 (99.68)	91.69 (92.98)
E100:CR 5	95.9 (96.2)	2.8 (3.0)	132 (103)	0.24 (0.19)	2.1 (2.5)	99.75 (99.8)	94.65 (95.68)
E100:CR 10	96.0 (96.2)	2.9 (3.0)	112 (73)	0.2 (0.13)	2.3 (3.1)	99.79 (99.86)	95.49 (96.99)
R50:Replace	94.7 (94.7)	1.6 (1.6)	711.4 (711.4)	1.29 (1.29)	1.1 (1.1)	98.62 (98.62)	77.8 (77.8)
R50:Majority Vote	95.7 (95.7)	2.6 (2.6)	327.4 (327.4)	0.6 (0.6)	1.4 (1.4)	99.37 (99.37)	89.06 (89.06)
R50:MB	95.8 (96.0)	2.7 (2.8)	196.4 (155.0)	0.36 (0.28)	1.7 (1.9)	99.62 (99.7)	93.41 (94.27)
R50:MBME	95.7 (95.9)	2.5 (2.7)	69.8 (106.0)	0.13 (0.19)	2.8 (2.3)	99.87 (99.8)	97.58 (96.2)
R50:CR 2	95.8 (96.0)	2.6 (2.9)	133.8 (99.8)	0.24 (0.18)	2.0 (2.4)	99.74 (99.81)	95.32 (96.25)
R50:CR 5	95.7 (96.0)	2.5 (2.8)	73.2 (59.2)	0.13 (0.11)	2.7 (3.4)	99.86 (99.89)	97.4 (97.77)
R50:CR 10	95.7 (95.9)	2.6 (2.7)	61.0 (40.8)	0.11 (0.07)	3.1 (4.3)	99.88 (99.92)	97.83 (98.47)
E50:Replace	95.2 (95.2)	2.0 (2.0)	1019 (1012)	1.85 (1.84)	1.1 (1.1)	98.04 (98.05)	64.97 (64.69)
E50:Majority Vote	96.1 (96.0)	2.9 (2.9)	428 (419)	0.78 (0.76)	1.3 (1.3)	99.18 (99.19)	84.48 (84.84)
E50:MB	96.1 (96.2)	2.9 (3.0)	260 (257)	0.47 (0.47)	1.6 (1.6)	99.5 (99.51)	90.63 (89.36)
E50:MBME	96.1 (96.1)	3.0 (3.0)	105 (170)	0.19 (0.31)	2.4 (1.9)	99.8 (99.67)	96.04 (93.08)
E50:CR 2	96.0 (96.1)	2.8 (3.0)	206 (170)	0.37 (0.31)	1.7 (1.9)	99.6 (99.68)	92.4 (92.92)
E50:CR 5	96.0 (96.2)	2.9 (3.0)	123 (102)	0.22 (0.19)	2.2 (2.5)	99.76 (99.81)	95.18 (95.72)
E50:CR 10	96.1 (96.2)	3.0 (3.0)	99 (71)	0.18 (0.13)	2.5 (3.1)	99.81 (99.86)	96.21 (97.07)
R30:Replace	94.6 (94.6)	1.4 (1.4)	439.6 (439.6)	0.8 (0.8)	1.2 (1.2)	99.15 (99.15)	86.74 (86.74)
R30:Majority Vote	95.3 (95.3)	2.2 (2.2)	238.4 (238.4)	0.43 (0.43)	1.5 (1.5)	99.54 (99.54)	92.54 (92.54)
R30:MB	95.2 (95.6)	2.1 (2.4)	146.2 (117.6)	0.27 (0.21)	1.7 (2.0)	99.72 (99.77)	95.46 (95.92)
R30:MBME	95.2 (95.6)	2.0 (2.4)	47.0 (88.4)	0.09 (0.16)	3.2 (2.4)	99.91 (99.83)	98.47 (96.94)
R30:CR 2	95.2 (95.5)	2.0 (2.4)	88.8 (72.6)	0.16 (0.13)	2.1 (2.6)	99.83 (99.86)	97.16 (97.5)
R30:CR 5	95.1 (95.4)	1.9 (2.3)	45.6 (40.8)	0.08 (0.07)	3.1 (3.8)	99.91 (99.92)	98.54 (98.61)
R30:CR 10	95.2 (95.4)	2.0 (2.2)	35.4 (28.8)	0.06 (0.05)	3.8 (4.8)	99.93 (99.94)	98.85 (99.03)
E30:Replace	95.6 (95.4)	2.5 (2.2)	688 (689)	1.25 (1.25)	1.2 (1.2)	98.68 (98.68)	75.18 (75.12)
E30:Majority Vote	95.9 (95.9)	2.8 (2.8)	380 (395)	0.69 (0.72)	1.4 (1.4)	99.27 (99.24)	86.44 (85.83)
E30:MB	96.0 (96.3)	2.8 (3.1)	253 (230)	0.46 (0.42)	1.6 (1.7)	99.51 (99.56)	91.14 (90.39)
E30:MBME	96.0 (96.2)	2.8 (3.0)	96 (163)	0.17 (0.3)	2.5 (1.9)	99.82 (99.69)	96.45 (93.3)
E30:CR 2	96.0 (96.2)	2.8 (3.1)	192 (160)	0.35 (0.29)	1.7 (2.0)	99.63 (99.69)	93.12 (93.28)
E30:CR 5	96.0 (96.2)	2.8 (3.1)	103 (94)	0.19 (0.17)	2.4 (2.6)	99.8 (99.82)	96.07 (96.01)
E30:CR 10	96.0 (96.3)	2.9 (3.1)	89 (57)	0.16 (0.1)	2.6 (3.8)	99.83 (99.89)	96.67 (97.65)
R20:Replace	94.5 (94.5)	1.3 (1.3)	274.6 (274.6)	0.5 (0.5)	1.2 (1.2)	99.47 (99.47)	91.94 (91.94)
R20:Majority Vote	94.9 (94.9)	1.7 (1.7)	195.2 (195.2)	0.35 (0.35)	1.4 (1.4)	99.62 (99.62)	94.06 (94.06)
R20:MB	94.7 (95.3)	1.6 (2.2)	107.0 (83.2)	0.19 (0.15)	1.7 (2.3)	99.79 (99.84)	96.86 (97.29)
R20:MBME	94.6 (95.3)	1.4 (2.2)	31.0 (62.0)	0.06 (0.11)	3.3 (2.7)	99.94 (99.88)	99.09 (97.98)
R20:CR 2	94.5 (95.2)	1.4 (2.0)	65.0 (56.4)	0.12 (0.1)	2.0 (2.8)	99.87 (99.89)	98.08 (98.17)
R20:CR 5	94.7 (95.2)	1.5 (2.0)	26.6 (25.6)	0.05 (0.05)	3.8 (4.9)	99.95 (99.95)	99.19 (99.16)
R20:CR 10	94.6 (94.9)	1.5 (1.8)	24.6 (19.6)	0.04 (0.04)	4.0 (5.6)	99.95 (99.96)	99.25 (99.38)
E20:Replace	95.8 (95.7)	2.6 (2.5)	462 (478)	0.84 (0.87)	1.3 (1.3)	99.11 (99.08)	83.35 (83.11)
E20:Majority Vote	95.9 (96.0)	2.8 (2.8)	336 (347)	0.61 (0.63)	1.4 (1.4)	99.35 (99.33)	88.2 (87.94)
E20:MB	96.0 (96.0)	2.9 (2.8)	217 (207)	0.39 (0.38)	1.7 (1.7)	99.58 (99.6)	92.45 (91.74)
E20:MBME	96.0 (95.9)	2.9 (2.8)	82 (148)	0.15 (0.27)	2.8 (1.9)	99.84 (99.72)	97.02 (94.17)
E20:CR 2	96.1 (96.0)	2.9 (2.8)	156 (133)	0.28 (0.24)	1.9 (2.1)	99.7 (99.75)	94.53 (94.67)
E20:CR 5	96.0 (96.0)	2.9 (2.9)	84 (80)	0.15 (0.15)	2.7 (2.8)	99.84 (99.85)	96.86 (96.73)
E20:CR 10	96.0 (96.1)	2.9 (2.9)	75 (55)	0.14 (0.1)	2.9 (3.7)	99.86 (99.89)	97.22 (97.89)
R10:Replace	94.1 (94.1)	0.9 (0.9)	129.2 (129.2)	0.23 (0.23)	1.3 (1.3)	99.75 (99.75)	96.31 (96.31)
R10:Majority Vote	94.3 (94.3)	1.1 (1.1)	109.4 (109.4)	0.2 (0.2)	1.5 (1.5)	99.79 (99.79)	96.86 (96.86)
R10:MB	94.0 (94.7)	0.9 (1.6)	64.0 (51.0)	0.12 (0.09)	1.7 (2.5)	99.88 (99.9)	98.23 (98.49)
R10:MBME	93.8 (94.6)	0.6 (1.4)	14.2 (36.2)	0.03 (0.07)	3.3 (2.9)	99.97 (99.93)	99.61 (98.93)
R10:CR 2	93.8 (94.7)	0.7 (1.5)	35.2 (32.4)	0.06 (0.06)	2.0 (3.3)	99.93 (99.94)	99.03 (99.03)
R10:CR 5	93.7 (94.3)	0.5 (1.1)	11.0 (14.0)	0.02 (0.03)	3.5 (5.1)	99.98 (99.97)	99.69 (99.59)
R10:CR 10	93.8 (94.3)	0.7 (1.1)	6.8 (7.4)	0.01 (0.01)	6.0 (8.5)	99.99 (99.99)	99.81 (99.78)
E10:Replace	95.3 (95.4)	2.1 (2.2)	206 (201)	0.37 (0.37)	1.5 (1.5)	99.6 (99.61)	93.45 (93.65)
E10:Majority Vote	95.4 (95.4)	2.2 (2.2)	200 (200)	0.36 (0.36)	1.6 (1.6)	99.61 (99.61)	93.67 (93.69)
E10:MB	95.3 (95.7)	2.1 (2.5)	140 (120)	0.25 (0.22)	1.8 (2.0)	99.73 (99.77)	95.59 (95.48)
E10:MBME	95.2 (95.6)	2.0 (2.5)	35 (99)	0.06 (0.18)	3.9 (2.3)	99.93 (99.81)	98.92 (96.32)
E10:CR 2	95.2 (95.7)	2.1 (2.5)	91 (78)	0.17 (0.14)	2.1 (2.6)	99.82 (99.85)	97.26 (97.08)
E10:CR 5	95.2 (95.6)	2.0 (2.4)	30 (52)	0.05 (0.09)	4.3 (3.3)	99.94 (99.9)	99.05 (98.04)
E10:CR 10	95.2 (95.5)	2.0 (2.3)	29 (37)	0.05 (0.07)	4.4 (4.1)	99.94 (99.93)	99.09 (98.73)

Table G.6: Full results of all experiments on **ImageNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing **and incorporating soft labels**. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote.

Strategy (Budget %)	**Acc** (%)	**ΔAcc** (%)	**Σ NF**	**NFR** (%)	**PF / NF**	**Avg. BTC**	**Avg. BEC**
100:Oracle	91.2 (91.2)	34.7 (34.7)	0 (0)	0.0 (0.0)	NaN (NaN)	100.0 (100.0)	100.0 (100.0)
R100:Replace	79.2 (79.2)	22.7 (22.7)	24214 (24214)	6.05 (6.05)	1.2 (1.2)	91.37 (91.37)	77.71 (77.71)
R100:Majority Vote	78.9 (78.9)	22.3 (22.3)	7352 (7352)	1.84 (1.84)	1.8 (1.8)	97.18 (97.18)	93.95 (93.95)
E100:MB	77.1 (77.6)	20.5 (21.0)	4378 (3463)	1.09 (0.87)	2.2 (2.5)	98.32 (98.65)	96.45 (97.25)
E100:MBME	77.3 (77.4)	20.7 (20.8)	3057 (3207)	0.76 (0.8)	2.7 (2.6)	98.78 (98.74)	97.69 (97.51)
E100:CR 2	77.1 (77.3)	20.6 (20.8)	3368 (1648)	0.84 (0.41)	2.5 (4.2)	98.72 (99.36)	97.19 (98.7)
E100:CR 5	77.1 (77.0)	20.5 (20.5)	2520 (990)	0.63 (0.25)	3.0 (6.2)	99.06 (99.63)	97.82 (99.19)
E100:CR 10	77.0 (76.8)	20.5 (20.3)	2112 (788)	0.53 (0.2)	3.4 (7.4)	99.22 (99.71)	98.15 (99.34)
R50:Replace	78.4 (78.4)	21.8 (21.8)	11478.8 (11463.8)	2.87 (2.87)	1.5 (1.5)	95.82 (95.83)	89.88 (89.89)
R50:Majority Vote	78.1 (78.1)	21.6 (21.6)	5048.2 (5048.8)	1.26 (1.26)	2.1 (2.1)	98.07 (98.07)	95.92 (95.92)
R50:MB	76.9 (76.9)	20.4 (20.4)	3022.8 (2509.8)	0.76 (0.63)	2.7 (3.0)	98.83 (99.02)	97.62 (98.05)
R50:MBME	76.5 (76.7)	20.0 (20.1)	2325.4 (2437.2)	0.58 (0.61)	3.1 (3.1)	99.08 (99.05)	98.27 (98.12)
R50:CR 2	76.8 (76.5)	20.3 (19.9)	2233.6 (1220.8)	0.56 (0.31)	3.3 (5.1)	99.14 (99.53)	98.22 (99.06)
R50:CR 5	76.6 (76.1)	20.1 (19.5)	1603.8 (700.4)	0.4 (0.18)	4.1 (8.0)	99.39 (99.73)	98.69 (99.44)
R50:CR 10	76.7 (75.8)	20.1 (19.2)	1314.6 (530.0)	0.33 (0.13)	4.8 (10.1)	99.51 (99.8)	98.92 (99.57)
E50:Replace	78.9 (79.8)	22.4 (23.2)	15732 (16138)	3.93 (4.03)	1.4 (1.4)	94.41 (94.3)	85.37 (84.73)
E50:Majority Vote	78.7 (79.2)	22.2 (22.6)	6318 (6282)	1.58 (1.57)	1.9 (1.9)	97.6 (97.64)	94.74 (94.67)
E50:MB	77.8 (77.9)	21.3 (21.4)	3969 (3212)	0.99 (0.8)	2.3 (2.7)	98.48 (98.77)	96.78 (97.4)
E50:MBME	77.6 (77.6)	21.1 (21.1)	2904 (2972)	0.73 (0.74)	2.8 (2.8)	98.85 (98.85)	97.79 (97.66)
E50:CR 2	77.8 (77.7)	21.2 (21.1)	3014 (1547)	0.75 (0.39)	2.8 (4.4)	98.86 (99.41)	97.5 (98.75)
E50:CR 5	77.7 (77.3)	21.2 (20.8)	2214 (900)	0.55 (0.22)	3.4 (6.8)	99.18 (99.67)	98.09 (99.24)
E50:CR 10	77.7 (77.1)	21.1 (20.6)	1832 (709)	0.46 (0.18)	3.9 (8.3)	99.33 (99.74)	98.4 (99.38)
R30:Replace	77.4 (77.3)	20.8 (20.8)	6546.4 (6537.8)	1.64 (1.63)	1.8 (1.8)	97.56 (97.56)	94.53 (94.55)
R30:Majority Vote	77.1 (77.1)	20.5 (20.5)	3616.4 (3613.6)	0.9 (0.9)	2.4 (2.4)	98.6 (98.6)	97.18 (97.18)
R30:MB	75.9 (75.9)	19.4 (19.4)	2186.2 (1860.6)	0.55 (0.47)	3.2 (3.6)	99.14 (99.27)	98.35 (98.6)
R30:MBME	75.3 (75.8)	18.7 (19.3)	1859.6 (1855.6)	0.46 (0.46)	3.5 (3.6)	99.26 (99.27)	98.65 (98.61)
R30:CR 2	75.5 (75.3)	19.0 (18.7)	1607.0 (918.2)	0.4 (0.23)	4.0 (6.1)	99.37 (99.64)	98.8 (99.31)
R30:CR 5	75.1 (74.5)	18.5 (17.9)	1087.2 (488.4)	0.27 (0.12)	5.3 (10.2)	99.58 (99.81)	99.17 (99.63)
R30:CR 10	74.9 (73.9)	18.4 (17.3)	875.4 (341.8)	0.22 (0.09)	6.2 (13.7)	99.66 (99.87)	99.33 (99.74)
E30:Replace	78.5 (79.1)	22.0 (22.6)	9708 (8764)	2.43 (2.19)	1.6 (1.6)	96.53 (96.87)	91.01 (91.87)
E30:Majority Vote	78.5 (78.9)	22.0 (22.3)	5232 (4469)	1.31 (1.12)	2.1 (2.2)	98.03 (98.32)	95.63 (96.24)
E30:MB	78.1 (78.1)	21.6 (21.5)	3375 (2450)	0.84 (0.61)	2.6 (3.2)	98.71 (99.06)	97.25 (98.02)
E30:MBME	77.8 (77.8)	21.2 (21.3)	2577 (2407)	0.64 (0.6)	3.1 (3.2)	98.98 (99.07)	98.04 (98.07)
E30:CR 2	78.0 (77.7)	21.5 (21.1)	2578 (1221)	0.64 (0.31)	3.1 (5.3)	99.02 (99.53)	97.86 (99.02)
E30:CR 5	78.0 (77.4)	21.5 (20.8)	1831 (670)	0.46 (0.17)	3.9 (8.8)	99.32 (99.75)	98.43 (99.43)
E30:CR 10	77.9 (77.1)	21.4 (20.6)	1517 (491)	0.38 (0.12)	4.5 (11.5)	99.44 (99.82)	98.67 (99.57)
R20:Replace	75.7 (75.7)	19.1 (19.1)	4171.0 (4183.8)	1.04 (1.05)	2.1 (2.1)	98.41 (98.41)	96.71 (96.7)
R20:Majority Vote	75.4 (75.4)	18.9 (18.8)	2690.2 (2696.6)	0.67 (0.67)	2.8 (2.7)	98.95 (98.94)	97.99 (97.98)
R20:MB	74.0 (74.6)	17.5 (18.0)	1662.8 (1421.2)	0.42 (0.36)	3.6 (4.2)	99.34 (99.44)	98.81 (98.97)
R20:MBME	73.5 (74.3)	16.9 (17.8)	1480.6 (1421.8)	0.37 (0.36)	3.9 (4.1)	99.4 (99.43)	98.97 (98.97)
R20:CR 2	73.2 (73.6)	16.7 (17.0)	1205.6 (718.2)	0.3 (0.18)	4.5 (6.9)	99.52 (99.71)	99.15 (99.48)
R20:CR 5	72.3 (72.2)	15.8 (15.7)	768.0 (344.0)	0.19 (0.09)	6.1 (12.4)	99.69 (99.86)	99.46 (99.75)
R20:CR 10	71.9 (71.0)	15.4 (14.4)	603.8 (218.4)	0.15 (0.05)	7.4 (17.5)	99.76 (99.91)	99.57 (99.84)
E20:Replace	78.5 (78.4)	22.0 (21.8)	6191 (5239)	1.55 (1.31)	1.9 (2.0)	97.74 (98.1)	94.47 (95.3)
E20:Majority Vote	78.3 (78.0)	21.8 (21.4)	4295 (3258)	1.07 (0.81)	2.3 (2.6)	98.37 (98.77)	96.46 (97.32)
E20:MB	77.9 (77.2)	21.3 (20.6)	2700 (1825)	0.68 (0.46)	3.0 (3.8)	98.96 (99.29)	97.86 (98.58)
E20:MBME	77.6 (77.2)	21.1 (20.6)	2183 (1825)	0.55 (0.46)	3.4 (3.8)	99.13 (99.29)	98.39 (98.58)
E20:CR 2	77.8 (76.6)	21.3 (20.1)	1999 (879)	0.5 (0.22)	3.7 (6.7)	99.23 (99.66)	98.41 (99.32)
E20:CR 5	77.7 (75.9)	21.2 (19.4)	1383 (448)	0.35 (0.11)	4.8 (11.8)	99.47 (99.83)	98.87 (99.63)
E20:CR 10	77.7 (75.1)	21.2 (18.6)	1101 (266)	0.28 (0.07)	5.8 (18.4)	99.58 (99.9)	99.08 (99.78)
R10:Replace	71.3 (71.3)	14.7 (14.8)	1958.4 (1952.6)	0.49 (0.49)	2.9 (2.9)	99.22 (99.22)	98.63 (98.63)
R10:Majority Vote	71.2 (71.2)	14.7 (14.6)	1481.4 (1489.6)	0.37 (0.37)	3.5 (3.5)	99.4 (99.4)	98.98 (98.98)
R10:MB	69.0 (70.0)	12.5 (13.4)	991.4 (865.4)	0.25 (0.22)	4.1 (4.9)	99.59 (99.65)	99.35 (99.42)
R10:MBME	68.9 (70.1)	12.4 (13.6)	944.8 (878.4)	0.24 (0.22)	4.3 (4.9)	99.61 (99.64)	99.38 (99.41)
R10:CR 2	67.9 (69.2)	11.4 (12.6)	703.6 (424.2)	0.18 (0.11)	5.0 (8.4)	99.71 (99.83)	99.54 (99.72)
R10:CR 5	66.2 (66.9)	9.6 (10.4)	393.4 (161.8)	0.1 (0.04)	7.1 (17.1)	99.84 (99.93)	99.75 (99.89)
R10:CR 10	65.3 (64.9)	8.8 (8.3)	282.6 (89.0)	0.07 (0.02)	8.7 (24.4)	99.88 (99.96)	99.82 (99.94)
E10:Replace	76.1 (77.0)	19.5 (20.4)	2468 (2216)	0.62 (0.55)	3.0 (3.3)	99.04 (99.17)	98.12 (98.18)
E10:Majority Vote	75.9 (76.1)	19.3 (19.5)	2417 (1954)	0.6 (0.49)	3.0 (3.5)	99.06 (99.25)	98.18 (98.45)
E10:MB	75.3 (75.2)	18.8 (18.7)	1557 (1118)	0.39 (0.28)	4.0 (5.2)	99.38 (99.56)	98.89 (99.18)
E10:MBME	75.2 (75.2)	18.7 (18.7)	1533 (1118)	0.38 (0.28)	4.0 (5.2)	99.38 (99.56)	98.92 (99.18)
E10:CR 2	75.3 (74.3)	18.7 (17.8)	1118 (503)	0.28 (0.13)	5.2 (9.8)	99.55 (99.8)	99.22 (99.63)
E10:CR 5	75.2 (72.7)	18.6 (16.2)	700 (165)	0.18 (0.04)	7.7 (25.5)	99.72 (99.93)	99.51 (99.88)
E10:CR 10	75.2 (70.7)	18.6 (14.2)	515 (71)	0.13 (0.02)	10.1 (50.8)	99.79 (99.97)	99.64 (99.95)

Table G.7: Full results of all experiments on **ObjectNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing **and incorporating soft labels**. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote.

Strategy (Budget %)	**Acc** (%)	**ΔAcc** (%)	Σ NF	NFR (%)	PF / NF	**Avg. BTC**	**Avg. BEC**
100:Oracle	50.5 (50.5)	42.6 (42.6)	0 (0)	0.0 (0.0)	NaN (NaN)	100.0 (100.0)	99.99 (99.99)
R100:Replace	31.9 (31.9)	24.0 (24.0)	16669 (16669)	5.62 (5.62)	1.3 (1.3)	72.65 (72.65)	92.61 (92.61)
R100:Majority Vote	29.6 (29.6)	21.6 (21.6)	4690 (4690)	1.58 (1.58)	1.9 (1.9)	89.99 (89.99)	98.02 (98.02)
E100:MB	29.1 (29.6)	21.2 (21.7)	2477 (1984)	0.83 (0.67)	2.6 (3.0)	94.46 (95.55)	98.96 (99.16)
E100:MBME	28.6 (29.1)	20.6 (21.1)	1599 (1774)	0.54 (0.6)	3.4 (3.2)	95.86 (95.83)	99.34 (99.26)
E100:CR 2	29.0 (29.2)	21.0 (21.2)	1876 (931)	0.63 (0.31)	3.1 (5.2)	95.92 (97.9)	99.21 (99.61)
E100:CR 5	28.8 (28.7)	20.8 (20.7)	1372 (543)	0.46 (0.18)	3.8 (8.1)	97.18 (98.86)	99.41 (99.77)
E100:CR 10	28.7 (28.4)	20.8 (20.5)	1084 (422)	0.37 (0.14)	4.6 (10.0)	97.82 (99.16)	99.54 (99.82)
R50:Replace	30.7 (30.7)	22.7 (22.7)	7583.8 (7588.8)	2.56 (2.56)	1.6 (1.6)	86.63 (86.61)	96.69 (96.69)
R50:Majority Vote	28.6 (28.7)	20.7 (20.7)	3281.6 (3291.2)	1.11 (1.11)	2.2 (2.2)	93.01 (93.01)	98.62 (98.62)
R50:MB	27.8 (28.7)	19.9 (20.7)	1676.6 (1410.0)	0.56 (0.48)	3.2 (3.7)	96.13 (96.83)	99.3 (99.41)
R50:MBME	26.9 (28.3)	18.9 (20.3)	1280.4 (1360.8)	0.43 (0.46)	3.7 (3.8)	96.77 (96.89)	99.48 (99.43)
R50:CR 2	27.5 (28.1)	19.6 (20.2)	1165.4 (663.6)	0.39 (0.22)	4.1 (6.6)	97.3 (98.51)	99.52 (99.72)
R50:CR 5	27.2 (27.3)	19.2 (19.4)	780.2 (361.4)	0.26 (0.12)	5.6 (10.9)	98.26 (99.23)	99.67 (99.85)
R50:CR 10	27.0 (26.7)	19.1 (18.8)	599.8 (253.0)	0.2 (0.09)	6.9 (14.8)	98.7 (99.47)	99.75 (99.89)
E50:Replace	31.0 (31.4)	23.1 (23.4)	7882 (7329)	2.66 (2.47)	1.5 (1.6)	86.26 (87.93)	96.53 (96.75)
E50:Majority Vote	29.5 (29.4)	21.5 (21.5)	3895 (3397)	1.31 (1.14)	2.0 (2.2)	91.73 (93.32)	98.36 (98.55)
E50:MB	28.9 (30.0)	20.9 (22.1)	2079 (1610)	0.7 (0.54)	2.9 (3.5)	95.23 (96.66)	99.13 (99.31)
E50:MBME	28.2 (29.5)	20.3 (21.5)	1452 (1534)	0.49 (0.52)	3.6 (3.6)	96.18 (96.75)	99.41 (99.35)
E50:CR 2	28.7 (29.6)	20.8 (21.7)	1506 (763)	0.51 (0.26)	3.6 (6.3)	96.64 (98.41)	99.37 (99.67)
E50:CR 5	28.5 (29.0)	20.6 (21.1)	1049 (414)	0.35 (0.14)	4.6 (10.5)	97.83 (99.21)	99.55 (99.82)
E50:CR 10	28.4 (28.7)	20.5 (20.7)	828 (309)	0.28 (0.1)	5.6 (13.4)	98.35 (99.43)	99.64 (99.86)
R30:Replace	29.0 (29.0)	21.0 (21.1)	4070.6 (4074.0)	1.37 (1.37)	2.0 (2.0)	92.19 (92.21)	98.26 (98.26)
R30:Majority Vote	27.3 (27.3)	19.4 (19.4)	2346.6 (2348.8)	0.79 (0.79)	2.5 (2.5)	94.91 (94.88)	99.02 (99.02)
R30:MB	25.9 (27.3)	18.0 (19.4)	1185.8 (1024.8)	0.4 (0.35)	3.8 (4.5)	97.12 (97.58)	99.52 (99.58)
R30:MBME	25.1 (27.1)	17.2 (19.2)	1008.4 (988.6)	0.34 (0.33)	4.2 (4.6)	97.41 (97.67)	99.59 (99.59)
R30:CR 2	25.4 (26.5)	17.5 (18.5)	782.2 (482.8)	0.26 (0.16)	5.2 (8.1)	98.05 (98.87)	99.68 (99.8)
R30:CR 5	24.6 (25.2)	16.6 (17.2)	476.2 (236.2)	0.16 (0.08)	7.5 (14.6)	98.8 (99.45)	99.81 (99.9)
R30:CR 10	24.4 (24.5)	16.4 (16.5)	331.6 (154.2)	0.11 (0.05)	10.2 (20.9)	99.17 (99.65)	99.86 (99.93)
E30:Replace	29.0 (30.0)	21.0 (22.1)	4316 (3761)	1.45 (1.27)	1.9 (2.1)	91.75 (93.51)	98.14 (98.35)
E30:Majority Vote	28.2 (28.2)	20.3 (20.2)	2970 (2359)	1.0 (0.79)	2.3 (2.6)	93.54 (95.37)	98.76 (99.0)
E30:MB	27.8 (28.9)	19.9 (20.9)	1565 (1015)	0.53 (0.34)	3.4 (4.8)	96.24 (97.78)	99.35 (99.57)
E30:MBME	26.9 (28.9)	18.9 (20.9)	1280 (1015)	0.43 (0.34)	3.7 (4.8)	96.64 (97.78)	99.48 (99.57)
E30:CR 2	27.7 (28.1)	19.7 (20.1)	1074 (496)	0.36 (0.17)	4.4 (8.5)	97.42 (98.9)	99.55 (99.79)
E30:CR 5	27.4 (26.9)	19.4 (19.0)	689 (230)	0.23 (0.08)	6.2 (16.3)	98.43 (99.52)	99.71 (99.9)
E30:CR 10	27.1 (25.9)	19.2 (18.0)	504 (140)	0.17 (0.05)	8.1 (24.8)	98.91 (99.71)	99.79 (99.94)
R20:Replace	27.0 (27.1)	19.1 (19.1)	2453.8 (2436.4)	0.83 (0.82)	2.4 (2.5)	94.82 (94.84)	98.97 (98.98)
R20:Majority Vote	25.8 (25.8)	17.8 (17.9)	1658.8 (1663.4)	0.56 (0.56)	3.0 (3.0)	96.17 (96.16)	99.32 (99.32)
R20:MB	23.8 (25.3)	15.8 (17.3)	887.2 (744.0)	0.3 (0.25)	4.3 (5.3)	97.74 (98.15)	99.64 (99.7)
R20:MBME	23.5 (25.5)	15.6 (17.5)	793.8 (736.0)	0.27 (0.25)	4.6 (5.4)	97.93 (98.18)	99.68 (99.7)
R20:CR 2	22.6 (24.3)	14.7 (16.4)	530.0 (339.0)	0.18 (0.11)	6.2 (10.0)	98.57 (99.14)	99.79 (99.86)
R20:CR 5	21.5 (22.7)	13.5 (14.8)	304.2 (160.8)	0.1 (0.05)	9.3 (18.1)	99.14 (99.6)	99.88 (99.93)
R20:CR 10	20.9 (21.5)	12.9 (13.6)	215.6 (98.8)	0.07 (0.03)	12.1 (26.6)	99.4 (99.74)	99.91 (99.96)
E20:Replace	26.9 (29.3)	19.0 (21.3)	2588 (2149)	0.87 (0.72)	2.4 (2.8)	94.54 (96.08)	98.91 (99.07)
E20:Majority Vote	26.2 (27.5)	18.3 (19.6)	2390 (1742)	0.81 (0.59)	2.4 (3.1)	94.8 (96.61)	99.0 (99.26)
E20:MB	26.0 (27.7)	18.1 (19.7)	1169 (763)	0.39 (0.26)	3.9 (5.8)	97.01 (98.26)	99.53 (99.68)
E20:MBME	25.8 (27.7)	17.8 (19.7)	1131 (763)	0.38 (0.26)	3.9 (5.8)	97.06 (98.26)	99.54 (99.68)
E20:CR 2	26.0 (26.6)	18.1 (18.6)	670 (353)	0.23 (0.12)	6.0 (10.8)	98.14 (99.16)	99.73 (99.85)
E20:CR 5	26.0 (25.1)	18.0 (17.2)	369 (132)	0.12 (0.04)	10.1 (25.2)	98.97 (99.7)	99.85 (99.94)
E20:CR 10	25.9 (24.2)	17.9 (16.2)	248 (73)	0.08 (0.02)	14.4 (42.2)	99.34 (99.83)	99.9 (99.97)
R10:Replace	22.1 (22.1)	14.2 (14.2)	996.6 (997.2)	0.34 (0.34)	3.6 (3.6)	97.49 (97.5)	99.6 (99.6)
R10:Majority Vote	21.6 (21.5)	13.6 (13.5)	808.8 (820.4)	0.27 (0.28)	4.1 (4.1)	97.86 (97.83)	99.68 (99.67)
R10:MB	19.5 (21.0)	11.6 (13.1)	450.2 (371.2)	0.15 (0.13)	5.8 (7.6)	98.7 (98.95)	99.82 (99.85)
R10:MBME	19.1 (20.8)	11.1 (12.8)	446.4 (381.6)	0.15 (0.13)	5.6 (7.2)	98.69 (98.92)	99.83 (99.85)
R10:CR 2	17.7 (19.8)	9.8 (11.9)	287.2 (177.2)	0.1 (0.06)	7.3 (13.4)	99.11 (99.48)	99.89 (99.93)
R10:CR 5	15.6 (17.6)	7.7 (9.7)	151.6 (75.4)	0.05 (0.03)	10.4 (24.9)	99.5 (99.77)	99.94 (99.97)
R10:CR 10	14.6 (15.8)	6.7 (7.9)	94.6 (36.0)	0.03 (0.01)	14.1 (41.8)	99.68 (99.89)	99.96 (99.99)
E10:Replace	23.7 (25.8)	15.7 (17.9)	996 (683)	0.34 (0.23)	3.9 (5.9)	97.5 (98.46)	99.6 (99.72)
E10:Majority Vote	23.7 (25.4)	15.7 (17.5)	996 (663)	0.34 (0.22)	3.9 (5.9)	97.5 (98.48)	99.6 (99.72)
E10:MB	22.7 (24.4)	14.8 (16.5)	696 (364)	0.23 (0.12)	4.9 (9.4)	98.08 (99.03)	99.72 (99.85)
E10:MBME	22.7 (24.4)	14.8 (16.5)	696 (364)	0.23 (0.12)	4.9 (9.4)	98.08 (99.03)	99.72 (99.85)
E10:CR 2	20.7 (22.7)	12.8 (14.8)	427 (176)	0.14 (0.06)	6.6 (16.6)	98.68 (99.52)	99.83 (99.93)
E10:CR 5	18.2 (19.4)	10.3 (11.4)	197 (50)	0.07 (0.02)	10.7 (43.4)	99.32 (99.86)	99.92 (99.98)
E10:CR 10	17.2 (15.9)	9.2 (7.9)	122 (28)	0.04 (0.01)	15.0 (53.5)	99.57 (99.92)	99.95 (99.99)

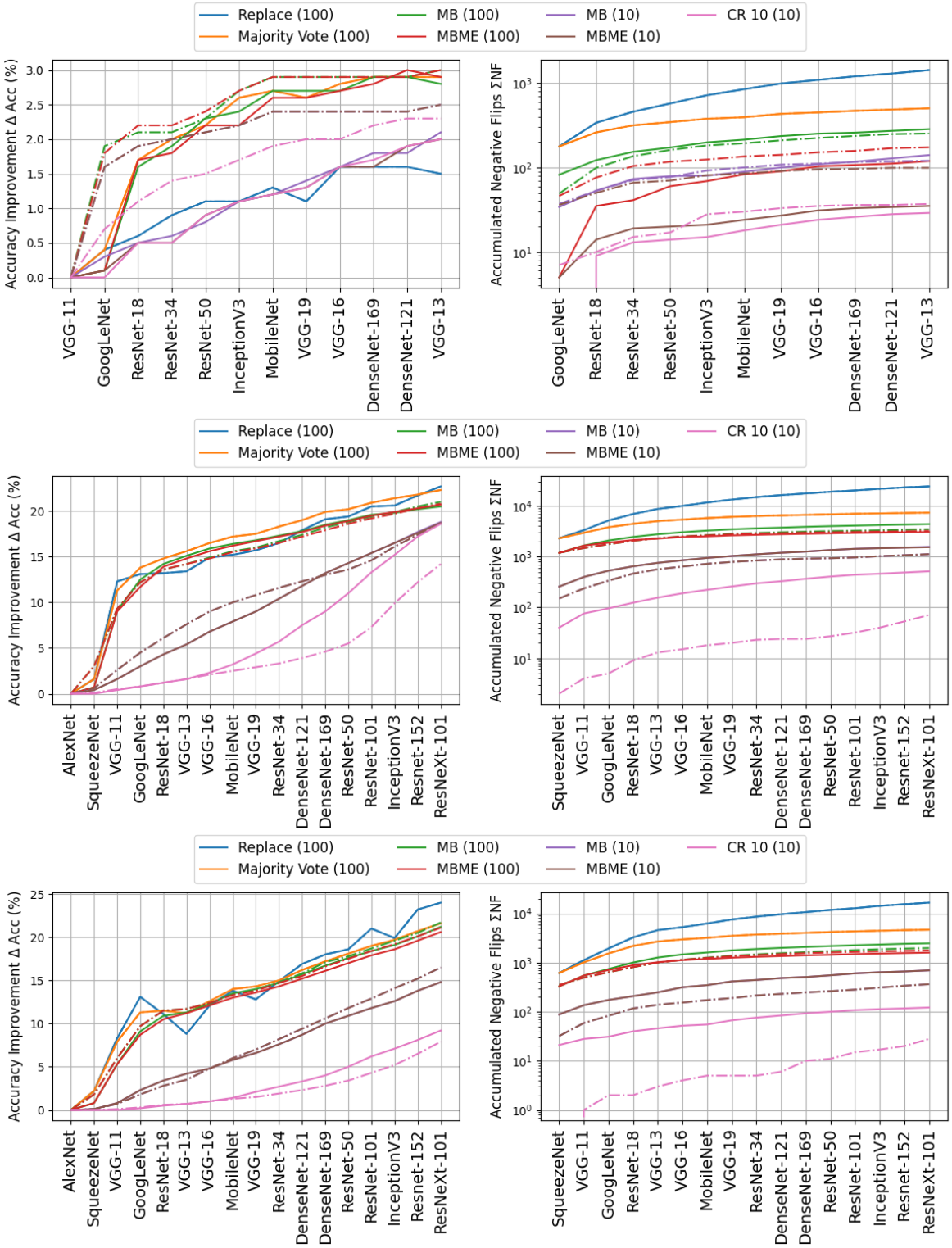


Figure G.13: Comparative temporal evolution plots for experiments incorporating uncalibrated softmax labels. Cifar10 (top), ImageNet (middle) and ObjectNet(bottom). Solid lines represent using hard labels with only diagonal estimated for confusion matrix and dashed lines reflect results using soft labels and estimating full confusion matrix with Laplace smoothing.

List of Figures

1.1 A schematic illustration of an *i.i.d.* learning scenario with $P_{\text{train}}(C_1, C_2) = P_{\text{test}}(C_1, C_2)$ and a *covariate shift* in the distribution of the latent variables with $P_{\text{train}}(C_1, C_2) \neq P_{\text{test}}(C_1, C_2)$. 3

1.2 Various examples of expected out-of-distribution generalizations in the latent variables such as recombination, extrapolation or interpolation along different variable axes. . . 4

1.3 An example of continual learning, where different tasks are presented only consecutively during the learning phase. Naive training of standard neural networks causes severe catastrophic forgetting of previously learned skills (red line), while humans can incrementally learn new skills without severe forgetting (black line). 5

1.4 Beyond traditional distribution shifts, we will also study the problem of how to deal with a shift in the output distribution. Such shifts could happen when an integrated ML model within a bigger prediction pipeline is being updated with presumably better models. 12

2.1 While in principle we would like to consider foot length and body height to be independent features of our model, they exhibit strong positive correlation in observed data. 15

2.2 **Left:** Latent traversals for the model with best DCI score (commonly used metric for measuring disentanglement [84]) among all 180 models trained on strongly correlated ($\sigma = 0.2$) Shapes3D data (A). The traversals in latent dimensions 5 and 9 (highlighted in black) encode a mixture of azimuth and object size, reflecting the main correlation line of the joint distribution and a smaller, locally orthogonal one. **Right:** A heat map of the GBT feature importance matrix of this model indicates an entanglement of azimuth and object size encoded into both latent codes. 22

2.3 Stronger correlations in the train data lead to substantially elevated unfairness scores for the correlated pair of factors across the three correlated dataset variants. 24

2.4 Generalization to out-of-distribution (OOD) test data. **Top:** Reconstructions of observations the model has never seen during training. **Bottom:** Latent space distribution of the two entangled dimensions. Circles without edges represent latent encodings of (correlated) training examples. Circles with edges are OOD examples that break the correlation pattern. 25

2.5 **Top left:** We show the disentanglement (DCI score) for models trained on Shapes3D *with weak supervision* when object size and azimuth are correlated with different strengths. Weak supervision helps to recover improved, often perfect, disentanglement. **Top middle:** Unfairness scores between correlated FoV are drastically reduced when using weak supervision (see scale) across all correlation strengths. **Top right:** Latent dimensions of the best DCI model with strong correlation ($\sigma = 0.2$) when using weak supervision. Representations are axis-aligned with respect to both of the correlated ground truth factors. **Lower:** Latent traversals of this model trained on strong correlation ($\sigma = 0.2$). . 28

3.1 Example tasks from the task generators provided in the benchmark. The goal shape is visualized in opaque red and the blocks in blue. 34

3.2 Example of do-interventions on exposed variables in `CausalWorld`. 34

3.3 Key components for generic training and evaluation of RL agents. Left: A learning curriculum which is composed of various intervention actors that decide which variables to intervene on (for a valid intervention, values need to be in the allowed training space (ATS)). Right: Evaluation protocols are shown which may intervene on variables at episode resets or within episodes (for a valid intervention, values need to be in the evaluation space (ES)). Middle: we represent the ATS and ES, where each intervention results in one point in the spaces. As shown ATS and ES may intersect, eg. if the protocols are meant to evaluate in-distribution generalization. 38
3.4 Fractional success curves averaged over five random seeds for the tasks and learning algorithms specified above, under three different training curricula: (0) no curriculum, (1) goal position and orientation randomization in space **A** every episode and (2) a curriculum where we intervene on all variables in space **A** simultaneously every episode. 42
3.5 Evaluation scores for pushing baselines. Each protocol was evaluated for 200 episodes and each bar is averaged over five models with different random seeds. The variables listed under each protocol are sampled from the specified space at the start of every episode while all other variables remain fixed [bp block pose, bm block mass, bs block size, gp goal pose, ff floor friction]. 43

4.1 **Left**: Images from the simulated dataset and from the real-world setup. **Right**: Factors of variation in the proposed dataset. Values are linearly spaced in the specified intervals. Joint angles are in radians, cube positions in meters. 52
4.2 Latent traversals of a trained model that perfectly disentangles the dataset's FoVs. In each column, all latent variables but one are fixed. 54
4.3 Disentanglement metrics aggregating all hyperparameters except for unsupervised and weakly-supervised learning. 55
4.4 Rank correlations (Spearman) of ELBO, reconstruction loss, and the test error of a GBT classifier trained on 10,000 labelled data points with disentanglement metrics. The upper rank correlations correspond to the unsupervised models and the lower to the weakly supervised models. 55
4.5 Higher disentanglement corresponds to better generalization across all OOD1 scenarios, as seen from the transfer scores. The transfer score is computed as the mean absolute prediction error of ground truth factor values (lower is better). This correlation is particularly evident in the GBT case, whereas MLPs appear to exhibit better OOD1 transfer with very high disentanglement only. 58
4.6 Spearman rank correlations between transfer scores and disentanglement metrics. . . 58
4.7 As seen from transfer scores, disentanglement (DCI) affects generalization across the OOD2 scenarios only minimally. 59
4.8 Rank correlations between transfer scores on OOD2 scenarios and various disentanglement metrics. 59
4.9 Noise improves generalization across the OOD2 scenarios and less so for the OOD1 scenarios as seen from the transfer scores. Top row: Spearman rank correlation coefficients between transfer metrics and presence of noise in the input. 60
4.10 Zero-shot transfer to real-world observations of our models trained in simulation. Left: input; right: reconstruction. 60

5.1 **Overview of our experimental setup for investigating out-of-distribution generalization in downstream tasks.** (1) We train 240 β-VAEs on the robotic dataset from [3]. (2) We then train downstream policies to solve *object reaching* or *pushing*, using multiple random RL seeds per VAE. The input to a policy consists of the output of a pre-trained encoder and additional task-related observable variables. Crucially, the policy is only trained on a subset of the cube colors from the pre-training dataset. (3) Finally, we evaluate these policies on their respective tasks in four different scenarios: (a) in-distribution, i.e. with cube colors used in policy training; (b) OOD1, i.e. with cube colours previously seen by the encoder but OOD for the policy; (c) OOD2-sim, having cube colours also OOD to the encoder; (d) sim-to-real zero-shot on the real-world setup. 67
5.2 Top: Average training success, aggregated over *all* policies from the sweep (median, quartiles, 5th/95th percentiles). Bottom: Rank correlations between representation metrics and in-distribution reward (evaluated when the policies are fully trained), in the case without regularization. Correlations are color-coded in red (positive) or blue (negative) when statistically significant ($p < 0.05$), otherwise they are gray. 73
5.3 Correlations between training (in distrib.) and OOD rewards (p<0.05). 74
5.4 Rank correlations of representation properties with OOD1 and OOD2 reward on *object reaching* without regularization. Numbering when splitting metrics by FoV: (1) cube color; (2–4) joint angles; (5–7) cube position and rotation. Correlations are color-coded as described in Figure 5.2. 74
5.5 Rank correlations of generalization scores with OOD1 and OOD2 reward on *object reaching* without regularization. Numbering when splitting metrics by FoV: (1) cube color; (2–4) joint angles; (5–7) cube position and rotation. Correlations are color-coded as described in Figure 5.2. 75
5.6 **Box plots**: Fractional success on *object reaching* split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. L1 regularization in the first layer of the MLP policy has a positive effect on OOD1 and OOD2 generalization with minimal sacrifice in terms of training reward (see scale). **Correlation matrix**: Although we observe a mild correlation between some disentanglement metrics and OOD1 (but not OOD2) generalization, this does not hold when adjusting for representation informativeness. Correlations are color-coded as described in Figure 5.2. We use disentanglement metrics from [60, 61, 84, 160]. 76
5.7 Rank-correlations of various representation metrics with the zero-shot sim-to-real OOD2 reward for the task of *object reaching* on the real platform with a red cube (color-coded as described in Figure 5.2). 77
5.8 Training encoders with additive noise improves zero-shot sim-to-real generalization on *object reaching* 77
5.9 Zero-shot sim-to-real on *object reaching* with histograms of fractional success in the more challenging OOD2-real-{green,blue} scenario from 50 policies across 4 different goal positions. 78
5.10 We select pushing policies with high GS-OOD2-real score. When deployed on the real robot without fine-tuning, they succeed in pushing the cube to a specified goal position (transparent blue cube). 78

6.1 **Overview of discrete key-value bottleneck**. First, a pre-trained encoder maps the high-dimensional input to a lower-dimensional representation. Second, this representation is projected into C lower-dimensional heads. $C = 2$ is chosen for illustrative purposes. Each head is processed as input by one out of C separate key-value codebooks: In the codebook, the head is discretized by snapping to the closest key code, and the corresponding value code for that key is fetched. Third, the values from different key-value codebooks are combined and fed as input to the decoder for the final prediction. Keys and values can be of different dimensions in the proposed model. 84

6.2 Incremental training on non-overlapping datasets $\mathcal{D}_1 \rightarrow \mathcal{D}_2 \rightarrow \mathcal{D}_3 \rightarrow \mathcal{D}_4$ (dots) using (a) Linear Probe (LP), (b) 1-layer MLP (MLP), (c) key-value bottleneck with C = 1 and ground truth feature heads, (d) key-value bottleneck with C = 20 and randomly projected feature heads. Using a discrete bottleneck of 400 key-value pairs with fixed key codes, and locally updated value codes, alleviates catastrophic forgetting. Color domains indicate one out of the 8 possible classes predicted by the model. 91

6.3 The key-value bottleneck architecture successfully learns and integrates new information in a challenging 5-split CIFAR10 class-incremental (CI) learning setting. Our method (green line) reduces catastrophic forgetting and yields strong final performance across various pre-trained frozen backbone architectures including (a) ConvMixer; (b) ResNet50 ; (c) ViT B/32 with CLIP, (d) ResNet50w2 with SwAV; (e) ResNet50 with DINO. At the end of the training, the proposed method outperforms the reported performance of the SDM method using the same pre-trained ConvMixer. Standard probe tuning exhibits major forgetting in this scenario (red, brown, purple curve). 92

6.4 We demonstrate that the proposed key-value bottleneck method maintains a constant loss measure, directly translating to improved generalization when training class incrementally, as opposed to standard approaches which can over-fit. 94

6.5 Assessing the role of individual model components starting from the base model architecture in Figure 6.3 b. Reducing the dimensionality of keys has a positive effect (a); increasing the number of pairs and codebooks enhances generalization performance (b + c); finally, key-value bottlenecks keys can be initialized from a different dataset without major drop in overall performance (d). 95

7.1 **Overview of our proposed Bayesian approach to the Prediction Update Problem.** Starting from a uniform prior, we maintain a posterior distribution $p(y = k|\hat{y}^{0:t})$ (*middle*) over the unknown true label y of an unlabelled sample which takes the predictions $\hat{y}^{0:t}$ from new ML models C^t (*top*) arriving over time $t = 0, ..., T$ into account. Given a limited compute budget B^t, we re-evaluate those samples with highest posterior label entropy S^t at each time step, e.g., the example shown is first selected in time step $t = 3$ after the initial annotation at $t = 0$. We then consider different strategies for deciding whether to update the stored prediction (*bottom*) based on our changed beliefs. Note that the non-probabilistic baselines "Replace" (always update to last prediction) and "Majority Vote" (resolve ties by using the last prediction) incorrectly update the stored prediction from "truck" to "deer" in step $t = 4$. Our strategies (MB, MBME, CR-10) which rely on the estimated label posterior, on the other hand, avoid such a *negative flip*, which is one of our key goals. 104

7.2 Comparison of prediction-update strategies across different budgets after $T = 16$ prediction-update steps for ImageNet → ImageNet (left) and ImageNet → ObjectNet (right). Dashed lines correspond to the ablation using a random selection strategy. 112

7.3 Temporal evolution of several key metrics for ImageNet → ImageNet over $T = 16$ prediction-update steps for a subset of strategies and budgets. 113
7.4 Temporal evolution of key metrics for ImageNet → ObjectNet over $T = 16$ prediction-update steps for a subset of strategies and budgets. 116
7.5 Under smaller accuracy gains (experiments on CIFAR-10), we outperform the Replace baseline w.r.t. desideratum (i). Temporal evolution for CIFAR-10 over $T = 11$ prediction-update steps for a subset of strategies and budgets. Our approaches achieve competitive overall accuracy gains (even at 10% of the budget) while introducing only a fraction of negative flips. 117
7.6 Experiment on ImageNet with random model sequence suggesting robustness of our approach to situations where an ordering by performance of may not be available. . . 118

B.1 Probability distributions for sampling training data in the correlated pair of FoVs in the respective datasets (A, B, C, D, E) considering correlation strengths of $\sigma = 0.2$, $\sigma = 0.4$, $\sigma = 0.7$ and $\sigma = \infty$, the uncorrelated limit (from left to right). 141
B.2 We show latent traversals (left) of the best DCI score model among all 180 trained models with weak correlation ($\sigma = 0.7$) in object size and azimuth. The traversals in latent codes 2, 6, and 7 (highlighted in black), suggest that these dimensions encode no mixture of azimuth and object size compared to the models with stronger correlation. This is supported by the GBT feature importance matrix of this model (right). 142
B.3 Disentangled representations trained on correlated data are anti-correlated with higher fairness properties. The plots show the mean unfairness scores between the correlated factors with decreasing correlation strength for Shapes3D (A), dSprites (B) and MPI3D-real (C). 144
B.4 Standard global disentanglement metrics evaluated on the correlated training set showing no clear trend for different correlation strengths. 145
B.5 Standard global disentanglement metrics evaluated on the uncorrelated (uc) dataset set showing no clear trend for different correlation strengths. 145
B.6 When disentanglement metrics are only evaluated regarding the 2 correlated FoV we can observe the still persisting entanglement in the latents using DCI score. 146
B.7 Generalization capabilities towards out-of-distribution test data. Latent traversals from an observations the model has never seen during training. The starting point corresponds to a factor configuration in point number 2 from Figure 2.4 Shown are the results of the model with highest DCI score among all 180 trained models on Shapes3d (A) with a very restricted correlation strength $\sigma = 0.2$ in object size and azimuth 146
B.8 Latent space distribution of the two entangled dimensions of the best DCI model in Shapes3d (E) with $\sigma = 0.2$ (top row), in Shapes3d (E) with $\sigma = 0.4$ (bottom row). Latent codes sampled from correlated observations (circle without edge) and (2) latent codes sampled with an object color-azimuth configuration not encountered during training (squares with black edge). Each column shows the ground truth values of the two correlated factors by color. 147
B.9 Latent space distribution of the two entangled dimensions of the best DCI model in Shapes3d (D) with $\sigma = 0.2$ (top row) and in Shapes3d (D) with $\sigma = 0.4$ (bottom row). Latent codes sampled from correlated observations (circle without edge) and (2) latent codes sampled with an object color-size configuration not encountered during training (squares with black edge). Each column shows the ground truth values of the two correlated factors by color. 148

B.10 Latent space distribution of the two entangled dimensions of the best DCI model in Shapes3D (A). Latent codes sampled from correlated observations (circle without edge) and (2) latent codes sampled with an object size-azimuth configuration not encountered during training (squares with black edge). Left column shows the latent space of the two correlated factors by color. Middle and right column show the fast adapted space using linear regression and 100 or 1000 labels respectively. 150
B.11 Left: For the weakly supervised scenario using correlated observational data trained models on Shapes3D (A), (D) and (E) correlating object color and azimuth learn consistently improved, often perfect, disentangled representation across all correlation strengths. Right: Latent dimensions of a best DCI model trained on strongly correlated observational data. Representations are strongly axis-aligned with respect to both of the correlated variables ground truth values (right). 151
B.12 Left: For the weakly supervised scenario with intervening capabilities (Scenario I-1) trained models on Shapes3D (A), (D) and (E) correlating object color and azimuth learn consistently improved, often perfect, disentangled representation across all correlation strengths. Right: Latent dimensions of a best DCI model with strong correlation ($\sigma = 0.2$). Representations are strongly axis-aligned with respect to both of the correlated variables ground truth values. 152
B.13 DCI scores and latent spaces show strong disentanglement using weak supervision with intervening capabilities (Scenario I-2) - even under the stronger assumption that sampling of observation pairs follow its causal generative model. We show the learned latent space encoding of the two correlated factors of variation for a model on Shapes3D with $\sigma = 0.1$. 153

C.1 Example "pixel" mode observations returned at each step of the environment. 156
C.2 Structured observation description. For the scene features, all the blocks feature vector are concatenated first. Following that the partial goals feature vector are concatenated in the same order. Lastly, if there is any obstacles/ fixed blocks, their feature vectors are concatenated at the end following the same description as the partial goal features. . . 156
C.3 The TriFinger platform. 157
C.4 An example of evaluating generalization across the various axes in `CausalWorld` across the various axes using the previously discussed protocols. Here we compare the mean last fractional success of two agents trained on the Pushing task under different curricula using PPO. 160
C.5 Evaluation scores, for pushing, picking, pick and place and stacking2 baselines, from top to bottom respectively. Each protocol was evaluated for 200 episodes and each bar is averaged over five models with different random seeds [bp block pose, bm block mass, bs block size, gp goal pose, ff floor friction]. 161

D.1 Schemes of the encoder (top) and decoder (bottom) architectures. In both schemes, information flows left to right. Blue blocks represent convolutional layers: those labeled “conv” have 5x5 kernels and stride 2, while those labeled “1x1” have 1x1 kernels. Each orange block represents a pair of residual blocks (implementation details of a residual block are provided in Table D.2). Green blocks in the encoder represent average pooling with stride 2, and those in the decoder denote bilinear upsampling by a factor of 2. Red blocks represent fully-connected layers. The block labeled “norm” indicates layer normalization. Dashed lines denote tensor reshaping. 165

D.2 Feasible states of the 2nd and 3rd DoF when the angle of the 1st DoF is 0. Angles are in radians. . . . 166
D.3 Density of feasible states of 2nd and 3rd DoF over the whole training dataset. Darker shades of blue indicate regions of higher density. Angles are in radians. . . . 166
D.4 Samples generated by a trained model. This model was selected based on the ELBO. . 167
D.5 Input reconstructions by a trained model. This model was selected based on the ELBO. Image inputs are on odd columns, reconstructions on even columns. . . . 167
D.6 Latent traversals for a model with low DCI score (0.15) in (a), medium DCI score (0.5) in (b), and high DCI score (1.0) in (c). . . . 168
D.7 Scatter plots of unsupervised metrics (left: ELBO; right: reconstruction loss) vs disentanglement (top: MIG; bottom: DCI) for 1,080 trained models, color-coded according to supervision. Each point represents a trained model. . . . 169
D.8 Transfer metric in OOD2-A (top) and OOD2-B (bottom) settings, decomposed according to the factor of variation and presence of input noise. . . . 170
D.9 Reconstructions of real-world images (OOD2-B) for a model with low DCI score (0.15) in (a), medium DCI score (0.5) in (b), and high DCI score (1.0) in (c). . . . 171
D.10 Reconstructions of simulated images with encoder out-of-distribution colors (OOD2-A) for a model with low DCI score (0.15) in (a), medium DCI score (0.5) in (b), and high DCI score (1.0) in (c). . . . 172

E.1 Rank correlations between metrics and in-distribution reward, with and without adjusting for informativeness. Correlations are color-coded as described in Figure 5.2. . . . 175
E.2 Rank correlations between metrics and OOD reward, with and without adjusting for informativeness. Correlations are color-coded as described in Figure 5.2. . . . 176
E.3 Fractional success on *object reaching* (**top**) and *pushing* (**bottom**), split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. Results for *object reaching* are also reported in Figure 5.6 in Section 5.4.2. . . . 177
E.4 Sample efficiency analysis for *object reaching*. Rank correlations of rewards with relevant metrics along multiple time steps. Correlations are color-coded as described in Figure 5.2. 178
E.5 Sample efficiency analysis for *pushing*. Rank correlations of rewards with relevant metrics along multiple time steps. Correlations are color-coded as described in Figure 5.2. . . 179
E.6 Testing policies for *object reaching* under the same in-distribution, OOD1, and OOD2 evaluation protocols regarding object color in simulation, but replacing the cube with a sphere, which was never used in training. . . . 180
E.7 Transferring policies for *object reaching* to the real robot setup without any fine-tuning on a green sphere (unseen shape *and* color). Correlations are color-coded as described in Figure 5.2. . . . 181

F.1 Analysing the effect of class-granularity during class incremental learning. In line with our theory, we observe *across all possible class-split sizes* the appealing learning property that the bottleneck yields the same performance at the end of training. . . . 188
F.2 Applying the FlyModel from [274] to the CIFAR10 learning setup with each split being repeated 2000 epochs. The model with hyperparameters from [271] learns new classes at the beginning of a new split but quickly drops to some lower-level performance until the end of the phase. Eventually, the model cannot integrate any new information and remains at a very low level. . . . 189

F.3 Analysing the **key codes** on the **ResNet50 backbone.** Left Plot: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle Plot. The key code distribution on the target test dataset. Keys exhibit increased clustering, which reflects the smaller feature diversity present in CIFAR10. Right Plot: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many different input samples, while the bottleneck retains some unused capacity. 190
F.4 Additional baseline experiments: Rerunning the CIFAR10 experiment on all five backbones using the same Linear Probe as in Figure 6.3. In contrast to these baselines already shown in the main paper, we now applied weight decay on the head parameters across four orders of magnitude with $w_d \in [10^{-2}, 10^{-4}, 10^{-6}]$. The learned models are unable to effectively integrate novel class information using this additional form of ridge regularization. 190
F.5 Additional baseline experiments: Rerunning the CIFAR10 experiment on all five backbones using the same MLP head as in Figure 6.3. In contrast to these baselines already shown in the main paper, we now applied weight decay on the head parameters across four orders of magnitude with $w_d \in [10^{-2}, 10^{-4}, 10^{-6}]$. The learned models are unable to effectively integrate novel class information using this additional form of ridge regularization. 191
F.6 Additional class-incremental learning experiment on CIFAR100: Using 50 splits, each consisting of 2 classes, we tested our model on the ResNet-50 and CLIP-Backbone architectures. Our model achieved final accuracies of 54.7% and 64% respectively, compared to the 43% achieved by SDMLP on the same task [271]. Keys were initialized on the CIFAR10 dataset for this experiment. 192
F.7 Additional class-incremental learning experiment on ImageNet: We evaluated our model on 500 splits, each comprising 2 classes, using the CLIP encoder. The final accuracy reached 49.9% for the i.i.d. setting and 49.0% for the class-incremental setting. To the best of our knowledge, no other method has been tested on such a large-scale class-incremental learning setting, encompassing 1000 classes and 500 splits. Keys were initialized on the CIFAR100 dataset for this experiment. 193
F.8 We have conducted additional experiments to assess the models' robustness. Specifically, we evaluated the performance of the models analyzed in Figures 8-11, which employ different backbones, when subjected to additive Gaussian noise spanning three orders of magnitude. We applied the noise either to the original input pixels (left) or the backbone embeddings (right). Results indicate that the model's performance remains largely unaffected by the noise and only starts to deteriorate substantially when the noise level exceeds $\alpha = 0.1$, at which point the image becomes nearly unrecognizable upon visual inspection. From a theoretical per 194

F.9 Analysing the **key codes** on the **CLIP-pre-trained ViT backbone**. Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Although the coverage appears more clustered, the keys still cover the majority of the target test set, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many different input samples, while the bottleneck retains some unused capacity. 196
F.10 Analysing the **key codes** on the **DINO-pre-trained backbone**. Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Although the coverage appears more clustered, the keys still cover the majority of the target test set, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many different input samples, while the bottleneck retains some unused capacity. 196
F.11 Analysing the **key codes** on the **SwAV-pre-trained backbone.** Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Keys exhibit increased clustering, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many input samples, while the bottleneck retains some unused capacity. 197
F.12 Analysing the **key codes** on the **ConvMixer backbone** from [271]. Left: UMAP of the codebook embedding heads from the key initialization dataset (light blue) with the discrete key codes (black dots) as obtained by EMA. Keys are broadly covering the data manifold. Middle. The key code distribution on the target test dataset. Keys exhibit increased clustering, which reflects the smaller feature diversity present in CIFAR10. Right: Key code utilization across all codebooks over the course of class-incremental training (2000 epochs). The majority of keys is being shared among many input samples, while the bottleneck retains some unused capacity. 197

G.1 Temporal evolution of accuracy improvement and accumulated negative flips on ImageNet for two scenarios where the models C^t do not exhibit improving performance, but instead arrive in (a) random or (b) adversarial order. As can be seen, our methods are robust in both these cases while introducing much fewer negative flips than the non-probabilistic baselines. 201
G.2 Investigation into the role of different confusion matrix estimates on ImageNet (left) and ObjectNet (right). Dashed lines in the scatter represent results for when the full confusion matrices (i.e., including the off-diagonals) are estimated from smoothed counts on a split of the validation set, and solid lines represent results under the less accurate confusion matrix estimates where only diagonal elements (i.e., the class-specific accuracies) are estimated, as in the experiments of the main paper. 202

G.3 Investigation into the role of different confusion matrix estimates on the evolution of key metrics on ImageNet. Only Diagonal Confusion Matrix Estimation (top) and Full Confusion Matrix Estimation (bottom). 203
G.4 Investigation into the role of different confusion matrix estimates on the evolution of key metrics on ObjectNet. Only Diagonal Confusion Matrix Estimation (top) and Full Confusion Matrix Estimation (bottom). 204
G.5 Summary of prediction-updates on CIFAR-10. Left: Dashed lines correspond to the ablation using a random selection strategy. Right: Similar to Figure G.2, dashed lines represent results with full confusion matrices estimated from smoothed counts and solid lines represent using a less accurate confusion matrix estimate where only diagonals are estimated from data. We observe very similar behaviour to that described for ImageNet and ObjectNet in Section 7.6.2 and Section 7.6.3, respectively. The most interesting difference compared to our experiments on ImageNet and ObjectNet, is that, on CIFAR-10, our methods achieve substantially larger accuracy gains compared with the baselines, even for small budgets. 205
G.6 Summary of prediction-updates on CIFAR-10. We observe very similar behaviour to that described for ImageNet and ObjectNet in Section 7.6.2 and Section 7.6.3, respectively. The most interesting difference compared to our experiments on ImageNet and ObjectNet, is that, on CIFAR-10, our methods achieve substantially larger accuracy gains compared with the baselines, even for small budgets. 206
G.7 BTC and BEC evolution on CIFAR10 with utilizing diagonal confusion matrix elements only. 207
G.8 Further investigation into the peaking phenomenon observed for prediction-updates on ImageNet in Figure 7.3 (right). Solid lines are as in Figure 7.3 (right), i.e., using a 50:50 split of the validation set, using one half to estimate the diagonal elements of the confusion matrices and the other half as our target evaluation set; dashed lines show the results of our ablation experiment where we use the full validation set to both estimate the exact statistics of the confusion matrix and to evaluate the different methods. Since the evaluation set for the latter is twice as large, we scaled the reported negative flips by 0.5 for ease of comparison. 207
G.9 Ablation results for comparing random (dashed lines in scatter plots) and entropy-ranked (solid lines in scatter plots) selection strategies on CIFAR-10. Different rows of the bar plots correspond to, from top to bottom, B =10, 15, 30, 50%. Note that at 100% budgets, the two selection strategies are identical as all samples are re-evaluated at each step. We find that random selection leads to substantially smaller accuracy gains, but also fewer negative flips. This is intuitive as random selection more often chooses "easy" samples for re-evaluation. The effect is particularly pronounced for small budgets. 209
G.10 Ablation results for comparing random (dashed lines in scatter plots) and entropy-ranked (solid lines in scatter plots) selection strategies on ImageNet. Different rows of the bar plots correspond to, from top to bottom, B =10, 15, 30, 50%. Note that at 100% budgets, the two selection strategies are identical as all samples are re-evaluated at each step. We find that random selection leads to substantially smaller accuracy gains, but also fewer negative flips. This is intuitive as random selection more often chooses "easy" samples for re-evaluation. The effect is particularly pronounced for small budgets. 210

G.11 Ablation results for comparing random (dashed lines in scatter plots) and entropy-ranked (solid lines in scatter plots) selection strategies on ObjectNet. Different rows of the bar plots correspond to, from top to bottom, B =10, 15, 30, 50%. Note that at 100% budgets, the two selection strategies are identical as all samples are re-evaluated at each step. We find that random selection leads to substantially smaller accuracy gains, but also fewer negative flips. This is intuitive as random selection more often chooses "easy" samples for re-evaluation. The effect is particularly pronounced for small budgets. 211
G.12 Count distribution evolution of correctly and incorrectly stored predictions on ImageNet using the MB update strategy under a budget of B = 10. Samples with currently stored predictions being false are depicted with blue, and correctly stored predictions with orange. We observe that under this MB strategy, samples with highest entropy tend to store false predictions, thus benefitting from re-evaluation particularly. 212
G.13 Comparative temporal evolution plots for experiments incorporating uncalibrated softmax labels. Cifar10 (top), ImageNet (middle) and ObjectNet(bottom). Solid lines represent using hard labels with only diagonal estimated for confusion matrix and dashed lines reflect results using soft labels and estimating full confusion matrix with Laplace smoothing. 221

List of Tables

2.1 Dataset variants with introduced correlations under investigation within this empirical study. . . . 21

2.2 The means of the pairwise entanglement scores for the correlated pair (red) and the median of the uncorrelated pairs. Stronger correlation leads to statistically more entangled latents compared to the baseline score without correlation (blue), thus uncovering still existent correlations in the latent representation. . . . 23

2.3 **Fast adaption with few labels:** Pairwise entanglement scores (averaged over 180 models each) for correlated FoV pair in Shapes3D (A). The correlated pair is highlighted (red). Zero labels reflects the unsupervised baseline. Fast adaption using linear regression can reduce these correlations with as little as 100 labels (blue). . . . 27

3.1 Comparison of Causal World with RLBench [120], MetaWorld [115], IKEA [130], BabyAI [129], CoinRun [118], AtariArcade [119] and MuJoBan [131] . . . 35

7.1 Results for ImageNet → ImageNet: all metrics refer to final performance for the improving model sequence from Figure 7.3. The character **E** or **R** in front of the strategy indicates that selection for re-evaluation is based on the entropy criterion or sampled randomly. 114

7.2 Results for ImageNet → ObjectNet: all metrics refer to final performance for the improving model sequence from Figure 7.4 respectively. The character **E** or **R** in front of the strategy indicates that selection for re-evaluation is based on the entropy criterion or sampled randomly. . . . 115

B.1 Pairwise entanglement scores help to uncover still-existent correlations in the latent representation. Mean of the pairwise entanglement scores for the correlated pair (red) and the median of the uncorrelated pairs. We see that stronger correlation leads to statistically more entanglement latents across all datasets studied compared to their baseline pairwise entanglement where the data exhibits no correlations (blue). Each score is the mean across 180 models for each dataset and correlation strength. Scores are based on GBT feature importance; scores in brackets are based on the Mutual Information. . . 140

B.2 Pairwise entanglement scores from Table B.1 separated along each disentanglement regularizer. Mean of the pairwise entanglement scores for the correlated pair (corr. pair) and the median of the uncorrelated pairs (uncorr. pairs). We see that stronger correlation leads to statistically more entanglement latents across all datasets and regularizers studied compared to their baseline pairwise entanglement where the data exhibits no correlations. Each pairwise score is the mean across 30 models for each dataset and correlation strength. Scores are based on GBT feature importance. . . . 143

B.3 Mean of the pairwise entanglement scores for the correlated pair (red) and the median of the uncorrelated pairs (based on GBT feature importance) for all pairs of variables in Shapes3D (D) (top), Shapes3D (E) (middle) and Shapes3D (A) (bottom) all with correlation strength $\sigma = 0.2$. Each pairwise score is the mean across 180 models for each dataset and correlation strength. First column is the unsupervised baseline without any fast adaptation and the second column shows that fast adaption using a one-hidden layer MLP reduces these correlations with as little as 1000 labels. . . . 149

C.1 Description of a subset of the high level variables, exposed in CausalWorld, and their corresponding spaces, h refers to the height of the block. 158
C.2 Example of task generators' specific high level variables, exposed in CausalWorld, and their corresponding spaces. 158
C.3 Description of the dense rewards applied in our experiments. The following notation was applied: $v^t \in \mathbf{R}^3$ joint velocities, $e_i^t \in \mathbf{R}^3$ i-th end-effector positions, $o_i^t \in \mathbf{R}^3$ i-th block position, $g_i^t \in \mathbf{R}^3$ i-th goal block position, $d^t(o, e) = \sum_i ||e_i^t - o^t||$ the distance between end-effectors and the block, $\Delta_{o,e}^t = d^t(o, e) - d^{t-1}(o, e)$ the distance difference w.r.t. the previous timestep. The target height parameter t for pick and place is 0.15 if block and goal are of different height. Otherwise, t is half the goal height. 159
C.4 PPO, SAC and TD3 Hyper-Parameters used in the discussed experiments. 160

D.1 Encoder (left) and decoder (right) architectures. The latent space dimensionality is denoted by d, and $K = 3$ indicates the number of image channels. Last line in the encoder architecture: the fully connected layer parameterizing the log variance of the approximate posterior distributions of the latent variables has custom initialization. The weights are initialized with 1/10 standard deviation than the default value, and the biases are initialized to -1 instead of 0. Empirically, this together with (learnable) LayerNorm was beneficial for training stability at the beginning of training. 164
D.2 Architecture of one residual block. The scalar gate is implemented by multiplying the tensor by a learnable scalar parameter before adding it to the block input. Initializing the residual block to the identity by setting this parameter to zero has been originally proposed by Bachlechner et al. [176]. The tensor shape is constant throughout the residual block. 165

F.1 Relative mean absolute distance between values trained i.i.d. and class-incremental for the five different backbones from the base experiment. 195

G.1 Time measurements of a single forward pass versus the mean average time to compute the posterior update (0.406 milliseconds) on the same computational ressources for comparability. For reference we extrapolated how long it would take to infer the predictions from 1B images. 213
G.2 Full results of all experiments on **ImageNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote. 214

G.3 Full results of all experiments on **ObjectNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote. 215
G.4 Full results of all experiments on **CIFAR10** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote. 216
G.5 Full results of all experiments on **CIFAR-10** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing **and incorporating soft labels**. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote. 218
G.6 Full results of all experiments on **ImageNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing **and incorporating soft labels**. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote. 219

G.7 Full results of all experiments on **ObjectNet** under the standard setting of an **improving sequence of models** and **estimating confusion matrices on a separate split** of the data (i.e., not on the target data set) as explained in Section 7.6.1. The first number in each cell refers to estimating only the diagonal elements (i.e., the class-specific accuracies) of confusion matrices, and the **second number (in brackets) refers to estimating the full confusion matrices** with smoothing **and incorporating soft labels**. The character **E** in front of a budget indicates that selection for re-evaluation is based on our posterior label **entropy criterion** (e.g., E30 refers to selecting the top 30% samples with highest entropy), and the character **R** indicates selecting a **randomly** sampled subset without replacement. Note that entropy-based selection requires a posterior and is thus not applicable for the baselines Replace and Majority Vote. 220

Bibliography

[1] **Frederik Träuble**, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. 'On Disentangled Representations Learned from Correlated Data'. In: *International Conference on Machine Learning, selected for Long Oral talk*. PMLR. 2021, pp. 10401–10412 (cited on pages ix, 9, 13, 49, 51, 54, 68, 71).

[2] **Ossama Ahmed**, **Frederik Träuble**, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf, and Stefan Bauer. 'Causalworld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning'. In: *International Conference for Learning Representations* (2021) (cited on pages ix, 9, 30, 31, 65, 66, 68, 70, 71, 79, 173).

[3] **Andrea Dittadi**, **Frederik Träuble**, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. 'On the Transfer of Disentangled Representations in Realistic Settings'. In: *International Conference on Learning Representations (ICLR)* (2021) (cited on pages ix, 10, 15, 40, 47, 65, 67–71, 73, 75–78, 173, 174).

[4] **Frederik Träuble**, **Andrea Dittadi**, Manuel Wüthrich, Felix Widmaier, Peter Vincent Gehler, Ole Winther, Francesco Locatello, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. 'The Role of Pretrained Representations for the OOD Generalization of RL Agents'. In: *International Conference on Learning Representations (ICLR)*. 2022 (cited on pages ix, 11, 30, 63).

[5] **Frederik Träuble**, Anirudh Goyal, Nasim Rahaman, Michael Curtis Mozer, Kenji Kawaguchi, Yoshua Bengio, and Bernhard Schölkopf. 'Discrete Key-Value Bottleneck'. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 34431–34455 (cited on pages ix, 11, 81).

[6] **Frederik Träuble**, Julius Von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter Gehler. 'Backward-Compatible Prediction Updates: A Probabilistic Approach'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 116–128 (cited on pages ix, 12, 99).

[7] **Yukun Chen**, Andrea Dittadi, Frederik Träuble, Stefan Bauer, and Bernhard Schölkopf. 'Boxhead: A Dataset for Learning Hierarchical Representations'. In: *SVRHM 2021 Workshop at NeurIPS*. 2021 (cited on pages x, 68).

[8] **Lukas Schott**, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. 'Visual Representation Learning Does Not Generalize Strongly Within the Same Domain'. In: *International Conference on Learning Representations (ICLR)* (2022) (cited on pages x, 122, 123, 135).

[9] **Davide Mambelli**, **Frederik Träuble**, Stefan Bauer, Bernhard Schölkopf, and Francesco Locatello. 'Compositional Multi-Object Reinforcement Learning with Linear Relation Networks'. In: *ICLR Workshop on the Elements of Reasoning: Objects, Structure and Causality* (2022) (cited on pages x, 122, 133).

[10] **Nasim Rahaman**, **Martin Weiss**, Frederik Träuble, Francesco Locatello, Alexandre Lacoste, Yoshua Bengio, Chris Pal, Li Erran Li, and Bernhard Schölkopf. 'A General Purpose Neural Architecture for Geospatial Systems'. In: *Workshop on AI for Humanitarian Assistance and Disaster Response at NeurIPS* (2022) (cited on page x).

[11] **Cian Eastwood**, **Andrei Liviu Nicolicioiu**, **Julius Von Kügelgen**, Armin Kekić, Frederik Träuble, Andrea Dittadi, and Bernhard Schölkopf. 'DCI-ES: An Extended Disentanglement Framework with Connections to Identifiability'. In: *International Conference on Learning Representations (ICLR)* (2023) (cited on page x).

[12] **Vedant Shah**, Frederik Träuble, Ashish Malik, Hugo Larochelle, Michael Mozer, Sanjeev Arora, Yoshua Bengio, and Anirudh Goyal. 'Unlearning via Sparse Representations'. In: *arXiv preprint arXiv:2311.15268* (2023) (cited on pages x, 123).

[13] **Frederik Träuble**, **Lachlan Stuart**, **Andreas Georgiou**, Pascal Notin, Arash Mehrjou, Ron Schwessinger, Mathieu Chevalley, Kim Branson, Bernhard Schölkopf, Cornelia van Duijn, Debora Marks, and Patrick Schwab. 'Multi-Megabase Scale Genome Interpretation with Genetic Language Models'. In: *arXiv preprint: arXiv:2501.07737* (2025) (cited on pages x, 127, 131).

[14] Kurt W Fischer. 'A theory of cognitive development: The control and construction of hierarchies of skills.' In: *Psychological review* 87.6 (1980), p. 477 (cited on page 1).

[15] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co, 1999 (cited on page 1).

[16] A. M. Turing. 'Computing Machinery and Intelligence'. In: *Mind* LIX.236 (Oct. 1950), pp. 433–460 (cited on page 1).

[17] Frank Rosenblatt. 'The perceptron: a probabilistic model for information storage and organization in the brain.' In: *Psychological review* 65.6 (1958), p. 386 (cited on page 1).

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 'Deep learning'. In: *nature* 521.7553 (2015) (cited on page 1).

[19] George Cybenko. 'Approximation by superpositions of a sigmoidal function'. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314 (cited on page 1).

[20] Ken-Ichi Funahashi. 'On the approximate realization of continuous mappings by neural networks'. In: *Neural networks* 2.3 (1989), pp. 183–192 (cited on page 1).

[21] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 'Multilayer feedforward networks are universal approximators'. In: *Neural networks* 2.5 (1989), pp. 359–366 (cited on page 1).

[22] Kurt Hornik. 'Approximation capabilities of multilayer feedforward networks'. In: *Neural networks* 4.2 (1991), pp. 251–257 (cited on page 1).

[23] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 'Learning representations by back-propagating errors'. In: *nature* 323.6088 (1986), pp. 533–536 (cited on page 1).

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cited on page 1).

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016 (cited on pages 1, 85, 112, 186).

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 'Attention is all you need'. In: *Advances in neural information processing systems* 30 (2017) (cited on pages 1, 125).

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 'Imagenet classification with deep convolutional neural networks'. In: *Advances in neural information processing systems* 25 (2012) (cited on pages 1, 33, 112).

[28] Karen Simonyan and Andrew Zisserman. 'Very deep convolutional networks for large-scale image recognition'. In: *International Conference on Learning Representations* (2015) (cited on pages 1, 112).

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. In: *International Conference on Learning Representations*. 2020 (cited on pages 1, 85, 101).

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 'Sequence to sequence learning with neural networks'. In: *Advances in neural information processing systems* 27 (2014) (cited on page 1).

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 'Language models are unsupervised multitask learners'. In: *OpenAI blog* 1.8 (2019), p. 9 (cited on pages 1, 125).

[32] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 'Language models are few-shot learners'. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901 (cited on pages 1, 83, 125, 126).

[33] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 'Gpt-4 technical report'. In: *arXiv preprint arXiv:2303.08774* (2023) (cited on pages 1, 125).

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 'Playing atari with deep reinforcement learning'. In: *Deep Learning Workshop at NIPS* (2020) (cited on pages 1, 33).

[35] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 'Mastering the game of go without human knowledge'. In: *nature* 550.7676 (2017), pp. 354–359 (cited on pages 1, 33).

[36] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 'Grandmaster level in StarCraft II using multi-agent reinforcement learning'. In: *Nature* 575.7782 (2019), pp. 350–354 (cited on pages 1, 65).

[37] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 'Highly accurate protein structure prediction with AlphaFold'. In: *Nature* 596.7873 (2021), pp. 583–589 (cited on page 1).

[38] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. 'Effective gene expression prediction from sequence by integrating long-range interactions'. In: *Nature methods* 18.10 (2021), pp. 1196–1203 (cited on pages 1, 132).

[39] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 'End-to-end training of deep visuomotor policies'. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373 (cited on page 1).

[40] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 'Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor'. In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870 (cited on pages 1, 42, 68).

[41] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 'Learning dexterous in-hand manipulation'. In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20 (cited on pages 1, 51, 52).

[42] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. 'The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization'. In: *ICCV* (2021) (cited on pages 1, 2, 124, 125).

[43] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 'Shortcut learning in deep neural networks'. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673 (cited on pages 1, 2, 18).

[44] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999 (cited on page 2).

[45] Judea Pearl. *Causality*. Cambridge university press, 2009 (cited on pages 3, 41).

[46] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 'Toward causal representation learning'. In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634 (cited on pages 3, 5, 7, 15, 39–41, 51, 65, 162).

[47] Yoshua Bengio. 'Deep learning of representations for unsupervised and transfer learning'. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. 2012, pp. 17–36 (cited on page 5).

[48] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. Vol. 1. MIT press Cambridge, 2016 (cited on page 5).

[49] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 'Representation learning: A review and new perspectives'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828 (cited on pages 5, 15, 65).

[50] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 'Building machines that learn and think like people'. In: *Behavioral and brain sciences* 40 (2017), e253 (cited on pages 5, 65).

[51] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. 'On causal and anticausal learning'. In: *International Conference on Machine Learning (ICML)*. 2012 (cited on pages 6, 7).

[52] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017 (cited on page 6).

[53] Yoshua Bengio. 'Deep learning of representations: Looking forward'. In: *International conference on statistical language and speech processing*. Springer. 2013, pp. 1–37 (cited on page 6).

[54] Diederik P Kingma and Max Welling. 'Auto-encoding variational Bayes'. In: *International Conference on Learning Representations (ICLR)*. 2014 (cited on pages 6, 17, 50, 67).

[55] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 'Stochastic backpropagation and approximate inference in deep generative models'. In: *International conference on machine learning*. PMLR. 2014, pp. 1278–1286 (cited on pages 6, 17, 50, 67).

[56] Michal Rolinek, Dominik Zietlow, and Georg Martius. 'Variational Autoencoders Recover PCA Directions (by Accident)'. In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*. 2019 (cited on pages 7, 29).

[57] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 'beta-VAE: Learning basic visual concepts with a constrained variational framework'. In: *International Conference on Learning Representations (ICLR)*. 2017 (cited on pages 7, 17, 18, 20, 50, 51, 67, 71, 138, 139).

[58] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 'Understanding disentangling in beta-VAE'. In: *arXiv preprint arXiv:1804.03599* (2018) (cited on pages 7, 17, 50, 67, 138).

[59] Hyunjik Kim and Andriy Mnih. 'Disentangling by factorising'. In: *International Conference on Machine Learning (ICML)*. 2018 (cited on pages 7, 17, 18, 50, 51, 71, 138, 139).

[60] Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. 'Isolating Sources of Disentanglement in Variational Autoencoders'. In: *Advances in Neural Information Processing Systems*. 2018 (cited on pages 7, 17, 29, 50, 51, 71, 76, 138).

[61] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 'Variational inference of disentangled latent concepts from unlabeled observations'. In: *International Conference on Learning Representations (ICLR)*. 2018 (cited on pages 7, 17, 50, 51, 76, 138, 139).

[62] Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. 'Disentangling Disentanglement in Variational Auto-Encoders'. In: *International Conference on Machine Learning (ICML)*. 2019 (cited on pages 7, 17).

[63] Dominik Zietlow, Michal Rolinek, and Georg Martius. 'Demystifying inductive biases for (beta-) VAE based architectures'. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12945–12954 (cited on page 7).

[64] David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. 'Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding'. In: *International Conference on Learning Representations* (2021) (cited on pages 7, 17, 18, 50, 51, 122).

[65] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 'Challenging common assumptions in the unsupervised learning of disentangled representations'. In: *International Conference on Machine Learning (ICML)*. 2019 (cited on pages 7, 17, 19, 22, 50, 53–55, 58, 138, 163, 164).

[66] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem. 'Disentangling Factors of Variations Using Few Labels'. In: *International Conference on Learning Representations (ICLR)*. Apr. 2020 (cited on pages 7, 17).

[67] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen. 'Weakly-Supervised Disentanglement Without Compromises'. In: *International Conference on Machine Learning (ICML)*. July 2020, pp. 7753–7764 (cited on pages 7, 15, 17–20, 50, 51, 53, 55, 68, 71, 149, 163).

[68] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. 'Are disentangled representations helpful for abstract visual reasoning?' In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on pages 7, 65).

[69] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 'Fairness through causal awareness: Learning causal latent-variable models for biased data'. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 349–358 (cited on pages 7, 20).

[70] Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. 'On the fairness of disentangled representations'. In: *Advances in Neural Information Processing Systems*. 2019, pp. 14611–14624 (cited on pages 7, 16, 20, 24, 49, 60, 75, 139).

[71] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. 'DARLA: Improving Zero-Shot Transfer in Reinforcement Learning'. In: *International Conference on Machine Learning (ICML)*. 2017 (cited on pages 15, 49, 51, 79).

[72] Tameem Adel, Zoubin Ghahramani, and Adrian Weller. 'Discovering interpretable representations for both deep generative and discriminative models'. In: *International Conference on Machine Learning (ICML)*. 2018, pp. 50–59 (cited on pages 15, 49).

[73] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bošnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. 'SCAN: Learning Hierarchical Compositional Visual Concepts'. In: *International Conference on Learning Representations (ICLR)*. 2018 (cited on pages 15, 49).

[74] Đorđe Miladinović, Muhammad Waleed Gondal, Bernhard Schölkopf, Joachim M Buhmann, and Stefan Bauer. 'Disentangled state space representations'. In: *ICLR Workshop on Deep Generative Models for Highly Structured Data*. 2020 (cited on page 15).

[75] Raphael Suter, Djordje Miladinović, Stefan Bauer, and Bernhard Schölkopf. 'Interventional Robustness of Deep Latent Variable Models'. In: *International Conference on Machine Learning (ICML)*. 2019 (cited on page 15).

[76] Arun Kumar Agnihotri, Brijesh Purwar, Kreshna Googoolye, Smriti Agnihotri, and Nilima Jeebun. 'Estimation of stature by foot length'. In: *Journal of forensic and legal medicine* 14.5 (2007), pp. 279–283 (cited on page 15).

[77] Theodoros B Grivas, Constantinos Mihas, Angeliki Arapaki, and Elias Vasiliadis. 'Correlation of foot length with height and weight in school age children'. In: *Journal of forensic and legal medicine* 15.2 (2008), pp. 89–95 (cited on pages 15, 20).

[78] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. 'Recurrent independent mechanisms'. In: *International Conference on Learning Representations* (2021) (cited on pages 15, 17, 18, 41).

[79] Felix Leeb, Giulia Lanzillotta, Yashas Annadani, Michel Besserve, Stefan Bauer, and Bernhard Schölkopf. 'Structure by architecture: Structured representations without regularization'. In: *The Eleventh International Conference on Learning Representations*. 2022 (cited on page 15).

[80] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. Cambridge, MA, USA: MIT Press, 2017 (cited on page 15).

[81] Agisilaos Chartsias, Thomas Joyce, Giorgos Papanastasiou, Scott Semple, Michelle Williams, David Newby, Rohan Dharmakumar, and Sotirios A Tsaftaris. 'Factorised spatial representation learning: Application in semi-supervised myocardial segmentation'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 490–498 (cited on pages 16, 60).

[82] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 'Learning adversarially fair and transferable representations'. In: *International Conference on Machine Learning (ICML)*. 2018 (cited on page 16).

[83] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. 'Flexibly fair representation learning by disentanglement'. In: *International Conference on Machine Learning (ICML)*. 2019 (cited on pages 16, 20, 29).

[84] Cian Eastwood and Christopher KI Williams. 'A framework for the quantitative evaluation of disentangled representations'. In: *International Conference on Learning Representations (ICLR)*. 2018 (cited on pages 17, 22, 50, 51, 76, 139).

[85] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. 'Multi-level variational autoencoder: Learning disentangled representations from grouped observations'. In: *AAAI Conference on Artificial Intelligence*. 2018 (cited on page 17).

[86] Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. 'Weakly Supervised Disentanglement with Guarantees'. In: *International Conference on Learning Representations (ICLR)*. 2020 (cited on pages 17, 18, 50).

[87] Peter Földiák. 'Learning invariance from transformation sequences'. In: *Neural Computation* 3.2 (1991), pp. 194–200 (cited on pages 17, 18).

[88] Mark Schmidt, Alexandru Niculescu-Mizil, Kevin Murphy, et al. 'Learning graphical model structure using L1-regularization paths'. In: *AAAI*. Vol. 7. 2007, pp. 1278–1283 (cited on pages 17, 18).

[89] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 'A meta-transfer objective for learning to disentangle causal mechanisms'. In: *International Conference for Learning Representations* (2020) (cited on pages 17, 18).

[90] Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. 'Learning neural causal models from unknown interventions'. In: *arXiv preprint arXiv:1910.01075* (2019) (cited on pages 17, 18).

[91] Scott Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. 'Deep visual analogy-making'. In: *Advances in Neural Information Processing Systems*. 2015 (cited on pages 17, 18, 51).

[92] Yann LeCun, Fu Jie Huang, and Leon Bottou. 'Learning methods for generic object recognition with invariance to pose and lighting'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2004 (cited on pages 17, 18, 51).

[93] Muhammad Waleed Gondal, Manuel Wüthrich, Djordje Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. 'On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset'. In: *Advances in Neural Information Processing Systems*. 2019 (cited on pages 17, 18, 51, 79).

[94] Zejian Li, Yongchuan Tang, Wei Li, and Yongxing He. 'Learning Disentangled Representation with Pairwise Independence'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4245–4252 (cited on pages 17, 18, 29).

[95] N. Kilbertus, M. Rojas Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf. 'Avoiding Discrimination through Causal Reasoning'. In: *Advances in Neural Information Processing Systems*. 2017, pp. 656–666 (cited on page 20).

[96] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 'Fairness through awareness'. In: *Proceedings of the 3rd innovations in theoretical computer science conference*. ACM. 2012, pp. 214–226 (cited on page 24).

[97] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. 'Bias and generalization in deep generative models: An empirical study'. In: *Advances in Neural Information Processing Systems*. 2018 (cited on page 25).

[98] Pierre Comon. 'Independent component analysis, a new concept?' In: *Signal Processing* 36.3 (1994), pp. 287–314 (cited on pages 29, 50).

[99] Christian Jutten and Juha Karhunen. 'Advances in nonlinear blind source separation'. In: *International Symposium on Independent Component Analysis and Blind Signal Separation*. 2003, pp. 245–256 (cited on pages 29, 50).

[100] Aapo Hyvärinen and Petteri Pajunen. 'Nonlinear independent component analysis: Existence and uniqueness results'. In: *Neural Networks* (1999) (cited on pages 29, 50).

[101] Aapo Hyvarinen, Hiroaki Sasaki, and Richard E Turner. 'Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning'. In: *International Conference on Artificial Intelligence and Statistics*. 2019 (cited on pages 29, 50).

[102] Aapo Hyvarinen and Hiroshi Morioka. 'Unsupervised feature extraction by time-contrastive learning and nonlinear ICA'. In: *Advances in Neural Information Processing Systems*. 2016 (cited on pages 29, 50).

[103] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. 'Variational autoencoders and nonlinear ica: A unifying framework'. In: *International Conference on Artificial Intelligence and Statistics*. 2020, pp. 2207–2217 (cited on pages 29, 50, 51).

[104] Peter Sorrenson, Carsten Rother, and Ullrich Köthe. 'Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN)'. In: *International Conference on Learning Representations (ICLR)*. 2020 (cited on pages 29, 50, 51).

[105] Sharon Zhou, Eric Zelikman, Fred Lu, Andrew Y Ng, and Stefano Ermon. 'Evaluating the Disentanglement of Deep Generative Models through Manifold Topology'. In: *International Conference on Learning Representations (ICLR)*. 2021 (cited on page 29).

[106] Jan Stühmer, Richard Turner, and Sebastian Nowozin. 'Independent subspace analysis for unsupervised learning of disentangled representations'. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1200–1210 (cited on page 29).

[107] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 'A 3D face model for pose and illumination invariant face recognition'. In: *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. Ieee. 2009, pp. 296–301 (cited on page 29).

[108] Rob Brekelmans, Daniel Moyer, Aram Galstyan, and Greg Ver Steeg. 'Exact rate-distortion in autoencoders via echo noise'. In: *Advances in Neural Information Processing Systems*. 2019, pp. 3889–3900 (cited on page 29).

[109] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 'CausalVAE: Structured Causal Disentanglement in Variational Autoencoder'. In: *arXiv preprint arXiv:2004.08697* (2020) (cited on page 29).

[110] Đorđe Miladinović, Aleksandar Stanić, Stefan Bauer, Jürgen Schmidhuber, and Joachim M Buhmann. 'Spatial Dependency Networks: Neural Layers for Improved Generative Image Modeling'. In: *International Conference on Learning Representations (ICLR)* (2021) (cited on page 30).

[111] Alex Krizhevsky, Geoffrey Hinton, et al. 'Learning multiple layers of features from tiny images'. In: . (2009) (cited on pages 33, 96, 111).

[112] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. 'Solving rubik's cube with a robot hand'. In: *arXiv preprint arXiv:1910.07113* (2019) (cited on pages 33, 36, 79).

[113] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 'Emergent tool use from multi-agent autocurricula'. In: *International Conference on Learning Representations* (2019) (cited on page 33).

[114] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 'Dota 2 with large scale deep reinforcement learning'. In: *arXiv preprint arXiv:1912.06680* (2019) (cited on page 33).

[115] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 'Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning'. In: *Conference on robot learning*. PMLR. 2020, pp. 1094–1100 (cited on pages 33, 35, 40).

[116] Shimon Whiteson, Brian Tanner, Matthew E Taylor, and Peter Stone. 'Protecting against evaluation overfitting in empirical reinforcement learning'. In: *2011 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*. IEEE. 2011, pp. 120–127 (cited on page 33).

[117] Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. 'Infobot: Transfer and exploration via the information bottleneck'. In: *arXiv preprint arXiv:1901.10902* (2019) (cited on page 33).

[118] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. 'Quantifying generalization in reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1282–1289 (cited on pages 33–35, 65, 79).

[119] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 'The arcade learning environment: An evaluation platform for general agents'. In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 253–279 (cited on pages 33, 35).

[120] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. 'Rlbench: The robot learning benchmark & learning environment'. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3019–3026 (cited on pages 33, 35, 40).

[121] Beth M Casey, Nicole Andrews, Holly Schindler, Joanne E Kersh, Alexandra Samper, and Juanita Copley. 'The development of spatial skills through interventions involving block building activities'. In: *Cognition and Instruction* 26.3 (2008), pp. 269–309 (cited on page 33).

[122] Yvonne M Caldera, Anne McDonald Culp, Marion O'Brien, Rosemarie T Truglio, Mildred Alvarez, and Aletha C Huston. 'Children's play preferences, construction play with blocks, and visual-spatial skills: Are they related?' In: *International Journal of Behavioral Development* 23.4 (1999), pp. 855–872 (cited on page 33).

[123] Constance Kamii, Yoko Miyakawa, and Yasuhiko Kato. 'The development of logico-mathematical knowledge in a block-building activity at ages 1–4'. In: *Journal of Research in Childhood Education* 19.1 (2004), pp. 44–57 (cited on page 33).

[124] Brian N Verdine, Roberta Michnick Golinkoff, Kathy Hirsh-Pasek, and Nora Newcombe. *Links between spatial and mathematical skills across the preschool years*. Wiley, 2017 (cited on pages 33, 34).

[125] Swiya Nath and Dénes Szücs. 'Construction play and cognitive skills associated with the development of mathematical abilities in 7-year-old children'. In: *Learning and Instruction* 32 (2014), pp. 73–80 (cited on pages 33, 34).

[126] Gwen Dewar. 'The benefits of toy blocks: The science of construction play'. In: *Parentig Science* (2018) (cited on pages 33, 34).

[127] Miles Richardson, Thomas E Hunt, and Cassandra Richardson. 'Children's construction task performance and spatial ability: Controlling task complexity and predicting mathematics performance'. In: *Perceptual and motor skills* 119.3 (2014), pp. 741–757 (cited on pages 33, 34).

[128] Manuel Wüthrich, Felix Widmaier, Felix Grimminger, Shruti Joshi, Vaibhav Agrawal, Bilal Hammoud, Majid Khadiv, Miroslav Bogdanovic, Vincent Berenz, Julian Viereck, et al. 'TriFinger: An Open-Source Robot for Learning Dexterity'. In: *Conference on Robot Learning*. PMLR. 2021, pp. 1871–1882 (cited on pages 34, 36, 37, 49, 52, 65, 68, 69, 157).

[129] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 'BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning'. In: *International Conference on Learning Representations*. 2018 (cited on pages 34, 35, 40).

[130] Youngwoon Lee, Edward S Hu, and Joseph J Lim. 'IKEA furniture assembly environment for long-horizon complex manipulation tasks'. In: *2021 ieee international conference on robotics and automation (icra)*. IEEE. 2021, pp. 6343–6349 (cited on page 35).

[131] Mehdi Mirza, Andrew Jaegle, Jonathan J. Hunt, Arthur Guez, Saran Tunyasuvunakool, Alistair Muldal, Théophane Weber, Peter Karkus, Sébastien Racanière, Lars Buesing, Timothy Lillicrap, and Nicolas Heess. *Physically Embedded Planning Problems: New Challenges for Reinforcement Learning*. 2020 (cited on page 35).

[132] Michael Janner, Sergey Levine, William T Freeman, Joshua B Tenenbaum, Chelsea Finn, and Jiajun Wu. 'Reasoning About Physical Interactions with Object-Oriented Prediction and Planning'. In: *International Conference on Learning Representations*. 2019 (cited on page 36).

[133] Victor Bapst, Alvaro Sanchez-Gonzalez, Carl Doersch, Kimberly Stachenfeld, Pushmeet Kohli, Peter Battaglia, and Jessica Hamrick. 'Structured agents for physical construction'. In: *International conference on machine learning*. PMLR. 2019, pp. 464–474 (cited on page 36).

[134] William P McCarthy, David Kirsh, and Judith E Fan. 'Learning to build physical structures better over time.' In: *CogSci*. 2020 (cited on page 36).

[135] Scott Elliott Fahlman. 'A planning system for robot construction tasks'. In: *Artificial intelligence* 5.1 (1974), pp. 1–49 (cited on page 36).

[136] Patrick H Winston. 'Learning Structual Descriptions from Examples'. In: *The Theory of Computer Vision* (1972), pp. 157–209 (cited on page 36).

[137] Terry Winograd. 'Understanding natural language'. In: *Cognitive psychology* 3.1 (1972), pp. 1–191 (cited on page 36).

[138] Erwin Coumans et al. 'Bullet real-time physics simulation'. In: *URL http://bulletphysics. org* (2013) (cited on page 36).

[139] Richard S Sutton and Andrew G Barto. 'Reinforcement learning'. In: *Journal of Cognitive Neuroscience* 11.1 (1999), pp. 126–134 (cited on page 39).

[140] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. 'Woulda, coulda, shoulda: Counterfactually-guided policy search'. In: *arXiv preprint arXiv:1811.06272* (2018) (cited on pages 39, 162).

[141] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 'Deepmind control suite'. In: *arXiv preprint arXiv:1801.00690* (2018) (cited on page 40).

[142] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 'Openai gym'. In: *arXiv preprint arXiv:1606.01540* (2016) (cited on page 40).

[143] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. 'Assessing generalization in deep reinforcement learning'. In: *arXiv preprint arXiv:1810.12282* (2018) (cited on page 40).

[144] Aravind Rajeswaran, Kendall Lowrey, Emanuel V Todorov, and Sham M Kakade. 'Towards generalization and simplicity in continuous control'. In: *Advances in Neural Information Processing Systems*. 2017, pp. 6550–6561 (cited on page 40).

[145] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 'Robust adversarial reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2817–2826 (cited on page 40).

[146] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. 'Preparing for the unknown: Learning a universal policy with online system identification'. In: *RSS* (2017) (cited on page 40).

[147] Peter Henderson, Wei-Di Chang, Florian Shkurti, Johanna Hansen, David Meger, and Gregory Dudek. 'Benchmark environments for multitask learning in continuous domains'. In: *Lifelong Learning: A Reinforcement Learning Approach Workshop at ICML* (2017) (cited on page 40).

[148] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. 'An empirical investigation of the challenges of real-world reinforcement learning'. In: *arXiv preprint arXiv:2003.11881* (2020) (cited on page 40).

[149] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 'Deep reinforcement learning that matters'. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018 (cited on page 41).

[150] Stephanie CY Chan, Samuel Fishman, Anoop Korattikara, John Canny, and Sergio Guadarrama. 'Measuring the Reliability of Reinforcement Learning Algorithms'. In: *International Conference on Learning Representations*. 2019 (cited on page 41).

[151] Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepezvari, Satinder Singh, et al. 'Behaviour suite for reinforcement learning'. In: *International Conference on Learning Representations (ICLR)* (2020) (cited on page 41).

[152] Thomas L Griffiths and Joshua B Tenenbaum. 'Structure and strength in causal induction'. In: *Cognitive psychology* 51.4 (2005), pp. 334–384 (cited on page 41).

[153] Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. 'CoPhy: Counterfactual Learning of Physical Dynamics'. In: *International Conference on Learning Representations*. 2019 (cited on page 41).

[154] Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. 'Phyre: A new benchmark for physical reasoning'. In: *Advances in Neural Information Processing Systems*. 2019, pp. 5082–5093 (cited on page 41).

[155] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 'Proximal policy optimization algorithms'. In: *arXiv:1707.06347* (2017) (cited on pages 42, 68, 71).

[156] Scott Fujimoto, Herke Hoof, and David Meger. 'Addressing function approximation error in actor-critic methods'. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1587–1596 (cited on pages 42, 68, 71).

[157] Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. 'Are Disentangled Representations Helpful for Abstract Visual Reasoning?' In: *Advances in Neural Information Processing Systems*. 2019 (cited on page 49).

[158] Francis Bach and Michael Jordan. 'Kernel independent component analysis'. In: *Journal of Machine Learning Research* 3.7 (2002), pp. 1–48 (cited on page 50).

[159] Luigi Gresele, Paul K. Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. 'The Incomplete Rosetta Stone Problem: Identifiability Results for Multi-View Nonlinear ICA'. In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. 2019 (cited on page 50).

[160] Karl Ridgeway and Michael C Mozer. 'Learning Deep Disentangled Embeddings with the F-Statistic Loss'. In: *Advances in Neural Information Processing Systems*. 2018 (cited on pages 50, 51, 76).

[161] Sanja Fidler, Sven Dickinson, and Raquel Urtasun. '3d object detection and viewpoint estimation with a deformable 3d cuboid model'. In: *Advances in neural information processing systems*. 2012, pp. 611–619 (cited on page 51).

[162] Peter Dayan. 'Improving generalization for temporal difference learning: The successor representation'. In: *Neural Computation* 5.4 (1993), pp. 613–624 (cited on page 51).

[163] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 'Domain generalization via invariant feature representation'. In: *International Conference on Machine Learning*. 2013, pp. 10–18 (cited on page 51).

[164] Christina Heinze-Deml and Nicolai Meinshausen. 'Conditional variance penalties and domain shift robustness'. In: *arXiv preprint arXiv:1710.11469* (2017) (cited on page 51).

[165] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 'Invariant models for causal transfer learning'. In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 1309–1342 (cited on page 51).

[166] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. 'Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6056–6065 (cited on page 51).

[167] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 'Deep domain generalization via conditional invariant adversarial networks'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 624–639 (cited on page 51).

[168] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 'Invariant risk minimization'. In: *arXiv preprint arXiv:1907.02893* (2019) (cited on pages 51, 124).

[169] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 'Out-of-distribution generalization via risk extrapolation (rex)'. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5815–5826 (cited on page 51).

[170] Sven Gowal, Chongli Qin, Po-Sen Huang, Taylan Cemgil, Krishnamurthy Dvijotham, Timothy Mann, and Pushmeet Kohli. 'Achieving robustness in the wild via adversarial mixing with disentangled representations'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1211–1220 (cited on page 51).

[171] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 'Domain randomization for transferring deep neural networks from simulation to the real world'. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30 (cited on pages 51, 52, 77).

[172] Andrei A Rusu, Matej Večerík, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. 'Sim-to-real robot learning from pixels with progressive nets'. In: *Conference on Robot Learning*. 2017, pp. 262–270 (cited on pages 51, 52, 77).

[173] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 'Sim-to-real transfer of robotic control with dynamics randomization'. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 1–8 (cited on pages 51, 52).

[174] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. 'Sim-to-real via sim-to-sim: Data-efficient robotic grasping via random-ized-to-canonical adaptation networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12627–12637 (cited on pages 51, 52, 79).

[175] Mengyuan Yan, Qingyun Sun, Iuri Frosio, Stephen Tyree, and Jan Kautz. 'How to Close Sim-Real Gap? Transfer with Segmentation!' In: *arXiv preprint arXiv:2005.07695* (2020) (cited on pages 51, 52).

[176] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. 'Rezero is all you need: Fast convergence at large depth'. In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 1352–1361 (cited on pages 53, 163, 165).

[177] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 'Generating sentences from a continuous space'. In: *arXiv preprint arXiv:1511.06349* (2015) (cited on page 53).

[178] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 'Ladder variational autoencoders'. In: *Advances in neural information processing systems*. 2016, pp. 3738–3746 (cited on page 53).

[179] Jocelyn Sietsma and Robert JF Dow. 'Creating artificial neural networks that generalize'. In: *Neural networks* 4.1 (1991), pp. 67–79 (cited on pages 53, 59).

[180] Chris M Bishop. 'Training with noise is equivalent to Tikhonov regularization'. In: *Neural computation* 7.1 (1995), pp. 108–116 (cited on pages 53, 59).

[181] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. 'On robustness and transferability of convolutional neural networks'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16458–16468 (cited on pages 65, 77).

[182] Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. 'Wilds: A benchmark of in-the-wild distribution shifts'. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5637–5664 (cited on pages 65, 124, 125).

[183] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. 'ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models'. In: *Advances in Neural Information Processing Systems*. 2019, pp. 9448–9458 (cited on pages 65, 111).

[184] Aharon Azulay and Yair Weiss. 'Why do deep convolutional networks generalize so poorly to small image transformations?' In: *Journal of Machine Learning Research* 20.184 (2019), pp. 1–25 (cited on page 65).

[185] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. 'Effects of degradations on deep neural network architectures'. In: *arXiv preprint 1807.10108* (2018) (cited on page 65).

[186] Ishaan Gulrajani and David Lopez-Paz. 'In Search of Lost Domain Generalization'. In: *International Conference on Learning Representations*. 2020 (cited on page 65).

[187] Dan Hendrycks and Thomas Dietterich. 'Benchmarking Neural Network Robustness to Common Corruptions and Perturbations'. In: *Proceedings of the International Conference on Learning Representations* (2019) (cited on pages 65, 124, 125).

[188] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. 'Benchmarking robustness in object detection: Autonomous driving when winter is coming'. In: *Machine Learning for Autonomous Driving Workshop at NeurIPS* (2019) (cited on pages 65, 124, 125).

[189] Niklas Funk, Charles Schaff, Rishabh Madan, Takuma Yoneda, Julen Urain De Jesus, Joe Watson, Ethan K. Gordon, Felix Widmaier, Stefan Bauer, Siddhartha S. Srinivasa, Tapomayukh Bhattacharjee, Matthew R. Walter, and Jan Peters. 'Benchmarking Structured Policies and Policy Optimization for Real-World Dexterous Object Manipulation'. In: *IEEE Robotics and Automation Letters Special Issue: Robotic Grasping and Manipulation Challenges and Progress* (2021) (cited on page 65).

[190] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. 'A study on overfitting in deep reinforcement learning'. In: *arXiv preprint arXiv:1804.06893* (2018) (cited on page 65).

[191] Niklas Pfister, Stefan Bauer, and Jonas Peters. 'Learning stable and predictive structures in kinetic systems'. In: *Proceedings of the National Academy of Sciences* 116.51 (2019), pp. 25405–25411 (cited on page 65).

[192] Nan Rosemary Ke, Aniket Rajiv Didolkar, Sarthak Mittal, Anirudh Goyal, Guillaume Lajoie, Stefan Bauer, Danilo Jimenez Rezende, Yoshua Bengio, Christopher Pal, and Michael Curtis Mozer. 'Systematic Evaluation of Causal Discovery in Visual Model Based Reinforcement Learning'. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021 (cited on page 65).

[193] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., 1982 (cited on page 65).

[194] Robert D Gordon and David E Irwin. 'What's in an object file? Evidence from priming studies'. In: *Perception & Psychophysics* 58.8 (1996), pp. 1260–1277 (cited on page 65).

[195] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. 'Unsupervised state representation learning in atari'. In: *Advances in neural information processing systems* 32 (2019) (cited on page 65).

[196] Elizabeth S Spelke. 'Principles of object perception'. In: *Cognitive science* 14.1 (1990), pp. 29–56 (cited on page 65).

[197] Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. 'Model Based Reinforcement Learning for Atari'. In: *International Conference on Learning Representations*. 2019 (cited on page 65).

[198] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. 'Deep spatial autoencoders for visuomotor learning'. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 512–519 (cited on page 65).

[199] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. 'Successor features for transfer in reinforcement learning'. In: *Advances in neural information processing systems*. 2017, pp. 4055–4065 (cited on page 65).

[200] Andrea Dittadi, Frederik K Drachmann, and Thomas Bolander. 'Planning from Pixels in Atari with Learned Symbolic Representations'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 4941–4949 (cited on page 65).

[201] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. 'Decoupling representation learning from reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9870–9879 (cited on page 65).

[202] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. 'Neural scene representation and rendering'. In: *Science* 360.6394 (2018), pp. 1204–1210 (cited on pages 65, 66).

[203] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. 'Learning Invariant Representations for Reinforcement Learning without Reconstruction'. In: *International Conference on Learning Representations*. 2020 (cited on page 65).

[204] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 'Curl: Contrastive unsupervised representations for reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5639–5650 (cited on page 65).

[205] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. 'Embed to control: A locally linear latent dynamics model for control from raw images'. In: *Advances in neural information processing systems* 28 (2015) (cited on page 66).

[206] Herke Van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. 'Stable reinforcement learning with autoencoders for tactile and visual data'. In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2016, pp. 3928–3934 (cited on page 66).

[207] Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. 'Deep predictive policy training using reinforcement learning'. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 2351–2358 (cited on page 66).

[208] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. 'Visual reinforcement learning with imagined goals'. In: *Advances in Neural Information Processing Systems*. 2018 (cited on page 66).

[209] David Ha and Jürgen Schmidhuber. 'Recurrent world models facilitate policy evolution'. In: *Advances in neural information processing systems* 31 (2018) (cited on page 66).

[210] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. 'Policy gradient methods for reinforcement learning with function approximation.' In: *NIPs*. Vol. 99. Citeseer. 1999, pp. 1057–1063 (cited on page 68).

[211] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 'High-dimensional continuous control using generalized advantage estimation'. In: *arXiv preprint arXiv:1506.02438* (2015) (cited on page 68).

[212] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 'Trust region policy optimization'. In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897 (cited on page 68).

[213] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 'Deterministic policy gradient algorithms'. In: *International conference on machine learning*. PMLR. 2014, pp. 387–395 (cited on page 68).

[214] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. 'Learning to walk via deep reinforcement learning'. In: *arXiv preprint arXiv:1812.11103* (2018) (cited on page 71).

[215] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 'Deep reinforcement learning for autonomous driving: A survey'. In: *IEEE Transactions on Intelligent Transportation Systems* (2021) (cited on page 71).

[216] Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. 'End-to-end robotic reinforcement learning without reward engineering'. In: *arXiv preprint arXiv:1904.07854* (2019) (cited on page 71).

[217] Chelsea Finn, Tianhe Yu, Justin Fu, Pieter Abbeel, and Sergey Levine. 'Generalizing skills with semi-supervised reinforcement learning'. In: *International Conference on Learning Representations*. 2017 (cited on page 77).

[218] Xander Steenbrugge, Sam Leroux, Tim Verbelen, and Bart Dhoedt. 'Improving Generalization for Abstract Reasoning Tasks Using Disentangled Feature Representations'. In: *Workshop on Relational Representation Learning at NeurIPS*. 2018 (cited on page 79).

[219] Andrea Dittadi, Samuele S Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. 'Generalization and Robustness Implications in Object-Centric Learning'. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 5221–5285 (cited on page 79).

[220] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. 'Structured disentangled representations'. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 2525–2534 (cited on page 79).

[221] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 'Compositionality and Generalization in Emergent Languages'. In: *ACL 2020-8th annual meeting of the Association for Computational Linguistics*. 2020 (cited on page 79).

[222] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 'Generalizing to unseen domains: A survey on domain generalization'. In: *IEEE Transactions on Knowledge and Data Engineering* (2022) (cited on page 79).

[223] Kavosh Asadi, David Abel, and Michael L Littman. 'Learning state abstractions for transfer in continuous control'. In: *arXiv preprint arXiv:2002.05518* (2020) (cited on page 79).

[224] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. 'Deep convolutional inverse graphics network'. In: *Advances in Neural Information Processing Systems*. 2015 (cited on page 79).

[225] Markus Wulfmeier, Arunkumar Byravan, Tim Hertweck, Irina Higgins, Ankush Gupta, Tejas Kulkarni, Malcolm Reynolds, Denis Teplyashin, Roland Hafner, Thomas Lampe, et al. 'Representation matters: Improving perception and exploration for robotics'. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 6512–6519 (cited on page 79).

[226] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. 'Challenges of real-world reinforcement learning'. In: *RL4RealLife at ICML* (2019) (cited on page 79).

[227] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. 'Reinforcement learning in robotics: Applications and real-world challenges'. In: *Robotics* 2.3 (2013), pp. 122–148 (cited on page 79).

[228] Zhiyuan Chen and Bing Liu. 'Lifelong machine learning'. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12.3 (2018), pp. 1–207 (cited on pages 83, 89).

[229] Sebastian Thrun. 'Is learning the n-th thing any easier than learning the first?' In: *Advances in neural information processing systems* 8 (1995) (cited on pages 83, 89).

[230] Gido M Van de Ven and Andreas S Tolias. 'Three scenarios for continual learning'. In: *arXiv preprint arXiv:1904.07734* (2019) (cited on page 83).

[231] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 'Big self-supervised models are strong semi-supervised learners'. In: *Advances in neural information processing systems* 33 (2020), pp. 22243–22255 (cited on pages 83, 96).

[232] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. 'Big self-supervised models advance medical image classification'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3478–3488 (cited on pages 83, 96).

[233] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 'Emerging Properties in Self-Supervised Vision Transformers'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2021 (cited on pages 83, 85, 96, 125, 186).

[234] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 'Parameter-efficient transfer learning for NLP'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2790–2799 (cited on page 83).

[235] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 'Learning to prompt for vision-language models'. In: *International Journal of Computer Vision* 130.9 (2022), pp. 2337–2348 (cited on page 83).

[236] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 'Learning to prompt for continual learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 139–149 (cited on page 83).

[237] Aaron Van Den Oord, Oriol Vinyals, et al. 'Neural discrete representation learning'. In: *Advances in neural information processing systems* 30 (2017) (cited on pages 84, 86, 90, 187).

[238] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. 'Generating diverse high-fidelity images with vq-vae-2'. In: *Advances in neural information processing systems* 32 (2019) (cited on pages 84, 86, 90, 91, 187).

[239] Dianbo Liu, Alex M Lamb, Kenji Kawaguchi, Anirudh Goyal, Chen Sun, Michael C Mozer, and Yoshua Bengio. 'Discrete-valued neural communication'. In: *Advances in Neural Information Processing Systems* 34 (2021) (cited on pages 84, 90, 91).

[240] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 'Unsupervised learning of visual features by contrasting cluster assignments'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9912–9924 (cited on pages 85, 186).

[241] Asher Trockman and J Zico Kolter. 'Patches Are All You Need?' In: *Transactions on Machine Learning Research* (2023) (cited on page 85).

[242] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 'Learning transferable visual models from natural language supervision'. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763 (cited on pages 85, 186).

[243] Jonathan Baxter. 'A model of inductive bias learning'. In: *Journal of artificial intelligence research* 12 (2000), pp. 149–198 (cited on page 89).

[244] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012 (cited on page 89).

[245] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 'Improving language models by retrieving from trillions of tokens'. In: *International conference on machine learning*. PMLR. 2022, pp. 2206–2240 (cited on page 89).

[246] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 'WebGPT: Browser-assisted question-answering with human feedback'. In: *arXiv preprint arXiv:2112.09332* (2021) (cited on page 89).

[247] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 'Latent retrieval for weakly supervised open domain question answering'. In: *arXiv preprint arXiv:1906.00300* (2019) (cited on page 89).

[248] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 'Retrieval-augmented generation for knowledge-intensive nlp tasks'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474 (cited on page 89).

[249] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 'Retrieval augmented language model pre-training'. In: *International conference on machine learning*. PMLR. 2020, pp. 3929–3938 (cited on page 89).

[250] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 'Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation'. In: *arXiv preprint arXiv:2107.02137* (2021) (cited on page 89).

[251] Anirudh Goyal, Abram Friesen, Andrea Banino, Theophane Weber, Nan Rosemary Ke, Adria Puigdomenech Badia, Arthur Guez, Mehdi Mirza, Peter C Humphreys, Ksenia Konyushova, et al. 'Retrieval-augmented reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 7740–7765 (cited on page 89).

[252] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 'Distilling the knowledge in a neural network'. In: *Deep Learning Workshop at NIPS* 2.7 (2014) (cited on page 89).

[253] Guy Davidson and Michael C Mozer. 'Sequential mastery of multiple visual tasks: Networks naturally learn to learn and forget to forget'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9282–9293 (cited on page 89).

[254] Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer New York, 1996 (cited on page 89).

[255] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 'icarl: Incremental classifier and representation learning'. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 2001–2010 (cited on page 89).

[256] Zhizhong Li and Derek Hoiem. 'Learning without forgetting'. In: *IEEE transactions on pattern analysis and machine intelligence* 40.12 (2017), pp. 2935–2947 (cited on page 89).

[257] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 'Continual learning with deep generative replay'. In: *Advances in neural information processing systems* 30 (2017) (cited on page 89).

[258] Yu-Ming Tang, Yi-Xing Peng, and Wei-Shi Zheng. 'Learning to imagine: Diversify memory for incremental learning using unlabeled data'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9549–9558 (cited on page 89).

[259] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 'Overcoming catastrophic forgetting in neural networks'. In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526 (cited on page 89).

[260] Friedemann Zenke, Ben Poole, and Surya Ganguli. 'Continual learning through synaptic intelligence'. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3987–3995 (cited on page 89).

[261] Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. 'Error-driven incremental learning in deep convolutional neural network for large-scale image classification'. In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 177–186 (cited on pages 89, 90).

[262] Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. 'Efficient feature transformations for discriminative and generative continual learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13865–13875 (cited on pages 89, 90).

[263] Qiang Gao, Zhipeng Luo, Diego Klabjan, and Fengli Zhang. 'Efficient architecture search for continual learning'. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022) (cited on pages 89, 90).

[264] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 'Overcoming catastrophic forgetting with hard attention to the task'. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4548–4557 (cited on pages 89, 90).

[265] Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 'Achieving Forgetting Prevention and Knowledge Transfer in Continual Learning'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22443–22456 (cited on pages 89, 90).

[266] James Harrison, Apoorva Sharma, Chelsea Finn, and Marco Pavone. 'Continuous meta-learning without tasks'. In: *Advances in neural information processing systems* 33 (2020), pp. 17571–17581 (cited on pages 89, 90).

[267] Xu He, Jakub Sygnowski, Alexandre Galashov, Andrei A Rusu, Yee Whye Teh, and Razvan Pascanu. 'Task agnostic continual learning via meta learning'. In: *arXiv preprint arXiv:1906.05201* (2019) (cited on pages 89, 90).

[268] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. 'Online meta-learning'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1920–1930 (cited on pages 89, 90).

[269] Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988 (cited on pages 89, 90).

[270] Pentti Kanerva. *Sparse distributed memory and related models*. Tech. rep. 1992 (cited on pages 89, 90).

[271] Trenton Bricken, Xander Davies, Deepak Singh, Dmitry Krotov, and Gabriel Kreiman. 'Sparse Distributed Memory is a Continual Learner'. In: *International Conference for Learning Representations (ICLR)* (2023) (cited on pages 89, 93, 94, 188, 189, 192, 197).

[272] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. 'Continual learning of context-dependent processing in neural networks'. In: *Nature Machine Intelligence* 1.8 (2019), pp. 364–372 (cited on pages 89, 90).

[273] Gobinda Saha, Isha Garg, and Kaushik Roy. 'Gradient Projection Memory for Continual Learning'. In: *International Conference on Learning Representations*. 2021 (cited on pages 89, 90).

[274] Yang Shen, Sanjoy Dasgupta, and Saket Navlakha. 'Algorithmic insights on continual learning from fruit flies'. In: *arXiv preprint arXiv:2107.07617* (2021) (cited on pages 89, 188, 189).

[275] Julien Pourcel, Ngoc-Son Vu, and Robert M French. 'Online Task-free Continual Learning with Dynamic Sparse Distributed Memory'. In: *European Conference on Computer Vision*. Springer. 2022, pp. 739–756 (cited on page 89).

[276] Léon Bottou and Vladimir Vapnik. 'Local learning algorithms'. In: *Neural computation* 4.6 (1992), pp. 888–900 (cited on page 90).

[277] Mohammadamin Banayeeanzade, Rasoul Mirzaiezadeh, Hosein Hasani, and Mahdieh Soleymani. 'Generative vs. Discriminative: Rethinking The Meta-Continual Learning'. In: *Advances in Neural Information Processing Systems* 34 (2021) (cited on page 90).

[278] Oleksiy Ostapenko, Timothee Lesort, Pau Rodriguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. 'Continual learning with foundation models: An empirical study of latent replay'. In: *Conference on Lifelong Learning Agents*. PMLR. 2022, pp. 60–91 (cited on page 90).

[279] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 'Soundstream: An end-to-end neural audio codec'. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2021) (cited on pages 90, 91).

[280] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 'Vector-quantized Image Modeling with Improved VQGAN'. In: *International Conference on Learning Representations*. 2022 (cited on pages 90, 91).

[281] Woncheol Shin, Gyubok Lee, Jiyoung Lee, Joonseok Lee, and Edward Choi. 'Translation-equivariant Image Quantizer for Bi-directional Image-Text Generation'. In: *arXiv preprint arXiv:2112.00384* (2021) (cited on pages 90, 91).

[282] Rayhane Mama, Marc S Tyndel, Hashiam Kadhim, Cole Clifford, and Ragavan Thurairatnam. 'NWT: Towards natural audio-to-video generation with representation learning'. In: *arXiv preprint arXiv:2106.04283* (2021) (cited on pages 90, 91).

[283] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 'End-to-end memory networks'. In: *Advances in neural information processing systems* 28 (2015) (cited on pages 90, 91).

[284] Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. 'Hierarchical memory networks'. In: *arXiv preprint arXiv:1605.07427* (2016) (cited on pages 90, 91).

[285] Taylor Whittington Webb, Ishan Sinha, and Jonathan Cohen. 'Emergent Symbols through Binding in External Memory'. In: *International Conference on Learning Representations*. 2021 (cited on pages 90, 91).

[286] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 'Large memory layers with product keys'. In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on pages 90, 91).

[287] Rina Panigrahy, Xin Wang, and Manzil Zaheer. 'Sketch based Memory for Neural Networks'. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3169–3177 (cited on pages 90, 91).

[288] Adam Coates, Andrew Ng, and Honglak Lee. 'An analysis of single-layer networks in unsupervised feature learning'. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 215–223 (cited on page 96).

[289] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 'How transferable are features in deep neural networks?' In: *Advances in neural information processing systems* 27 (2014) (cited on page 96).

[290] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. 'Unsupervised deep learning by neighbourhood discovery'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2849–2858 (cited on page 96).

[291] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. 'Understanding the robustness in vision transformers'. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 27378–27394 (cited on page 96).

[292] Rui Tian, Zuxuan Wu, Qi Dai, Han Hu, and Yugang Jiang. 'Deeper Insights into ViTs Robustness towards Common Corruptions'. In: *arXiv preprint arXiv:2204.12143* (2022) (cited on page 96).

[293] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 'Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution'. In: *International Conference on Learning Representations*. 2022 (cited on page 96).

[294] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 'Self-training with noisy student improves imagenet classification'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020 (cited on page 101).

[295] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. 'Fixing the train-test resolution discrepancy: Fixefficientnet'. In: *Advances in Neural Information Processing Systems 32* (2019) (cited on page 101).

[296] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 'Sharpness-aware Minimization for Efficiently Improving Generalization'. In: *International Conference on Learning Representations*. 2021 (cited on page 101).

[297] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 'Meta pseudo labels'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 11557–11568 (cited on page 101).

[298] Joelle Pineau, Genevieve Fried, Rosemary Nan Ke, and Hugo Larochelle. 'ICLR2018 reproducibility challenge'. In: *ICLR workshop on Reproducibility in Machine Learning*. 2018 (cited on page 101).

[299] Koustuv Sinha, Joelle Pineau, Jessica Forde, Rosemary Nan Ke, and Hugo Larochelle. 'NeurIPS 2019 Reproducibility Challenge'. In: *ReScience C* 6.2 (2020) (cited on page 101).

[300] Joelle Pineau, Koustuv Sinha, Genevieve Fried, Rosemary Nan Ke, and Hugo Larochelle. 'ICLR Reproducibility Challenge 2019'. In: *ReScience C* 5.2 (2019) (cited on page 101).

[301] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. 'Updates in Human-AI teams: Understanding and addressing the performance/compatibility tradeoff'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019 (cited on pages 102, 103, 110, 111, 124).

[302] Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. 'Positive-Congruent Training: Towards Regression-Free Model Updates'. In: *Conference on Computer Vision and Pattern Recognition* (2021) (cited on pages 103, 110, 124).

[303] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. 'Towards backward-compatible representation learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020 (cited on pages 103, 124).

[304] Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. 'An Empirical Analysis of Backward Compatibility in Machine Learning Systems'. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020 (cited on pages 103, 111, 113, 124, 199).

[305] Hidetoshi Shimodaira. 'Improving predictive inference under covariate shift by weighting the log-likelihood function'. In: *Journal of statistical planning and inference* 90.2 (2000) (cited on pages 105, 109).

[306] Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012 (cited on pages 105, 109).

[307] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. 'Domain adaptation under target and conditional shift'. In: *International Conference on Machine Learning (ICML)*. PMLR. 2013, pp. 819–827 (cited on page 105).

[308] Amos Storkey. 'When training and test sets are different: characterizing learning transfer'. In: *Dataset shift in machine learning* 30 (2009), pp. 3–28 (cited on page 105).

[309] Claude E Shannon. 'A mathematical theory of communication'. In: *The Bell system technical journal* 27.3 (1948) (cited on page 107).

[310] David JC MacKay. 'Bayesian neural networks and density networks'. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 354.1 (1995) (cited on pages 109, 217).

[311] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 'Intriguing properties of neural networks'. In: *International Conference on Learning Representations* (2014) (cited on pages 109, 217).

[312] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 'Explaining and harnessing adversarial examples'. In: *International Conference on Learning Representations* (2015) (cited on pages 109, 217).

[313] Anh Nguyen, Jason Yosinski, and Jeff Clune. 'Deep neural networks are easily fooled: High confidence predictions for unrecognizable images'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015 (cited on pages 109, 217).

[314] Sinno Jialin Pan and Qiang Yang. 'A survey on transfer learning'. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359 (cited on page 109).

[315] Yaroslav Ganin and Victor Lempitsky. 'Unsupervised domain adaptation by backpropagation'. In: *International conference on machine learning*. PMLR. 2015 (cited on page 109).

[316] Irving J Good. 'The population frequencies of species and the estimation of population parameters'. In: *Biometrika* 40.3-4 (1953) (cited on page 109).

[317] Herbert Robbins. 'An empirical Bayes approach to statistics'. In: *Proc. 3rd Berkeley Symp. Math. Statist. Probab., 1956*. Vol. 1. 1956 (cited on page 109).

[318] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. 'Are we done with ImageNet?' In: *arXiv preprint arXiv:2006.07159* (2020) (cited on page 110).

[319] Armen Der Kiureghian and Ove Ditlevsen. 'Aleatory or epistemic? Does it matter?' In: *Structural safety* 31.2 (2009) (cited on page 110).

[320] Alex Kendall and Yarin Gal. 'What uncertainties do we need in Bayesian deep learning for computer vision?' In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017 (cited on page 110).

[321] Dennis V Lindley. 'On a measure of the information provided by an experiment'. In: *The Annals of Mathematical Statistics* (1956) (cited on page 110).

[322] Burr Settles. 'Active learning literature survey'. In: *University of Wisconsin, Madison, Department of Computer Sciences* (2009) (cited on page 110).

[323] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 'Active learning from imperfect labelers'. In: *Advances in Neural Information Processing Systems 29* (2016) (cited on page 110).

[324] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cited on page 110).

[325] Yoav Freund and Robert E Schapire. 'A decision-theoretic generalization of on-line learning and an application to boosting'. In: *Journal of computer and system sciences* 55.1 (1997) (cited on page 111).

[326] Leo Breiman. 'Bagging predictors'. In: *Machine learning* 24.2 (1996) (cited on page 111).

[327] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. 'Bayesian model averaging: a tutorial'. In: *Statistical science* (1999) (cited on page 111).

[328] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014 (cited on page 111).

[329] Shmuel Nitzan and Jacob Paroush. 'Optimal decision rules in uncertain dichotomous choice situations'. In: *International Economic Review* (1982) (cited on page 111).

[330] Daniel Berend and Aryeh Kontorovich. 'A finite sample analysis of the Naive Bayes classifier.' In: *J. Mach. Learn. Res.* 16.1 (2015) (cited on page 111).

[331] Hyun-Chul Kim and Zoubin Ghahramani. 'Bayesian classifier combination'. In: *Artificial Intelligence and Statistics*. 2012 (cited on page 111).

[332] Philip J Boland, Frank Proschan, and Yung Liang Tong. 'Modelling dependence in simple and indirect majority systems'. In: *Journal of Applied Probability* (1989) (cited on page 111).

[333] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 'Imagenet: A large-scale hierarchical image database'. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009 (cited on page 111).

[334] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 'SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size'. In: *arXiv preprint arXiv:1602.07360* (2016) (cited on page 112).

[335] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 'Going deeper with convolutions'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015 (cited on page 112).

[336] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 'Rethinking the inception architecture for computer vision'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016 (cited on page 112).

[337] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 'Mobilenetv2 Inverted residuals and linear bottlenecks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018 (cited on page 112).

[338] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 'Densely connected convolutional networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017 (cited on page 112).

[339] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 'Aggregated residual transformations for deep neural networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017 (cited on page 112).

[340] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on page 112).

[341] Huy Phan. *huyvnphan/PyTorch_CIFAR10*. Version v3.0.1. 2021. DOI: 10.5281/zenodo.4431043 (cited on page 112).

[342] Christina M Funke, Paul Vicol, Kuan-Chieh Wang, Matthias Kümmerer, Richard Zemel, and Matthias Bethge. 'Disentanglement and generalization under correlation shifts'. In: *Conference on Lifelong Learning Agents*. PMLR. 2022, pp. 116–141 (cited on page 122).

[343] Tobias Leemann, Michael Kirchhof, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci. 'Disentangling embedding spaces with minimal distributional assumptions'. In: *arXiv preprint arXiv:2206.13872* (2022) (cited on page 122).

[344] Yixin Wang and Michael I Jordan. 'Desiderata for representation learning: A causal perspective'. In: *Advances in neural information processing systems* (2021) (cited on page 122).

[345] Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. 'Disentanglement of Correlated Factors via Hausdorff Factorized Support'. In: *The Eleventh International Conference on Learning Representations*. 2023 (cited on page 122).

[346] Sumedh A Sontakke, Arash Mehrjou, Laurent Itti, and Bernhard Schölkopf. 'Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning'. In: *International conference on machine learning*. PMLR. 2021, pp. 9848–9858 (cited on page 122).

[347] Jikun Kang, Miao Liu, Abhinav Gupta, Christopher Pal, Xue Liu, and Jie Fu. 'Learning Multi-Objective Curricula for Robotic Policy Learning'. In: *Conference on Robot Learning*. PMLR. 2023, pp. 847–858 (cited on page 122).

[348] Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. 'Generalizing goal-conditioned reinforcement learning with variational causal reasoning'. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 26532–26548 (cited on page 122).

[349] Arthur Allshire, Mayank MittaI, Varun Lodaya, Viktor Makoviychuk, Denys Makoviichuk, Felix Widmaier, Manuel Wüthrich, Stefan Bauer, Ankur Handa, and Animesh Garg. 'Transferring dexterous manipulation from gpu simulation to a remote real-world trifinger'. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 11802–11809 (cited on page 122).

[350] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. 'Self-supervised Visual Reinforcement Learning with Object-centric Representations'. In: (2021) (cited on page 122).

[351] Richard Li, Allan Jabri, Trevor Darrell, and Pulkit Agrawal. 'Towards Practical Multi-Object Manipulation using Relational Reinforcement Learning'. In: *arXiv preprint arXiv:1912.11032* (2019) (cited on page 122).

[352] Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. 'Causal influence detection for improving efficiency in reinforcement learning'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22905–22918 (cited on page 122).

[353] Adrien Ali Taiga, Rishabh Agarwal, Jesse Farebrother, Aaron Courville, and Marc G Bellemare. 'Investigating multi-task pretraining and generalization in reinforcement learning'. In: *The Eleventh International Conference on Learning Representations*. 2023 (cited on page 123).

[354] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. 'An investigation into pre-training object-centric representations for reinforcement learning'. In: *International Conference on Machine Learning* (2023) (cited on page 123).

[355] Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 'Aging with grace: Lifelong model editing with discrete key-value adaptors'. In: *Advances in Neural Information Processing Systems* 36 (2024) (cited on page 123).

[356] Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, Luo Si, and Yongbin Li. 'Lifelong learning for question answering with hierarchical prompts'. In: *arXiv preprint arXiv:2208.14602* (2022) (cited on page 123).

[357] Avinash Kori, Francesco Locatello, Fabio De Sousa Ribeiro, Francesca Toni, and Ben Glocker. 'Grounded Object-Centric Learning'. In: *The Twelfth International Conference on Learning Representations*. 2024 (cited on page 123).

[358] Rahul Duggal, Hao Zhou, Shuo Yang, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. 'Compatibility-aware heterogeneous visual search'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 10723–10732 (cited on page 124).

[359] Rahul Duggal, Hao Zhou, Shuo Yang, Jun Fang, Yuanjun Xiong, and Wei Xia. 'Towards regression-free neural networks for diverse compute platforms'. In: *European Conference on Computer Vision*. Springer. 2022, pp. 598–614 (cited on page 124).

[360] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 'Adversarial training for free!' In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on page 124).

[361] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. 'Improving robustness against common corruptions by covariate shift adaptation'. In: *Advances in neural information processing systems* 33 (2020), pp. 11539–11551 (cited on page 124).

[362] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 'A simple framework for contrastive learning of visual representations'. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607 (cited on page 124).

[363] Ishan Misra and Laurens van der Maaten. 'Self-supervised learning of pretext-invariant representations'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6707–6717 (cited on page 124).

[364] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 'Bootstrap your own latent-a new approach to self-supervised learning'. In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284 (cited on page 124).

[365] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. 'Visda: The visual domain adaptation challenge'. In: *arXiv preprint arXiv:1710.06924* (2017) (cited on pages 124, 125).

[366] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. 'Robustbench: a standardized adversarial robustness benchmark'. In: *arXiv preprint arXiv:2010.09670* (2020) (cited on pages 124, 125).

[367] Nico Gürtler, Felix Widmaier, Cansu Sancaktar, Sebastian Blaes, Pavel Kolev, Stefan Bauer, Manuel Wüthrich, Markus Wulfmeier, Martin Riedmiller, Arthur Allshire, et al. 'Real robot challenge 2022: Learning dexterous manipulation from offline data in the real world'. In: *NeurIPS 2022 Competition Track*. PMLR. 2023, pp. 133–150 (cited on pages 124, 125).

[368] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 'End-to-end object detection with transformers'. In: *European conference on computer vision*. Springer. 2020, pp. 213–229 (cited on page 125).

[369] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 'Masked autoencoders are scalable vision learners'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16000–16009 (cited on page 125).

[370] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 'Opt: Open pre-trained transformer language models'. In: *arXiv preprint arXiv:2205.01068* (2022) (cited on page 125).

[371] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 'Dinov2: Learning robust visual features without supervision'. In: *arXiv preprint arXiv:2304.07193* (2023) (cited on page 125).

[372] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 'Segment anything'. In: *arXiv preprint arXiv:2304.02643* (2023) (cited on page 125).

[373] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 'Palm-e: An embodied multimodal language model'. In: *International Conference on Machine Learning* (2023) (cited on page 125).

[374] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 'Llama: Open and efficient foundation language models'. In: *arXiv preprint arXiv:2302.13971* (2023) (cited on page 125).

[375] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 'LoRA: Low-Rank Adaptation of Large Language Models'. In: *International Conference on Learning Representations*. 2022 (cited on pages 125, 126).

[376] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 'Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment'. In: *arXiv preprint arXiv:2312.12148* (2023) (cited on pages 125, 126).

[377] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 'Sparks of artificial general intelligence: Early experiments with gpt-4'. In: *arXiv preprint arXiv:2303.12712* (2023) (cited on page 126).

[378] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. 'Cladder: Assessing causal reasoning in language models'. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023 (cited on page 126).

[379] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 'Evaluating the logical reasoning ability of chatgpt and gpt-4'. In: *arXiv preprint arXiv:2304.03439* (2023) (cited on page 126).

[380] Kiho Park, Yo Joong Choe, and Victor Veitch. 'The Linear Representation Hypothesis and the Geometry of Large Language Models'. In: *Causal Representation Learning Workshop at NeurIPS 2023*. 2023 (cited on page 126).

[381] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 'Causal reasoning and large language models: Opening a new frontier for causality'. In: *arXiv preprint arXiv:2305.00050* (2023) (cited on page 126).

[382] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 'Emergent abilities of large language models'. In: *arXiv preprint arXiv:2206.07682* (2022) (cited on page 126).

[383] Anthropic. 'The Claude 3 Model Family: Opus, Sonnet, Haiku'. In: (2024) (cited on page 126).

[384] Hugo Fitipaldi and Paul W Franks. 'Ethnic, gender and other sociodemographic biases in genome-wide association studies for the most burdensome non-communicable diseases: 2005–2022'. In: *Human Molecular Genetics* 32.3 (2023), pp. 520–532 (cited on page 127).

[385] Ben Hayes. 'Overview of statistical methods for genome-wide association studies (GWAS)'. In: *Genome-wide Association Studies and Genomic Prediction* (2013), pp. 149–169 (cited on page 132).

[386] Cathryn M Lewis and Evangelos Vassos. 'Polygenic risk scores: from research tools to clinical instruments'. In: *Genome Medicine* 12.1 (2020), pp. 1–11 (cited on page 132).

[387] Kelly A Frazer, Sarah S Murray, Nicholas J Schork, and Eric J Topol. 'Human genetic variation and its contribution to complex traits'. In: *Nature Reviews Genetics* 10.4 (2009), pp. 241–251 (cited on page 132).

[388] Matthew R Nelson, Hannah Tipney, Jeffery L Painter, Judong Shen, Paola Nicoletti, Yufeng Shen, Aris Floratos, Pak Chung Sham, Mulin Jun Li, Junwen Wang, et al. 'The support of human genetic evidence for approved drug indications'. In: *Nature Genetics* 47.8 (2015), pp. 856–860 (cited on page 132).

[389] Melina Claussnitzer, Judy H Cho, Rory Collins, Nancy J Cox, Emmanouil T Dermitzakis, Matthew E Hurles, Sekar Kathiresan, Eimear E Kenny, Cecilia M Lindgren, Daniel G MacArthur, et al. 'A brief history of human disease genetics'. In: *Nature* 577.7789 (2020), pp. 179–189 (cited on page 132).

[390] Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. 'The role of Disentanglement in Generalisation'. In: *International Conference on Learning Representations*. 2021 (cited on page 136).

[391] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 'A Sober Look at the Unsupervised Learning of Disentangled Representations and their Evaluation'. In: *Journal of Machine Learning Research* 21.209 (2020), pp. 1–62 (cited on pages 139, 144).

[392] Safa Alver and Doina Precup. 'A Brief Look at Generalization in Visual Meta-Reinforcement Learning'. In: *LifelongML Workshop at ICML* (2020) (cited on page 162).

[393] Diederik P Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014) (cited on page 163).

[394] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. *Stable Baselines*. https://github.com/hill-a/stable-baselines. 2018 (cited on page 173).

[395] Kenji Kawaguchi, Zhun Deng, Kyle Luh, and Jiaoyang Huang. 'Robustness implies generalization via data-dependent generalization bounds'. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 10866–10894 (cited on page 184).

[396] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. 'Combined Scaling for Open-Vocabulary Image Classification'. In: *arXiv preprint arXiv:2111.10050* (2021) (cited on page 184).

[397] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 'Imagenet large scale visual recognition challenge'. In: *International journal of computer vision* 115.3 (2015), pp. 211–252 (cited on page 186).

[398] Niklas Pfister, Peter Bühlmann, Bernhard Schölkopf, and Jonas Peters. 'Kernel-based tests for joint independence'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80.1 (2018), pp. 5–31 (cited on page 199).