



# THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e. g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

# **Shift happens: How can machine learning systems be best prepared?**

*Cian Eastwood*



Doctor of Philosophy  
Institute for Adaptive and Neural Computation  
School of Informatics  
The University of Edinburgh  
2023



# Abstract

Machine learning systems have made headlines in recent years, defeating world champions in Go, enhancing medical diagnoses, and redefining how we work with tools like ChatGPT. However, despite these impressive feats, machine learning systems remain fragile when faced with test data that differs from their training data. This fragility stems from a fundamental mismatch between textbook machine-learning methods and their real-world application. While textbook methods assume that the conditions under which a system is developed are similar to those in which it is deployed, in reality, systems tend to be developed under one set of conditions (e.g., in a lab) and deployed to another (e.g., a clinic). As a result, many machine learning systems are not prepared for the condition differences or *distribution shifts* they face upon deployment, leading to some high-profile and costly failures. For safety-critical settings like healthcare and autonomous driving, such failures represent a major barrier to real-world deployment.

In this thesis, I argue that we must first accept that shift happens, and subsequently focus on how we can best prepare. To do so, I present four of my works that illustrate how machine learning systems can be prepared for (and adapted to) real-world distribution shifts. Together, these contributions take us closer to reliable machine learning systems that can be deployed in safety-critical settings.

In the first work, the setting is source-free domain adaptation, i.e., adapting a model to unlabelled test data without the original training data. Here, we prepare for a change in measurement device (e.g., X-rays from a different scanner) by storing lightweight statistics of the training data. By restoring these statistics on the test data, we see improved accuracy, calibration and data efficiency over prior methods.

In the second work, the setting is domain generalisation, i.e., performing well on test data from new environments or *domains* by leveraging data from multiple related domains at training time. Here, we prepare for more flexible and unknown changes by exploiting invariances across the training domains that hold *with high probability* in unseen test domains. In particular, by minimising a particular quantile of a model’s performance distribution over domains, we learn models that perform well with the corresponding probability.

In the third work, the setting is again domain generalisation, but this time we focus on ways to harness so-called “spurious” features *without test-domain labels*. In particular, we show that predictions based on invariant/*stable* features can be used to adapt our usage of spurious/*unstable* features to new test domains, so long as the stable and



unstable features are *complementary* (i.e., conditionally independent given the label). By safely harnessing complementary spurious features, we boost performance without sacrificing robustness.

Finally, in the fourth work, the setting is disentangled representation learning which, in the context of this thesis, can be viewed as preparing for a change in the task itself by recovering and separating the underlying factors of variation. To this end, we extend an existing evaluation framework by first introducing a measure of representation *explicitness* or *ease of use*, and then connecting the framework to identifiability.

# Lay summary

Machine learning systems have made headlines in recent years, defeating world champions in Go, enhancing medical diagnoses, and redefining how we work with tools like ChatGPT. However, despite these impressive feats, machine learning systems remain fragile when faced with test data that differs from their training data. This fragility stems from a fundamental mismatch between textbook machine-learning methods and their real-world application. While textbook methods assume that the conditions under which a system is developed are similar to those in which it is deployed, in reality, systems tend to be developed under one set of conditions (e.g., in a lab) and deployed to another (e.g., a clinic). As a result, many machine learning systems are not prepared for the condition differences or *distribution shifts* they face upon deployment, leading to some high-profile and costly failures. For safety-critical settings like healthcare and autonomous driving, such failures represent a major barrier to real-world deployment.

In this thesis, I argue that we must first accept that shift happens, and subsequently focus on how we can best prepare. To do so, I present four of my works that illustrate how machine learning systems can be prepared for (and adapted to) real-world distribution shifts. Together, these contributions take us closer to reliable machine learning systems that can be deployed in safety-critical settings.

In the first work, we prepare for a change in measurement device (e.g., X-rays from a different scanner) by storing lightweight statistics of the training data. This allows us to adapt a model trained on data from one setting (e.g., hospital A) to perform well on unlabelled data from another setting (e.g., hospital B, which has a different scanner). In the second work, we handle more flexible and unknown changes by leveraging data from multiple related settings (e.g., multiple hospitals). This allows us to train a model that performs well in new settings (e.g., hospitals) *with high probability*, eliminating the need for adaptation. In the third work, we address changes in unstable or “spurious” correlations in the data. We first separate stable and unstable correlations and then use the stable ones to *guide our use of the unstable ones*. This allows us to train a model that correctly uses both types of correlations in new settings, boosting performance without sacrificing robustness. Finally, in the fourth work, we prepare for changes in the task itself. To do so, we evaluate the quality of learned data representations, aiming to find representations that are *easy to use*.

# Acknowledgements

This thesis would not have been possible without the support and guidance of my supervisors, peers, friends and family.

I want to start by sincerely thanking my supervisor at The University of Edinburgh, Chris Williams, for his support over the past five-and-a-half years. I learned a lot from our interactions and they have undoubtedly shaped me as a researcher. I also want to thank my supervisor at The Max Planck Institute for Intelligent Systems, Bernhard Schölkopf, for his support and direction during the latter half of my PhD, as well as the other members of my supervision panel, Matthias Hennig and Timothy Hospedales, for their helpful feedback and support during my annual reviews.

Next, I want to thank my peers and colleagues for the interactions, discussions, brainstorming, coffee breaks, getaways and venting sessions that made the PhD both fulfilling and enjoyable—you know who you are! In particular, I'd like to send special thanks to two colleagues-turned-friends, Ian Mason and Julius von Kügelgen. Ian, had we not joined forces, I think I'd still be circling George Square, working on unit-level surprise, and saying "I can't believe it's not better". Julius, while short, our time together in Tübingen helped ease my transition and represents a stand-out memory of my PhD.

Perhaps most importantly, I want to thank my friends and family for all their love and support over the past five-and-a-half years. From quelling the occasional existential crisis to providing timely reminders of the important things in life: it is safe to say I would not have completed this journey without you all. In particular, Nora, ...

Finally, I want to thank the School of Informatics at The University of Edinburgh for their financial support and to give credit to the organisers of the ICML 2022 workshop "Shift happens: ..." for the title of this thesis.

# Publications

The following publications of mine feature prominently within this dissertation. Each chapter details the relevant relations to these publications.

1. Eastwood, C., Mason, I., Williams, C. K. I., and Schölkopf, B. (2022a). Source-free adaptation to measurement shift via bottom-up feature restoration. In *The Tenth International Conference on Learning Representations*
2. Eastwood, C., Robey, A., Singh, S., Kügelgen, J. V., Hassani, H., Pappas, G. J., and Schölkopf, B. (2022c). Probable domain generalization via quantile risk minimization. In *Advances in Neural Information Processing Systems*, volume 35, pages 17340–17358
3. Eastwood, C., Singh, S., Nicolicioiu, L. A., Von Kügelgen, J., and Schölkopf, B. (2023b). Spuriousity didn't kill the classifier: Using invariant predictions to harness spurious features. In *Advances in Neural Information Processing Systems*
4. Eastwood, C., Nicolicioiu, A. L., Kügelgen, J. V., Kekić, A., Träuble, F., Dittadi, A., and Schölkopf, B. (2023a). DCI-ES: An extended disentanglement framework with connections to identifiability. In *The Eleventh International Conference on Learning Representations*

In addition, the following publications were part of my PhD research, but do not feature prominently within this dissertation:

5. Eastwood, C., von Kügelgen, J., Ericsson, L., Bouchacourt, D., Vincent, P., Schölkopf, B., and Ibrahim, M. (2023c). Self-supervised disentanglement by leveraging structure in data augmentations. *Preprint arXiv:2311.08815*
6. Li, N., Eastwood, C., and Fisher, R. (2020b). Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems*, volume 33, pages 5656–5666
7. Eastwood, C., Mason, I., and Williams, C. K. I. (2021). Unit-level surprise in neural networks. In *NeurIPS 2021 Workshop "I (Still) Can't Believe It's Not Better!"*, volume 163 of *Proceedings of Machine Learning Research*, pages 33–40
8. Eastwood, C., Nanbo, L., and Williams, C. K. I. (2022b). Align-Deform-Subtract: an interventional framework for explaining object differences. In *ICLR 2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*

# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

*(Cian Eastwood, Edinburgh, 2023)*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Outline . . . . .	3
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Domain Adaptation . . . . .	5
2.1.1	Few-shot domain adaptation . . . . .	6
2.1.2	Unsupervised domain adaptation . . . . .	6
2.1.3	Source-free domain adaptation . . . . .	6
2.2	Domain Generalisation . . . . .	7
2.3	Disentangled Representation Learning . . . . .	9
2.4	Meta-Learning . . . . .	11
2.5	Summary . . . . .	11
<b>3</b>	<b>Source-Free Domain Adaptation</b>	<b>13</b>
3.1	Contribution . . . . .	13
3.2	Paper . . . . .	14
3.3	Comments on the paper . . . . .	29
<b>4</b>	<b>Domain Generalisation: A Probabilistic Framework</b>	<b>31</b>
4.1	Contribution . . . . .	32
4.2	Paper . . . . .	32
4.3	Comments on the paper . . . . .	51
<b>5</b>	<b>Domain Generalisation: Harnessing Spurious Features</b>	<b>53</b>
5.1	Contribution . . . . .	54
5.2	Paper . . . . .	54
5.3	Comments on the paper . . . . .	70

<b>6</b>	<b>Disentangled Representation Learning</b>	<b>73</b>
6.1	Contribution . . . . .	74
6.2	Paper . . . . .	74
6.3	Comments on the paper . . . . .	87
<b>7</b>	<b>Conclusions</b>	<b>89</b>
7.1	Future directions . . . . .	90
	<b>Bibliography</b>	<b>93</b>
<b>A</b>	<b>Paper Appendices</b>	<b>105</b>
A.1	Source-Free Domain Adaptation (§ 3.2) . . . . .	105
A.2	Domain Generalisation: A Probabilistic Framework (§ 4.2) . . . . .	129
A.3	Domain Generalisation: Harnessing Spurious Features (§ 5.2) . . . . .	158
A.4	Disentangled Representations (§ 6.2) . . . . .	178

# 1

## Introduction

Machine learning systems have made headlines in recent years, defeating world champions in Go (Silver et al., 2016), enhancing medical diagnoses (Singhal et al., 2023), and redefining how we work with tools like ChatGPT (Brown et al., 2020). However, despite these impressive feats, machine learning systems remain fragile when faced with test data that is subtly different from the training data—be it due to changes in location (e.g., X-rays from new hospitals, Zech et al. 2018), time (e.g., more recent satellite images, Hansen et al. 2013), sub-populations (e.g., text from different demographic groups, Borkan et al. 2019) or other naturally-occurring variations (e.g., common data corruptions, Hendrycks and Dietterich 2019). These failures are of particular concern in safety-critical applications such as healthcare (Beede et al., 2020; Jovicich et al., 2009) and autonomous driving (Dai and Van Gool, 2018; Michaelis et al., 2019), where they represent one of the most significant barriers to the real-world deployment of machine learning systems (Koh et al., 2021).

At the heart of this issue is a fundamental mismatch between textbook machine-learning methods and their real-world application. On the one hand, textbook machine-learning methods assume that the conditions under which a model is developed are similar to those in which it is deployed (Storkey, 2009), or, more precisely, that the training and test data come from the same distribution (Vapnik, 1998)<sup>1</sup>. On the other hand, real-world machine learning systems tend to be developed under one set of conditions (e.g., in a lab) and then deployed to another (e.g., a clinic), leading to a *shift* in the data distribution between training and test domains (Quiñonero-Candela et al., 2008). As a result of this mismatch, many machine learning systems are *unprepared for the distribution shift that they (inevitably) encounter upon deployment in the test domain*

---

<sup>1</sup>Through the assumption of *i.i.d.* (independent and identically distributed) data.



*of interest*, leading to the aforementioned failures.

To address this issue, and ultimately close the gap between textbook machine learning methods and their real-world application, I argue that we must first accept that *shift happens* and subsequently focus our attention on how we can best prepare. To do so, I present four of my works that illustrate how machine learning systems can be prepared for (and adapted to) real-world distribution shifts. Together, these contributions take us closer to reliable machine-learning systems that can be deployed in safety-critical settings.

In the first work, the setting is domain adaptation, in particular, *source-free domain adaptation* (SFDA, [Li et al. 2020c](#)). Here, a model is adapted to unlabelled and previously-unseen test data *without access to the original training data*. This problem can arise when deploying healthcare models to new hospitals (due to privacy regulations) or deploying image/language models to mobile devices (due to storage constraints). We address this problem for one particular type of distribution shift, termed measurement shift, that stems from a change in measurement device (e.g., X-rays from a different scanner) and can therefore be resolved by restoring the same features on the test data (rather than learning new ones). To do so, we store a lightweight approximation of the feature distribution on the training data and then adapt the model to the test data by restoring or realigning the feature distribution. On both synthetic and real-world measurement shifts, we show improved accuracy, calibration, and data efficiency.

In the second work, the setting is *domain generalisation* (DG, [Blanchard et al. 2011](#); [Muandet et al. 2013](#)). Here, a model is trained on data from multiple related environments or *domains* (e.g., hospitals) with the goal of performing well on data from related but unseen domains. In general, preparation involves exploiting invariances across the training domains in the hope that these invariances also hold in test domains. In particular, prior works have sought to do so by learning models that perform well *on-average* ([Blanchard et al., 2021](#); [Zhang et al., 2021](#)) or *in-the-worst-case* ([Arjovsky et al., 2019](#); [Sagawa\\* et al., 2020](#)). While the former approach tends to lack robustness ([Nagarajan et al., 2021](#)), the latter tends to be overly conservative ([Tsipras et al., 2019](#)). We address these issues by proposing a new probabilistic framework wherein the goal is to learn models that *perform well with high probability*. In particular, by explicitly relating the training and test domains as draws from the same underlying meta-distribution, we ensure that distribution shifts seen during training inform us of *probable* shifts at test time. Then, by minimising a particular quantile of a model’s performance distribution over training domains, we learn models that perform well on

unseen test domains with the corresponding probability.

In the third work, the setting is again *domain generalisation*, but this time we focus on ways to safely harness “spurious” (Geirhos et al., 2020) features. Prior works sought robustness by discarding the spurious or *unstable* features whose relationship with the label changes across domains, restricting the model to features with an invariant or *stable* relationship with the label across domains (Arjovsky et al., 2019; Krueger et al., 2021; Peters et al., 2016). However, unstable features often carry *complementary* information about the label that could boost performance if used correctly in the test domain. We show that it is possible to do so *without test-domain labels*, using only predictions based on the stable features, so long as the stable and unstable features are conditionally independent given the label. We then use this theoretical insight to propose an algorithm for safely harnessing complementary spurious features without test-domain labels. On real and synthetic datasets, we show that this boosts performance without sacrificing robustness.

In the fourth and final work, the setting is representation learning, in particular, *disentangled representation learning*. One of the primary goals of representation learning is to learn representations of complex data that make it easier for downstream tasks to extract useful information (Bengio et al., 2013). In the context of this thesis, this can be viewed as an extreme setting for distribution shift in which the *task changes or shifts* at test time. With this view in mind, disentangled representation learning can be seen as preparing for an unknown test-time task by recovering and separating the data’s underlying factors of variation, discarding as little information as possible (Desjardins et al., 2012; Kulkarni et al., 2015). To better facilitate the learning and comparison of methods for disentangled representation learning, prior works have proposed protocols or frameworks for evaluating disentangled representations. We build on one such framework, that of Eastwood and Williams (2018), by first connecting it to identifiability and then extending it to contain new complementary measures of representation quality which better correlate with downstream performance.

## 1.1 Outline

The remainder of this thesis is structured as follows:

- **Chapter 2** provides background material for the following chapters.
- **Chapter 3** focuses on domain adaptation, in particular, *source-free domain ad-*

*aptation*, and is based on the following paper:

Eastwood, C., Mason, I., Williams, C. K. I., and Schölkopf, B. (2022a). Source-free adaptation to measurement shift via bottom-up feature restoration. In *The Tenth International Conference on Learning Representations*

- **Chapter 4** focuses on *domain generalisation* and the introduction of a new probabilistic framework. It is based on the following paper:

Eastwood, C., Robey, A., Singh, S., Kügelgen, J. V., Hassani, H., Pappas, G. J., and Schölkopf, B. (2022c). Probable domain generalization via quantile risk minimization. In *Advances in Neural Information Processing Systems*, volume 35, pages 17340–17358

- **Chapter 5** focuses *domain generalisation* and how “spurious” features can be safely harnessed. It is based on the following paper:

Eastwood, C., Singh, S., Nicolicioiu, L. A., Von Kügelgen, J., and Schölkopf, B. (2023b). Spuriousity didn’t kill the classifier: Using invariant predictions to harness spurious features. In *Advances in Neural Information Processing Systems*

- **Chapter 6** focuses on representation learning, in particular, *disentangled representation learning*, and is based on the following paper:

Eastwood, C., Nicolicioiu, A. L., Kügelgen, J. V., Kekić, A., Träuble, F., Dittadi, A., and Schölkopf, B. (2023a). DCI-ES: An extended disentanglement framework with connections to identifiability. In *The Eleventh International Conference on Learning Representations*

- **Chapter 7** presents conclusions and avenues for future research.

## 2

# Background

This chapter provides a general background for the remainder of the thesis, with more detailed background information, including notation setup and formal definitions, deferred to the papers themselves. In particular, this chapter provides a general background for each of the distribution-shift settings considered in this work: domain adaptation (§ 2.1), domain generalisation (§ 2.2) and disentangled representation learning (§ 2.3). This chapter also briefly discusses the related setting of meta-learning (§ 2.4), and ends by comparing each of the settings considered in this work to make clear their similarities and differences (§ 2.5).

## 2.1 Domain Adaptation

When there is a shift in the data distribution between training and test domains, one strategy might be to re-collect and annotate enough examples in the test domain to re-train the model. However, this process can be extremely expensive. A cheaper strategy is that of *few-shot domain adaptation*, where models are trained such that they can be adapted in the test domain given only a few labelled examples. An even cheaper strategy is that of *unsupervised domain adaptation* (UDA), where unlabelled test-domain data is incorporated into the training process in order to minimise the domain ‘gap’, e.g., by aligning statistics of the training and test distributions (Ganin and Lempitsky, 2015; Long et al., 2015). However, UDA methods require *simultaneous* access to the training and test datasets—an often impractical requirement due to privacy regulations or transmission constraints, e.g., deploying healthcare models (trained on private data) to different hospitals or deploying image-processing models (trained on huge datasets) to new mobile devices. This leads to the setting of *source-free domain adaptation* (SFDA),

where models are adapted to previously unseen test data without labels and without access to the original training or ‘source’ dataset. We now discuss few-shot, unsupervised and source-free domain adaptation in more detail.

### 2.1.1 Few-shot domain adaptation

Few-shot learning aims to learn a model that can be adapted given only a few labelled examples. The most common formulation involves a *task shift* at test time, where new classes are encountered that were not seen at training time. These task shifts are usually described using the  $N$ -shot  $K$ -way terminology, where  $N$  is the number of labelled examples available and  $K$  is the number of new classes encountered. While this formulation is the most common, few-shot learning can be applied more generally to any type of distribution shift, including domain adaptation (Motiian et al., 2017) and domain generalisation (Li et al., 2018a).

The simplest approach to few-shot learning is to only adapt a small subset of the model’s parameters depending on the expected shift type, e.g., the last layer for label shift or encountering new classes (Yosinski et al., 2014) and the first layer for low-level corruptions (Eastwood et al., 2021). Another approach is to explicitly optimise for few-shot performance using meta-learning, as discussed in § 2.4.

### 2.1.2 Unsupervised domain adaptation

In unsupervised domain adaptation (UDA), unlabelled test-domain data is incorporated into the training process in order to minimise the domain ‘gap’. Inspired by the theory of Ben-David et al. (2010, 2007), the most common approach is to align the training and test domains by matching their distributions in feature space (Ganin and Lempitsky, 2015; Ganin et al., 2016; Long et al., 2015, 2018; Shu et al., 2018; Tzeng et al., 2017).

### 2.1.3 Source-free domain adaptation

In source-free domain adaptation (SFDA), models are adapted to previously unseen test data without labels and without access to the original training or ‘source’ dataset (see Table 2.1). Thus, it can be seen as a further restriction of UDA where the training and test domains are never available *simultaneously* due to privacy, transmission or storage constraints. The most common approach to SFDA is entropy minimisation, i.e., adapting the model in the test domain by making its predictions more confident (Kundu

Table 2.1: **Distribution-shift settings at test/adaptation time.** The settings considered in this work can be distinguished based on the data that is available during adaptation. In particular, whether or not the training or source data is available, and the type of test or target data that is available (labelled or unlabelled). Based on Table 7 of Chapter 3 (given in App. A.1).

Setting	Source data	Target data	Adapt. loss
Fine-tuning & few-shot DA	-	$x^t, y^t$	$L(x^t, y^t)$
UDA	$x^s, y^s$	$x^t$	$L(x^s, y^s) + L(x^s, x^t)$
Source-free DA	-	$x^t$	$L(x^t)$
Domain generalisation	-	-	-

et al., 2020; Li et al., 2020c; Liang et al., 2020; Morerio et al., 2020). In particular, Liang et al. (2020) recently achieved compelling results by re-purposing the semi-supervised information-maximisation loss (Krause et al., 2010) and combining it with a pseudo-labelling loss (Lee et al., 2013). Another approach is that of adaptive batch normalisation (AdaBN, Li et al. 2017), where the training-data batch-normalisation statistics are replaced with those of the test data. Surprisingly, this simple and parameter-free approach is often competitive with more complex techniques, encouraging more recent works to combine AdaBN with entropy minimisation (Wang et al., 2021). Finally, another approach is to train generative models of the training-domain data-distribution so that samples can be drawn and leveraged in the test domain (Kundu et al., 2020; Kurmi et al., 2021; Li et al., 2020c; Morerio et al., 2020; Stan and Rostami, 2021; Yeh et al., 2021).

## 2.2 Domain Generalisation

In domain generalisation (DG), a model is trained on data from multiple related domains with the goal of performing well on data from other related but unseen test domains. For example, in the iWildCam dataset (Beery et al., 2021), the task is to classify different animal species in images, and the domains correspond to the different camera-traps which captured the images (see Fig. 2.1). In general, preparation involves exploiting invariances across the training domains in the hope that these invariances also hold in related but distinct test domains. To do so, the most common approaches involve

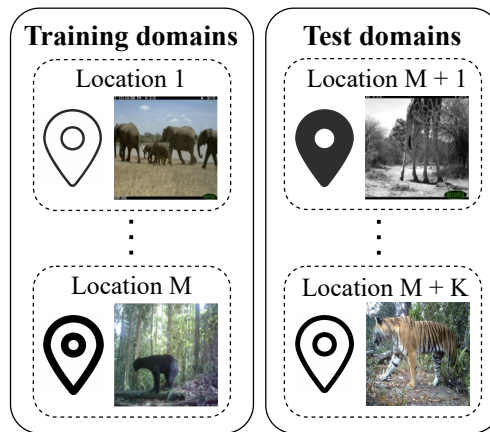


Figure 2.1: **Overview of the domain generalisation problem.** Training and test data are drawn from multiple related distributions or domains. For example, in the iWildCam dataset (Beery et al., 2021), which contains camera-trap images of animal species, the domains correspond to the different camera-traps which captured the images. Based on Fig. 1a of Chapter 4.

learning models that perform well *on-average* (Blanchard et al., 2021; Zhang et al., 2021) or *in-the-worst-case* (Arjovsky et al., 2019; Sagawa\* et al., 2020).

In particular, one line of work (see, e.g., Ahuja et al. 2021; Arjovsky et al. 2019; Krueger et al. 2021) formulates the DG problem through the lens of *robust optimisation* (Ben-Tal et al., 2009), with various approaches solving constrained (Robey et al., 2021) and distributionally robust (Sagawa\* et al., 2020) objectives to maximise worst-case performance.

Another line of work focuses on the links between *invariant prediction and causality*. Here, one goal is to identify components which are stable, robust, or *invariant*, and find means to transfer them across problems (Bareinboim and Pearl, 2014; Gong et al., 2016; Huang et al., 2017; Zhang et al., 2015, 2013), and another is to leverage different forms of invariance across domains in order to discover causal relationships which, under the invariant mechanism assumption (Peters et al., 2017), generalise to new domains (Arjovsky et al., 2019; Gamella and Heinze-Deml, 2020; Heinze-Deml et al., 2018; Krueger et al., 2021; Peters et al., 2016; Pfister et al., 2019; Rojas-Carulla et al., 2018).

Outside of these two lines of work, many methods have been proposed for DG which draw on insights from a diverse array of fields, including approaches based on tools from meta-learning (Balaji et al., 2018; Dou et al., 2019; Li et al., 2018a; Shu



et al., 2021; Zhang et al., 2021), kernel methods (Deshmukh et al., 2019; Dubey et al., 2021), and information theory (Ahuja et al., 2021). Also prominent are works that design regularisers to generalise OOD (Kim et al., 2021; Li et al., 2020a; Zhao et al., 2020) and works that seek domain-invariant representations (Ganin et al., 2016; Huang et al., 2020; Li et al., 2018b).

## 2.3 Disentangled Representation Learning

A primary goal of representation learning is to learn representations  $r(\mathbf{x})$  of complex data  $\mathbf{x}$  that “make it easier to extract useful information when building classifiers or other predictors” (Bengio et al., 2013). *Disentangled* representations, which aim to recover and separate (or, more formally, *identify*) the underlying factors of variation  $\mathbf{z}$  that generate the data as  $\mathbf{x} = g(\mathbf{z})$ , are a promising step in this direction. In particular, it has been argued that such representations are not only interpretable (Chen et al., 2016; Kulkarni et al., 2015) but also make it easier for downstream tasks to extract useful information (Bengio et al., 2013; Desjardins et al., 2012; Lake et al., 2017; Schmidhuber, 1992).

While there is no single, widely-accepted definition, many evaluation protocols have been proposed to capture different notions of disentanglement based on the relationship between the learnt representation or *code*  $\mathbf{c} = r(\mathbf{x})$  and the ground-truth data-generative factors  $\mathbf{z}$  (see Fig. 2.2) (Chen et al., 2018; Eastwood and Williams, 2018; Higgins et al., 2017; Kim and Mnih, 2018; Ridgeway and Mozer, 2018; Shu et al., 2020; Suter et al., 2019). In particular, the metrics of Eastwood and Williams (2018)—*disentanglement* (D), *completeness* (C) and *informativeness* (I)—estimate this relationship by learning a *probe*  $f$  to predict  $\mathbf{z}$  from  $\mathbf{c}$  and can be used to relate many other notions of disentanglement (see Locatello et al. 2020a, § 6).

Approaches for learning disentangled representations can be grouped based on their level of supervision. *Unsupervised approaches* are mostly based on the variational autoencoder (VAE, Kingma and Welling 2013) and tend to encourage disentanglement through an unrealistic assumption of statistically independent factors  $\mathbf{z}$  (Chen et al., 2018; Higgins et al., 2017; Kim and Mnih, 2018). Perhaps more worryingly, it has been shown that the unsupervised learning of disentangled representations is theoretically impossible from i.i.d. observations without assumptions on both the data and model (Hyvärinen and Pajunen, 1999; Locatello et al., 2019)—assumptions which are often difficult to justify and impossible to test. Fortunately, many real-world obser-



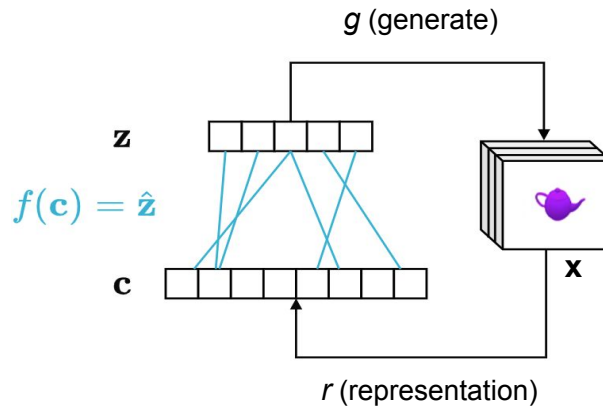


Figure 2.2: **Evaluating disentangled representations.** First, the data is generated as  $\mathbf{x} = g(\mathbf{z})$ . Next, a model for disentangled representation learning is training on  $\mathbf{x}$ , ultimately producing a representation or code  $\mathbf{c} = r(\mathbf{x})$ . Finally, the relationship between the learned representation  $\mathbf{c}$  and ground-truth data-generating factors  $\mathbf{z}$  (cyan links) is used to evaluate the quality of the learned representation.

variations are not i.i.d. as they arise from changes in only a few underlying factors of variation—providing a weak supervision signal for disentangled representation learning (Bengio et al., 2013, 2020; Dayan, 1993; Schölkopf et al., 2021).

Many *weakly-supervised approaches* assume access to paired or grouped observations across which only a single, known factor changes (Bouchacourt et al., 2018; Hosoya, 2019; Kulkarni et al., 2015; Li et al., 2020b; Reed et al., 2015; Shu et al., 2020). However, this kind of exact knowledge about which underlying factor has changed typically requires explicit human annotation or strong control over the data acquisition process. Locatello et al. (2020b) weaken this requirement by learning disentangled representations from pairs of observations without knowledge of which or how many factors have changed, so long as they do not *all* change. For example, given two temporally-close video frames of a scene, we may expect some object properties and positions to change, but not all. Another line of work assumes access to proxy counterfactual interventions which can be used to approximately align factor values (Eastwood et al., 2022b), e.g., using learned semantic-alignment networks to change the position and orientation of objects in images (Rocco et al., 2018).

In the context of this thesis, disentangled representation learning can be viewed as an extreme setting for distribution shift in which the *task* changes or shifts at test time. The goal is thus to prepare for an unknown test-time task by recovering and separating the data’s underlying factors of variation, discarding as little information as

possible (Bengio et al., 2013; Desjardins et al., 2012).

## 2.4 Meta-Learning

In meta-learning, or learning to learn (Schmidhuber, 1987; Thrun and Pratt, 1998), multiple *inner* learning episodes are used to learn an *outer/meta*-learning algorithm itself. In particular, an inner learning algorithm updates model parameters  $\theta$  (e.g., neural network weights) to solve a task like image classification, while an outer/meta-algorithm updates the inner learning algorithm’s (hyper)parameters  $\phi$  (e.g., learning rate) to improve an outer objective like generalisation performance or speed of learning. While meta-learning has been applied to a whole host of settings (see, e.g., Hospedales et al. 2021 for an overview), we focus on those most relevant to this thesis, namely *domain adaptation* and *domain generalisation*. We now give a brief overview of these meta-learning settings, deferring detailed method comparisons to the papers themselves.

For domain adaptation, meta-learning can be used to define a meta-objective that directly optimises the performance of an inner algorithm on a held-out domain (Li and Hospedales, 2020). This inner algorithm may leverage unlabelled examples (unsupervised domain adaptation, as in § 2.1.2) or a small number of labelled examples (few-shot domain adaptation, as in § 2.1.1).

For domain generalisation, meta-learning can be used to learn regularisers (Balaji et al., 2018), losses (Li et al., 2019) and data augmentations/transformations (Tseng et al., 2020) that maximise the robustness of an inner algorithm to held-out domain shifts (Li et al., 2018a).

## 2.5 Summary

To summarise the distribution-shift settings considered in this work, we now make clear their connections using Fig. 2.3 which, through a number of questions, provides a visual overview of distribution shift as it relates to this thesis. Note that the lines separating these settings are often blurred. In particular, despite adapting in the test domain, our approach in Chapter 5 is best deemed a domain generalisation approach as it more closely aligns with that literature.

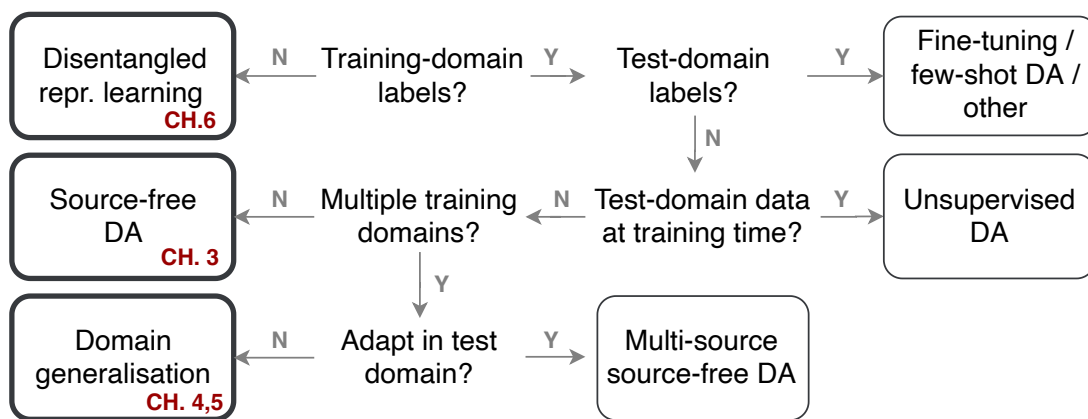


Figure 2.3: **Overview of distribution shift as it relates to this thesis.** The settings considered in this thesis can *generally* be distinguished from each other, and related topics in the literature, based on the above questions.

---

# 3

## Source-Free Domain Adaptation

This chapter focuses on domain adaptation, in particular, *source-free domain adaptation* (SFDA, [Li et al. 2020c](#)). Here, a model is adapted to unlabelled and previously-unseen test data *without access to the original training data*, e.g., due to privacy regulations or storage constraints. For example, this situation arises when deploying health-care models trained on private datasets to new hospitals, or deploying image/language models trained on enormous datasets to mobile devices. Prior works prepared for SFDA by generating artificial negative datasets ([Kundu et al., 2020](#)) or introducing special training techniques that make the model easier to adapt ([Liang et al., 2020](#)), and then adapted via entropy-minimisation, i.e., making predictions more confident on the test data ([Liang et al., 2020](#)). While this approach can be effective, it relies on good initial predictions, destroys model calibration, and only applies to classification. We address these issues for one particular type of distribution shift, termed measurement shift, which can be resolved by restoring the same features on the test data rather than learning new ones. In particular, we prepare for source-free adaptation to measurement shift by storing a lightweight approximation of the feature distribution on the training data, and then adapt the model by restoring or realigning its feature distribution on the test data. On both synthetic and real-world measurement shifts, we show improved accuracy, calibration, and data efficiency.

### 3.1 Contribution

I led this project from conceptualisation to final form. In particular, I was heavily involved in coming up with the main idea, formalising the concept of measurement shift, designing the experimental analyses, running the experimental analyses, and writing

the manuscript. Some of these tasks, including running the experiments and writing, were shared with Ian Mason, with whom I share first authorship.

## **3.2 Paper**

# SOURCE-FREE ADAPTATION TO MEASUREMENT SHIFT VIA BOTTOM-UP FEATURE RESTORATION

Cian Eastwood<sup>\*†§</sup> Ian Mason<sup>\*†</sup> Christopher K. I. Williams<sup>†‡</sup> Bernhard Schölkopf<sup>§</sup>

<sup>†</sup> School of Informatics, University of Edinburgh

<sup>‡</sup> Alan Turing Institute, London

<sup>§</sup> MPI for Intelligent Systems, Tübingen

## ABSTRACT

Source-free domain adaptation (SFDA) aims to adapt a model trained on labelled data in a source domain to unlabelled data in a target domain *without access to the source-domain data during adaptation*. Existing methods for SFDA leverage entropy-minimization techniques which: (i) apply only to classification; (ii) destroy model calibration; and (iii) rely on the source model achieving a good level of feature-space class-separation in the target domain. We address these issues for a particularly pervasive type of domain shift called *measurement shift* which can be resolved by *restoring* the source features rather than extracting new ones. In particular, we propose *Feature Restoration* (FR) wherein we: (i) store a lightweight and flexible approximation of the feature distribution under the source data; and (ii) adapt the feature-extractor such that the approximate feature distribution under the target data realigns with that saved on the source. We additionally propose a bottom-up training scheme which boosts performance, which we call *Bottom-Up Feature Restoration* (BUFR). On real and synthetic data, we demonstrate that BUFR outperforms existing SFDA methods in terms of accuracy, calibration, and data efficiency, while being less reliant on the performance of the source model in the target domain.

## 1 INTRODUCTION

In the real world, the conditions under which a system is developed often differ from those in which it is deployed—a concept known as *dataset shift* (Quiñero-Candela et al., 2009). In contrast, conventional machine learning methods work by ignoring such differences, assuming that the development and deployment domains match or that it makes no difference if they do not match (Storkey, 2009). As a result, machine learning systems often fail in spectacular ways upon deployment in the test or *target* domain (Torralba & Efros, 2011; Hendrycks & Dietterich, 2019)

One strategy might be to re-collect and annotate enough examples in the target domain to re-train or fine-tune the model (Yosinski et al., 2014). However, manual annotation can be extremely expensive. Another strategy is that of *unsupervised domain adaptation* (UDA), where unlabelled data in the target domain is incorporated into the development process. A common approach is to minimize the domain ‘gap’ by aligning statistics of the source and target distributions in feature space (Long et al., 2015; 2018; Ganin & Lempitsky, 2015). However, these methods require simultaneous access to the source and target datasets—an often impractical requirement due to privacy regulations or transmission constraints, e.g. in deploying healthcare models (trained on private data) to hospitals with different scanners, or deploying image-processing models (trained on huge datasets) to mobile devices with different cameras. Thus, UDA *without access to the source data at deployment time* has high practical value.

Recently, there has been increasing interest in methods to address this setting of *source-free domain adaptation* (SFDA, Kundu et al. 2020; Liang et al. 2020; Li et al. 2020; Morerio et al. 2020) where the source dataset is unavailable during adaptation in the deployment phase. However, to adapt to the target domain, most of these methods employ entropy-minimization techniques which: (i) apply only to classification (discrete labels); (ii) destroy model calibration—minimizing prediction-entropy causes every sample to be classified (correctly or incorrectly) with extreme confidence; and (iii) assume that, in the target domain, the feature space of the unadapted source model contains reasonably well-separated data clusters, where samples within a cluster tend to share the same class label. As

<sup>\*</sup>Equal contribution. Correspondence to

or [ianxmason@gmail.com](mailto:ianxmason@gmail.com).

demonstrated in Section 5, even the most innocuous of shifts can destroy this *initial feature-space class-separation* in the target domain, and with it, the performance of these techniques.

We address these issues for a specific type of domain shift which we call *measurement shift* (MS). Measurement shift is characterized by a change in measurement system and is particularly pervasive in real-world deployed machine learning systems. For example, medical imaging systems often fail when deployed to hospitals with different scanners (Zech et al., 2018; AlBadawy et al., 2018; Beede et al., 2020) or different staining techniques (Tellez et al., 2019), while self-driving cars often struggle under “shifted” deployment conditions like natural variations in lighting (Dai & Van Gool, 2018) or weather conditions (Volk et al., 2019). Importantly, in contrast to many other types of domain shift, measurement shifts can be resolved by simply *restoring* the source features in the target domain—we do not need to learn *new* features in the target domain to discriminate well between the classes. Building on this observation, we propose Feature Restoration (FR)—a method which seeks to extract features with the same semantics from the target domain as were previously extracted from the source domain, under the assumption that this is sufficient to restore model performance. At development time, we train a source model and then use softly-binned histograms to save a lightweight and flexible approximation of the feature distribution under the source data. At deployment time, we adapt the source model’s feature-extractor such that the approximate feature distribution under the target data aligns with that saved on the source. We additionally propose Bottom-Up Feature Restoration (BUFR)—a bottom-up training scheme for FR which significantly improves the degree to which features are restored by preserving learnt structure in the later layers of a network. While the assumption of measurement shift does reduce the generality of our methods—they do not apply to all domain shifts, but rather a subset thereof—our experiments demonstrate that, in exchange, we get improved performance on this important real-world problem. To summarize our main contributions, we:

- Identify a subset of domain shifts, which we call *measurement shifts*, for which restoring the source features in the target domain is sufficient to restore performance (Sec. 2);
- Introduce a *lightweight* and *flexible* distribution-alignment method for the source-free setting in which softly-binned histograms approximate the marginal feature distributions (Sec. 3);
- Create & release EMNIST-DA, a simple but challenging dataset for studying MS (Sec. 5.1);
- Demonstrate that BUFR generally outperforms existing SFDA methods in terms of accuracy, calibration, and data efficiency, while making less assumptions about the performance of the source model in the target domain (i.e. the initial feature-space class-separation) (Sec. 5.2–5.5);
- Highlight & analyse issues with entropy-minimization in existing SFDA methods (Sec. 5.5).

## 2 SETTING: SOURCE-FREE ADAPTATION TO MEASUREMENT SHIFT

We now describe the two phases of source-free domain adaptation (SFDA), development and deployment, before exploring measurement shift. For concreteness, we work with discrete outputs (i.e. classification) but FR can easily be applied to continuous outputs (i.e. regression).

**Source-free adaptation.** At **development time**, a source model is trained *with the expectation that an unknown domain shift will occur upon deployment in the target domain*. Thus, the primary objective is to equip the model for source-free adaptation at deployment time. For previous work, this meant storing per-class means in feature space (Chidlovskii et al., 2016), generating artificial negative datasets (Kundu et al., 2020), or introducing special training techniques (Liang et al., 2020). For us, this means storing lightweight approximate parameterizations of the marginal feature distributions, as detailed in the next section. More formally, a source model  $f_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$  is trained on  $n_s$  labelled examples from the source domain  $\mathcal{D}_s = \{(\mathbf{x}_s^{(i)}, y_s^{(i)})\}_{i=1}^{n_s}$ , with  $\mathbf{x}_s^{(i)} \in \mathcal{X}_s$  and  $y_s^{(i)} \in \mathcal{Y}_s$ , before saving any lightweight statistics of the source data  $\mathcal{S}_s$ . At **deployment time**, we are given a pre-trained source model  $f_s$ , lightweight statistics of the source data  $\mathcal{S}_s$ , and  $n_t$  unlabelled examples from the target domain  $\mathcal{D}_t = \{\mathbf{x}_t^{(i)}\}_{i=1}^{n_t}$ , with  $\mathbf{x}_t^{(i)} \in \mathcal{X}_t$ . The goal is to learn a target model  $f_t : \mathcal{X}_t \rightarrow \mathcal{Y}_t$  which accurately predicts the unseen target labels  $\{y_t^{(i)}\}_{i=1}^{n_t}$ , with  $y_t^{(i)} \in \mathcal{Y}_t$ . Importantly, the source dataset  $\mathcal{D}_s$  is not accessible during adaptation in the deployment phase.

**Domain shift.** As depicted in Figure 1a, *domain shift* (Storkey, 2009, Section 9) can be understood by supposing some underlying, domain-invariant latent representation  $L$  of a sample  $(X, Y)$ . This combines with the domain (or environment) variable  $E$  to produce the observed covariates  $X = m_E(L)$ , where  $m_E$  is some domain-dependent mapping. For example,  $L$  could describe the shape,

Published as a conference paper at ICLR 2022

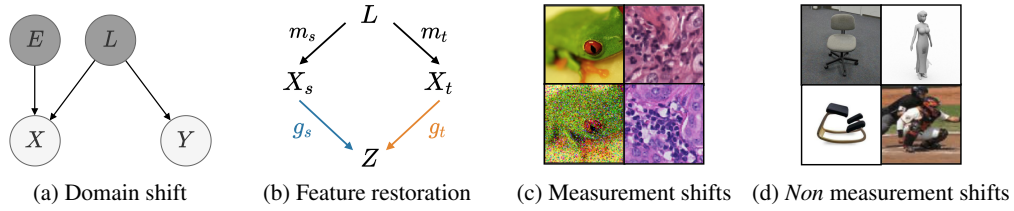


Figure 1: Domain shift, feature restoration and measurement shift. (c,d): Top=source, bottom=target. (c): CIFAR-10-C ‘frog’ & CAMELYON17 ‘tumor’. (d): Office-31 ‘desk chair’ & VisDA-C ‘person’.

appearance and pose parameters of scene objects, with  $X$  obtained by “rendering” the scene  $L$ , taking into account parameters in  $E$  that prescribe e.g. lighting, camera properties, background etc.

**Feature restoration.** In the source domain we learn a feature space  $Z = g_s(X_s) = g_s(m_s(L))$ , where our source model  $f_s$  decomposes into a feature-extractor  $g_s$  and a classifier  $h$ , with  $f_s = h \circ g_s$  (left path of Figure 1b). For our source model  $f_s$  to achieve good predictive accuracy, the features  $Z$  *must* capture the information in  $L$  about  $Y$  and ignore the variables in  $E = s$  that act as “nuisance variables” for obtaining this information from  $X_s$  (e.g. lighting or camera properties). In the target domain ( $E = t$ ), we often cannot extract the same features  $Z$  due to a change in nuisance variables. This hurts predictive accuracy as it reduces the information about  $L$  in  $Z = g_s(X_t)$  (and thus about  $Y$ ). We can *restore* the source features in the target domain by learning a target feature-extractor  $g_t$  such that the target feature distribution aligns with that of the source (right path of Figure 1b), i.e.  $p(g_t(X_t)) \approx p(g_s(X_s))$ . Ultimately, we desire that for any  $L$  we will have  $g_s(m_s(L)) = g_t(m_t(L))$ , i.e. that for source  $X_s = m_s(L)$  and target  $X_t = m_t(L)$  images generated from the same  $L$ , their corresponding  $Z$ ’s will match. We can use synthetic data, where we have source and target images generated from the same  $L$ , to quantify the degree to which the source features are *restored* in the target domain with  $|g_s(m_s(L)) - g_t(m_t(L))|$ . In Section 5.5, we use this to compare quantitatively the degree of restoration achieved by different methods.

**Measurement shifts.** For many real-world domain shifts, restoring the source features in the target domain is sufficient to restore performance—we do not need to learn new features in order to discriminate well between the classes in the target domain. We call these *measurement shifts* as they generally arise from a change in measurement system (see Figure 1c). For such shifts, it is preferable to restore the same features rather than learn new ones via e.g. entropy minimization as the latter usually comes at the cost of model calibration—as we demonstrate in Section 5.

**Common UDA benchmarks are *not* measurement shifts.** For many other real-world domain shifts, restoring the source features in the target domain is *not* sufficient to restore performance—we need *new* features to discriminate well between the classes in the target domain. This can be caused by *concept shift* (Moreno-Torres et al., 2012, Sec. 4.3), where the features that define a concept change across source and target domains, or by the source model exploiting spurious correlations or “shortcuts” (Arjovsky et al., 2019; Geirhos et al., 2020) in the source domain which are not discriminative—or do not even exist—in the target domain. Common UDA benchmark datasets like Office-31 (Saenko et al., 2010) and VisDA-C (Peng et al., 2018) fall into this category of domain shifts. In particular, Office-31 is an example concept shift—‘desk chair’ has very different meanings (and thus features) in the source and target domains (left column of Fig. 1d)—while VisDA-C is an example of source models tending to exploit shortcuts. More specifically, in the synthetic-to-real task of VisDA-C (right column of Fig. 1d), source models tend not to learn general geometric aspects of the synthetic classes. Instead, they exploit peculiarities of the e.g. person-class which contains only 2 synthetic “people” rendered from different viewpoints with different lighting. Similarly, if we consider the real-to-synthetic task, models tend to exploit textural cues in the real domain that do not exist in the synthetic domain (Geirhos et al., 2019). As a result, the standard approach is to first pretrain on ImageNet to gain more “general” visual features and then carefully<sup>1</sup> fine-tune these features on (i) the source domain and then (ii) the target domain, effectively making the adaptation task ImageNet  $\rightarrow$  synthetic  $\rightarrow$  real. In Appendix D we illustrate that existing methods actually fail without this ImageNet pretraining as successful discrimination in the target domain *requires* learning new combinations of the general base ImageNet features. In summary, common UDA benchmarks like Office and VisDA-C *do not contain measurement shift* and thus are not suitable for evaluating our methods. We nonetheless report and analyse results on VisDA-C in Appendix D.

<sup>1</sup>Many works lower the learning rate of early layers in source and target domains, e.g. Liang et al. (2020).



Published as a conference paper at ICLR 2022

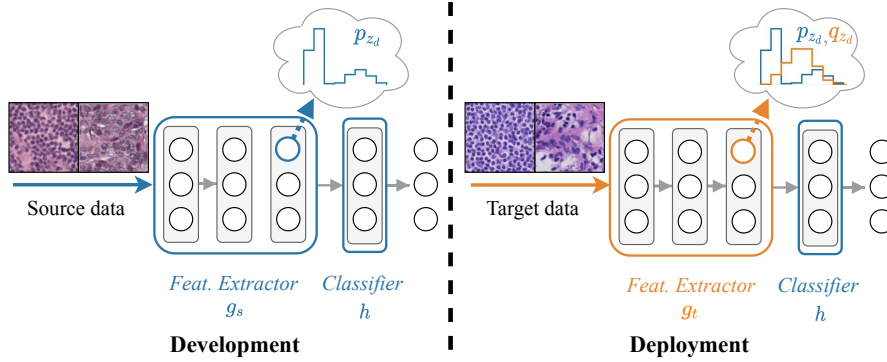


Figure 2: The Feature Restoration framework. *Left*: At development time, a source model is trained before saving approximations of the  $D$  marginal feature distributions under the source data  $\{p_{z_d}\}_{d=1}^D$ . *Right*: At deployment time, the feature-extractor is adapted such that the approximations of the marginal feature distributions on the target data  $\{q_{z_d}\}_{d=1}^D$  align with those saved on the source.

### 3 FEATURE RESTORATION

Below we detail the *Feature Restoration* (FR) framework. During development we train a model and then save a lightweight approximation of the feature distribution under the source data. At deployment time, we adapt the model’s feature-extractor such that the approximate feature distribution under the target data aligns with that saved on the source. Figure 2 gives an overview of the FR framework.

#### 3.1 DEVELOPMENT

**Setup.** The source model  $f_s$  is first trained using some loss, e.g. cross-entropy. Unlike most existing SFDA methods (Chidlovskii et al., 2016; Liang et al., 2020; Kundu et al., 2020), we make no modification to the standard training process, allowing pretrained source models to be utilized. We decompose the source model  $f_s$  into a feature-extractor  $g_s : \mathcal{X}_s \rightarrow \mathbb{R}^D$  and a classifier  $h : \mathbb{R}^D \rightarrow \mathcal{Y}_s$ , where  $D$  is the dimensionality of the feature space. So  $\mathbf{z}_s^{(i)} = g_s(\mathbf{x}_s^{(i)})$  denotes the features extracted for source sample  $i$ , and  $\hat{y}_s^{(i)} = f_s(\mathbf{x}_s^{(i)}) = h(g_s(\mathbf{x}_s^{(i)}))$  denotes the model’s output for source sample  $i$ . Under the assumption of measurement shift, the feature extractor should be adapted to unlabelled target data to give  $\mathbf{z}_t^{(i)} = g_t(\mathbf{x}_t^{(i)})$ , but the classifier  $h$  should remain unchanged, so that  $\hat{y}_t^{(i)} = f_t(\mathbf{x}_t^{(i)}) = h(g_t(\mathbf{x}_t^{(i)}))$ .

**Choosing an approximation of the feature distribution.** For high-dimensional feature spaces, storing the full joint distribution can be prohibitively expensive<sup>2</sup>. Thus, we choose to store only the marginal feature distributions. To accurately capture these marginal distributions, we opt to use soft binning (Dougherty et al., 1995) for its (i) *flexibility*—bins/histograms make few assumptions about distributional form, allowing us to accurately capture marginal feature distributions which we observe empirically to be heavily-skewed and bi-modal (see Appendix I); (ii) *scalability*—storage size does not scale with dataset size (Appendix A, Table 5), permitting very large source datasets (for a fixed number of bins  $B$  and features  $D$ , soft binning requires constant  $O(BD)$  storage and simple matrix-multiplication to compute soft counts); and (iii) *differentiability*—the use of soft (rather than “hard”) binning, detailed in the next section, makes our approximation differentiable.

**Estimating the parameters of our approximation on the source data.** We now use the soft binning function of Yang et al. (2018, Sec. 3.1) to approximately parameterize the  $D$  marginal feature distributions on the source data  $\{p_{z_d}\}_{d=1}^D$ , where  $p_{z_d}$  denotes the marginal distribution of the  $d$ -th feature  $z_d$ . Specifically, we approximately parameterize  $p_{z_d}$  using  $B$  normalized bin counts  $\pi_{z_d}^s = [\pi_{z_d,1}^s, \dots, \pi_{z_d,B}^s]$ , where  $\pi_{z_d,b}^s$  represents the probability that a sample  $z_d^{(i)}$  falls into bin  $b$  under the source data and  $\sum_{b=1}^B \pi_{z_d,b}^s = 1$ .  $\pi_{z_d}^s$  is calculated using

$$\pi_{z_d}^s = \sum_{i=1}^{n_s} \frac{\mathbf{u}(z_d^{(i)})}{n_s} = \sum_{i=1}^{n_s} \frac{\mathbf{u}(g(\mathbf{x}^{(i)})_d; z_d^{\min}, z_d^{\max})}{n_s}, \quad (1)$$

where  $z_d^{(i)} = g(\mathbf{x}^{(i)})_d$  denotes the  $d$ -th dimension of the  $i$ -th sample in feature space,  $\mathbf{u}$  is the vector-

<sup>2</sup>If we assume features are jointly Normal, computational complexity is  $O(ND^2)$  per update, where  $N$  is the batch size. If we bin the feature space into histograms ( $B$  bins per dimension), memory complexity is  $O(B^D)$ .

valued soft binning function (see Appendix A),  $z_d^{min} = \min_{i=1}^{n_s} z_d^{(i)}$ , and  $z_d^{max}$  is defined analogously to  $z_d^{min}$ . Repeating this for all  $D$  features, we get  $\pi_{\mathbf{z}}^s = [\pi_{z_1}^s, \pi_{z_2}^s, \dots, \pi_{z_D}^s]$ . In the left-hand “cloud” of Figure 2, the blue curve depicts one such approximate marginal feature distribution  $\pi_{z_d}^s$ . We find it useful to additionally store approximate parameterizations of the marginal logit distributions on the source data  $\pi_{\mathbf{a}}^s$ , where the logit (i.e. pre-softmax) activations  $\mathbf{a}^{(i)}$  are a linear combination of the feature activations  $\mathbf{z}^{(i)}$ , and  $\pi_{\mathbf{a}}^s$  is defined analogously to  $\pi_{\mathbf{z}}^s$ . Note that we can parameterize a similar distribution for regression. Intuitively, aligning the marginal logit distributions further constrains the ways in which the marginal feature distributions can be aligned. We validate this intuition in the ablation study of Appendix J.2. Finally, we equip the model for source-free adaptation at deployment time by saving the parameters/statistics of the source data  $\mathcal{S}_s = \{\pi_{\mathbf{z}}^s, \pi_{\mathbf{a}}^s, \mathbf{z}^{min}, \mathbf{z}^{max}, \mathbf{a}^{min}, \mathbf{a}^{max}\}$ , where  $\mathbf{z}^{min} = [z_1^{min}, z_2^{min}, \dots, z_D^{min}]$  and  $\mathbf{z}^{max}, \mathbf{a}^{min}$ , and  $\mathbf{a}^{max}$  are defined analogously.

### 3.2 DEPLOYMENT

At deployment time, we adapt the feature-extractor such that the approximate marginal distributions on the target data ( $\pi_{\mathbf{z}}^t, \pi_{\mathbf{a}}^t$ ) align with those saved on the source ( $\pi_{\mathbf{z}}^s, \pi_{\mathbf{a}}^s$ ). More specifically, we learn the target feature-extractor  $g_t$  by minimizing the following loss on the target data,

$$\mathcal{L}_{tgt}(\pi_{\mathbf{z}}^s, \pi_{\mathbf{z}}^t, \pi_{\mathbf{a}}^s, \pi_{\mathbf{a}}^t) = \sum_{d=1}^D D_{SKL}(\pi_{z_d}^s || \pi_{z_d}^t) + \sum_{k=1}^K D_{SKL}(\pi_{a_k}^s || \pi_{a_k}^t), \quad (2)$$

where  $D_{SKL}(p||q) = \frac{1}{2}D_{KL}(p||q) + \frac{1}{2}D_{KL}(q||p)$  is the symmetric KL divergence, and  $D_{KL}(\pi_{z_d}^s || \pi_{z_d}^t)$  is the KL divergence between the *distributions* parameterized by normalized bin counts  $\pi_{z_d}^s$  and  $\pi_{z_d}^t$ , which is calculated using

$$D_{KL}(\pi_{z_d}^s || \pi_{z_d}^t) = \sum_{b=1}^B \pi_{z_d,b}^s \log \frac{\pi_{z_d,b}^s}{\pi_{z_d,b}^t}, \quad (3)$$

with  $\pi_{z_d,b}^s$  representing the probability of a sample from feature  $d$  falling into bin  $b$  under the source data, and  $\pi_{z_d,b}^t$  under the target data. Practically, to update on a batch of target samples, we first approximate  $\pi_{\mathbf{z}}^t$  and  $\pi_{\mathbf{a}}^t$  on that batch using Eq. 1, and then compute the loss. Appendix B details the FR algorithm at development and deployment time, while Appendix L summarizes the notations.

### 3.3 BOTTOM-UP FEATURE RESTORATION

A simple gradient-based adaptation of  $g_t$  would adapt the weights of all layers at the same time. Intuitively, however, we expect that many measurement shifts like brightness or blurring can be resolved by only updating the weights of early layers. If the early layers can learn to extract the same features from the target data as they did from the source (e.g. the same edges from brighter or blurrier images of digits), then the subsequent layers shouldn’t need to update. Building on this intuition, we argue that adapting all layers simultaneously unnecessarily destroys learnt structure in the later layers of a network, and propose a bottom-up training strategy to alleviate the issue. Specifically, we adapt  $g_t$  in a bottom-up manner, training for several epochs on one “block” before “unfreezing” the next. Here, a block can represent a single layer or group of layers (e.g. a residual block, He et al. 2016), and “unfreezing” simply means that we allow the block’s weights to be updated. We call this method *Bottom-Up Feature Restoration* (BUFR). In Section 5 we illustrate that BU training *significantly* improves accuracy, calibration, and data efficiency by preserving learnt structure in later layers of  $g_t$ .

## 4 RELATED WORK

**Fine-tuning.** A well-established paradigm in deep learning is to first pretrain a model on large-scale “source” data (e.g. ImageNet) and then fine-tune the final layer(s) on “target” data of interest (Girshick et al., 2014; Zeiler & Fergus, 2014). This implicitly assumes that new high-level concepts should be learned by recombining old (i.e. fixed) low-level features. In contrast, under the assumption of measurement shift, we fix the final layer and fine-tune the rest. This assumes that the same high-level concepts should be *restored* by learning new low-level features. Royer & Lampert (2020) fine-tune each layer of a network individually and select the one that yields the best performance. For many domain shifts, they find it best to fine-tune an early or intermediate layer rather than the final one. This supports the idea that *which layer(s)* should update depends on *what* should be transferred.

**Unsupervised DA.** Inspired by the theory of Ben-David et al. (2007; 2010), many UDA methods seek to align source and target domains by matching their distributions in feature space (Long et al., 2015; 2018; Ganin & Lempitsky, 2015; Ganin et al., 2016; Tzeng et al., 2017; Shu et al., 2018).

Published as a conference paper at ICLR 2022

However, as most of these methods are nonparametric (i.e. make no assumptions about distributional form), they require the source data during adaptation to align the distributions. In addition, parametric methods like Deep CORAL (Sun & Saenko, 2016) are not designed for the source-free setup—they prevent degenerate solutions during alignment with a classification loss on the source data and have storage requirements that are at least quadratic in the number of features. In contrast, our method works without the source data and its storage is linear in the number of features.

**Source-free DA.** Recently, Liang et al. (2020) achieved compelling results by re-purposing the semi-supervised information-maximization loss (Krause et al., 2010) and combining it with a pseudo-labelling loss (Lee et al., 2013). However, their entropy-minimizing losses are classification-specific, destroy model calibration, and rely on good initial source-model performance in the target domain (as demonstrated in the next section). Other works have trained expensive generative models so that the source data-distribution can be leveraged in the target domain (Li et al., 2020; Morerio et al., 2020; Kundu et al., 2020; Kurmi et al., 2021; Yeh et al., 2021; Stan & Rostami, 2021). However, these methods are still classification-specific and rely on good initial feature-space class-separation for entropy minimization (Li et al., 2020; Kundu et al., 2020), pseudo-labelling (Morerio et al., 2020; Stan & Rostami, 2021), and aligning the predictions of the source and target models (Kurmi et al., 2021; Yeh et al., 2021). Another approach is to focus on the role of batch-normalization (BN). Li et al. (2017) propose Adaptive BN (AdaBN) where the source data BN-statistics are replaced with those of the target data. This simple parameter-free method is often competitive with more complex techniques. Wang et al. (2021) also use the target data BN-statistics but additionally train the BN-parameters on the target data via entropy minimization, while Ishii & Sugiyama (2021) retrain the feature-extractor to align BN-statistics. Our method also attempts to match statistics of the marginal feature distributions, but is not limited to matching only the first two moments—hence can better handle non-Gaussian distributions.

## 5 EXPERIMENTS

In this section we evaluate our methods on multiple datasets (shown in Appendix F), compare to various baselines, and provide insights into *why* our method works through a detailed analysis.

### 5.1 SETUP

**Datasets and implementation.** Early experiments on MNIST-M (Ganin et al., 2016) and MNIST-C (Mu & Gilmer, 2019) could be well-resolved by a number of methods due to the small number of classes and relatively mild corruptions. Thus, to better facilitate model comparison, we additionally create and release EMNIST-DA—a domain adaptation (DA) dataset based on the 47-class Extended MNIST (EMNIST) character-recognition dataset (Cohen et al., 2017). We also evaluate on object recognition with CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019), and on real-world measurement shifts with CAMELYON17 (Bandi et al., 2018). We use a simple 5-layer convolutional neural network (CNN) for digit and character datasets and a ResNet-18 (He et al., 2016) for the rest. Full dataset details are provided in Appendix F and implementation details in Appendix G. Code is available at <https://github.com/cianeastwood/bufr>.

**Baselines and their relation.** We show the performance of the source model on the source data as *No corruption*, and the performance of the source model on the target data (before adapting) as *Source-only*. We also implement the following baselines for comparison: *AdaBN* (Li et al., 2017) replaces the source BN-statistics with the target BN-statistics; *PL* is a basic pseudo-labelling approach (Lee et al., 2013); *SHOT-IM* is the information-maximization loss from Liang et al. (2020) which consists of a prediction-entropy term and a prediction-diversity term; and *target-supervised* is an upper-bound that uses labelled target data (we use a 80-10-10 training-validation-test split, reporting accuracy on the test set). For digit and character datasets we additionally implement *SHOT* (Liang et al., 2020), which uses the SHOT-IM loss along with special pre-training techniques (e.g. label smoothing) and a self-supervised PL loss; and *BNM-IM* (Ishii & Sugiyama, 2021), which combines the SHOT-IM loss from Liang et al. with a BN-matching (BNM) loss that aligns feature mean and variances on the target data with BN-statistics of the source. We additionally explore simple alternative parameterizations to match the source and target feature distributions: *Marg. Gauss.* is the BNM loss from Ishii & Sugiyama which is equivalent to aligning 1D Gaussian marginals; and *Full Gauss.* matches the mean and full covariance matrix. For object datasets we additionally implement *TENT* (Wang et al., 2021), which updates only the BN-parameters to minimize prediction-entropy, and also compare to some UDA methods. For all methods we report the classification accuracy and Expected Calibration Error (ECE, Naeini et al. 2015) which measures the difference in expectation between confidence and accuracy.

Table 1: Digit and character results. Shown are the mean and 1 standard deviation.

Model	EMNIST-DA		EMNIST-DA-SEVERE		EMNIST-DA-MILD	
	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$
No corruption	89.4 $\pm$ 0.1	2.3 $\pm$ 0.1	89.4 $\pm$ 0.1	2.3 $\pm$ 0.1	89.4 $\pm$ 0.1	2.3 $\pm$ 0.1
Source-only	29.5 $\pm$ 0.5	30.8 $\pm$ 1.6	3.8 $\pm$ 0.4	42.6 $\pm$ 3.5	78.5 $\pm$ 0.7	4.8 $\pm$ 0.5
AdaBN (Li et al., 2017)	46.2 $\pm$ 1.1	30.3 $\pm$ 1.1	3.7 $\pm$ 0.7	52.4 $\pm$ 4.9	84.9 $\pm$ 0.2	4.9 $\pm$ 0.3
Marg. Gauss. (Ishii & Sugiyama, 2021)	51.8 $\pm$ 1.1	26.7 $\pm$ 1.1	4.8 $\pm$ 0.5	51.6 $\pm$ 6.4	85.8 $\pm$ 0.3	4.5 $\pm$ 0.3
Full Gauss.	67.9 $\pm$ 0.7	17.4 $\pm$ 0.7	29.8 $\pm$ 9.8	45.8 $\pm$ 8.4	85.7 $\pm$ 0.2	4.9 $\pm$ 0.2
PL (Lee et al., 2013)	50.0 $\pm$ 0.6	49.9 $\pm$ 0.6	2.7 $\pm$ 0.4	97.2 $\pm$ 0.4	83.5 $\pm$ 0.1	16.4 $\pm$ 0.1
BNM-IM (Ishii & Sugiyama, 2021)	63.7 $\pm$ 2.2	35.6 $\pm$ 2.2	8.3 $\pm$ 1.3	90.2 $\pm$ 1.1	86.5 $\pm$ 0.1	13.0 $\pm$ 0.1
SHOT-IM (Liang et al., 2020)	70.3 $\pm$ 3.7	29.6 $\pm$ 3.7	24.0 $\pm$ 7.5	76.0 $\pm$ 7.5	86.3 $\pm$ 0.1	13.7 $\pm$ 0.1
SHOT (Liang et al., 2020)	80.0 $\pm$ 4.4	19.7 $\pm$ 4.4	55.1 $\pm$ 23.5	42.7 $\pm$ 23.0	86.1 $\pm$ 0.1	14.8 $\pm$ 0.1
FR (ours)	74.4 $\pm$ 0.8	12.9 $\pm$ 0.9	15.3 $\pm$ 6.8	58.0 $\pm$ 6.8	86.4 $\pm$ 0.1	4.6 $\pm$ 0.3
BUFR (ours)	<b>86.1 <math>\pm</math> 0.1</b>	<b>4.7 <math>\pm</math> 0.2</b>	<b>84.6 <math>\pm</math> 0.2</b>	<b>5.6 <math>\pm</math> 0.3</b>	<b>87.0 <math>\pm</math> 0.2</b>	<b>4.2 <math>\pm</math> 0.2</b>
Target-supervised	86.8 $\pm$ 0.6	7.3 $\pm$ 0.7	85.7 $\pm$ 0.6	7.0 $\pm$ 0.5	87.3 $\pm$ 0.7	8.4 $\pm$ 1.1

## 5.2 CHARACTER-RECOGNITION RESULTS

Table 1 reports classification accuracies and ECEs for EMNIST-DA, with Appendix K reporting results for MNIST datasets (K.1) and full, per-shift results (K.4 and K.5). The severe and mild columns represent the most and least “severe” shifts respectively, where a shift is more severe if it has lower AdaBN performance (see Appendix K.5). On EMNIST-DA, BUFR convincingly outperforms all other methods—particularly on severe shifts where the initial feature-space class-separation is likely poor. Note the large deviation in performance across random runs for SHOT-IM and SHOT, suggesting that initial feature-space clustering has a big impact on how well these entropy-minimization methods can separate the target data. This is particularly true for the severe shift, where only BUFR achieves high accuracy across random runs. For the mild shift, where all methods perform well, we still see that: (i) BUFR performs the best; and (ii) PL, BNM-IM, SHOT-IM and SHOT are poorly calibrated due to their entropy-minimizing (i.e. confidence-maximizing) objectives. In fact, these methods are only reasonably calibrated if accuracy is very high. In contrast, our methods, and other methods that lack entropy terms (AdaBN, Marg. Gauss., Full Gauss.), maintain reasonable calibration as they do not work by making predictions more confident. This point is elucidated in the reliability diagrams of Appendix H.

## 5.3 OBJECT-RECOGNITION RESULTS

Table 2 reports classification accuracies and ECEs for CIFAR-10-C and CIFAR-100-C. Here we observe that FR is competitive with existing SFDA methods, while BUFR outperforms them on almost all fronts (except for ECE on CIFAR-100-C). We also observe the same three trends as on EMNIST-DA: (i) while the entropy-minimizing methods (PL, SHOT-IM, TENT) do well in terms of accuracy, their confidence-maximizing objectives lead to higher ECE—particularly on CIFAR-100-C where their ECE is even higher than that of the unadapted source-only model; (ii) the addition of bottom-up training significantly boosts performance; (iii) BUFR gets the largest boost on the most severe shifts—for example, as shown in the full per-shift results of Appendix K.6, BUFR achieves 89% accuracy on the impulse-noise shift of CIFAR-10-C, with the next best SFDA method achieving just 75%. Surprisingly, BUFR even outperforms target-supervised fine-tuning on both CIFAR-10-C and CIFAR-100-C in terms of accuracy. We attribute this to the regularization effect of bottom-up training, which we explore further in the next section.

We also report results for the “online” setting of Wang et al. (2021), where we may only use a single pass through the target data, applying mini-batch updates along the way. As shown in Table 13 of Appendix K.2, FR outperforms existing SFDA methods on CIFAR-10-C and is competitive on CIFAR-100-C. This includes TENT (Wang et al., 2021)—a method designed specifically for this online setting.

## 5.4 REAL-WORLD RESULTS

Table 4 reports results on CAMELYON17—a dataset containing real-world (i.e. naturally occurring) measurement shift. Here we report the average classification accuracy over 4 target hospitals. Note that the accuracy on the source hospital (i.e. no corruption) was 99.3%. Also note that this particular dataset is an ideal candidate for entropy-minimization techniques due to: (i) high AdaBN accuracy on the target data (most pseudo-labels are correct since updating only the BN-statistics gives  $\sim$ 84%); (ii) a low number of classes (random pseudo-labels have a 50% chance of being correct); and (iii) a large target dataset. Despite this, our methods achieve competitive accuracy and show greater data efficiency—with 50 examples-per-class or less, only our methods meaningfully improve upon the simple AdaBN baseline which uses the target-data BN-statistics. These results illustrate that: (i) our method performs

Published as a conference paper at ICLR 2022

Table 2: Object-recognition results. \*: result adopted from Wang et al. (2021).

Model	CIFAR-10-C		CIFAR-100-C	
	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$
No corruption	95.3 $\pm$ 0.2	2.4 $\pm$ 0.1	76.4 $\pm$ 0.2	4.8 $\pm$ 0.1
DANN* (Ganin et al., 2016)	81.7	-	61.1	-
UDA-SS.* (Sun et al., 2019)	83.3	-	53	-
Source-only	57.8 $\pm$ 0.7	28.2 $\pm$ 0.4	36.4 $\pm$ 0.5	19.4 $\pm$ 0.9
AdaBN (Li et al., 2018)	80.4 $\pm$ 0.1	11.2 $\pm$ 0.1	56.6 $\pm$ 0.3	<b>12.5 <math>\pm</math> 0.1</b>
PL (Lee et al., 2013)	82.5 $\pm$ 0.3	17.5 $\pm$ 0.3	62.1 $\pm$ 0.2	37.7 $\pm$ 0.2
SHOT-IM (Liang et al., 2020)	85.4 $\pm$ 0.2	14.6 $\pm$ 0.2	67.0 $\pm$ 0.2	32.9 $\pm$ 0.2
TENT (Wang et al., 2021)	86.6 $\pm$ 0.3	12.8 $\pm$ 0.3	66.0 $\pm$ 0.4	25.7 $\pm$ 0.4
FR (ours)	87.2 $\pm$ 0.7	11.3 $\pm$ 0.3	65.5 $\pm$ 0.2	15.7 $\pm$ 0.1
BUFR (ours)	<b>89.4 <math>\pm</math> 0.2</b>	<b>10.0 <math>\pm</math> 0.2</b>	<b>68.5 <math>\pm</math> 0.2</b>	14.5 $\pm$ 0.3
Target-supervised	88.4 $\pm$ 0.9	6.4 $\pm$ 0.6	68.1 $\pm$ 1.2	9.6 $\pm$ 0.7

Table 3: EMNIST-DA degree of restoration.

Model	D
Source-only.	3.2 $\pm$ 0.0
AdaBN	3.1 $\pm$ 0.1
Marg. Gauss.	2.9 $\pm$ 0.0
Full Gauss.	2.0 $\pm$ 0.0
PL	2.6 $\pm$ 0.0
BNM-IM	2.5 $\pm$ 0.1
SHOT-IM	2.9 $\pm$ 0.1
FR (ours)	1.8 $\pm$ 0.0
BUFR (ours)	<b>1.2 <math>\pm</math> 0.0</b>

Table 4: CAMELYON<sub>17</sub> accuracies for a varying number of examples-per-class in the target domain.

Model	5	10	50	500	All(> 15k)
Source-only	55.8 $\pm$ 1.6	55.8 $\pm$ 1.6	55.8 $\pm$ 1.6	55.8 $\pm$ 1.6	55.8 $\pm$ 1.6
AdaBN (Li et al., 2018)	82.6 $\pm$ 2.2	83.3 $\pm$ 2.3	83.7 $\pm$ 1.0	83.9 $\pm$ 0.8	84.0 $\pm$ 0.5
PL (Lee et al., 2013)	82.5 $\pm$ 2.0	83.7 $\pm$ 1.7	83.6 $\pm$ 1.2	85.0 $\pm$ 0.8	<b>90.6 <math>\pm</math> 0.9</b>
SHOT-IM (Liang et al., 2020)	82.6 $\pm$ 2.2	83.4 $\pm$ 2.5	83.7 $\pm$ 1.2	86.4 $\pm$ 0.7	89.9 $\pm$ 0.2
FR (ours)	<b>84.6 <math>\pm</math> 0.6</b>	86.0 $\pm$ 0.7	86.0 $\pm$ 1.1	89.0 $\pm$ 0.6	89.5 $\pm$ 0.4
BUFR (ours)	84.5 $\pm$ 0.8	<b>86.1 <math>\pm</math> 0.2</b>	<b>87.0 <math>\pm</math> 1.2</b>	<b>89.1 <math>\pm</math> 0.8</b>	89.7 $\pm$ 0.5

well in practice; (ii) measurement shift is an important real-world problem; and (iii) source-free methods are important to address such measurement shifts as, e.g., medical data is often kept private.

### 5.5 ANALYSIS

**Feature-space class-separation.** Measurement shifts can cause the target data to be poorly-separated in feature space. This point is illustrated in Figure 3 where we provide t-SNE visualizations of the feature-space class-separation on the EMNIST-DA crystals shift. Here, Figure 3a shows the initial class-separation *before* adapting the source model. We see that the source data is well separated in feature space (dark colours) but the target data is not (light colours). Figure 3b shows the performance of an entropy-minimization method when applied to such a “degraded” feature space where initial class-separation is poor on the target data. While accuracy and class-separation improve, the target-data clusters are not yet (i) fully homogeneous and (ii) returned to their original location (that of the source-data clusters). As shown in Figure 3(c,d), our methods of FR and BUFR better restore class-separation on the target data with more homogeneous clusters returned to their previous location.

**Quantifying the degree of restoration.** We quantify the degree to which the EMNIST source features are *restored* in each of the EMNIST-DA target domains by calculating the average pairwise distance:  $D = \frac{1}{T} \sum_{t=1}^T \frac{1}{N} \sum_{i=1}^N |g_s(m_s(X^{(i)})) - g_t(m_t(X^{(i)}))|$ , where  $T$  is the number of EMNIST-DA target domains,  $N$  is the number of EMNIST images,  $X^{(i)}$  is a clean or uncorrupted EMNIST image,  $m_s$  is the identity transform, and  $m_t$  is the shift of target domain  $t$  (e.g. Gaussian blur). Table 3 shows that the purely alignment-based methods (Marg. Gauss., Joint Gauss., FR, BUFR) tend to better restore the features than the entropy-based methods (PL, BNM-IM, SHOT-IM), with our alignment-based methods doing it best. The only exception is Marg. Gauss.—the weakest form of alignment. Finally, it is worth noting the strong rank correlation (0.6) between the degree of restoration in Table 3 and the ECE in Table 1. This confirms that, for measurement shifts, it is preferable to restore the same features rather than learn new ones as the latter usually comes at the cost of model calibration.

**Restoring the semantic meaning of features.** The left column of Figure 4a shows the activation distribution (bottom) and maximally-activating image patches (top) for a specific filter in the first layer of a CNN trained on the standard EMNIST dataset (white digit, black background). The centre column shows that, when presented with shifted target data (pink digit, green background), the filter detects similar patterns of light and dark colours but no longer carries the same semantic meaning of detecting a horizontal edge. Finally, the right column shows that, when our BUFR method aligns the marginal feature distributions on the target data (orange curve, bottom) with those saved on the source data (blue curve, bottom), this restores a sense of semantic meaning to the filters (image patches, top). Note that we explicitly align the *first-layer* feature/filter distributions in this illustrative experiment.

Published as a conference paper at ICLR 2022

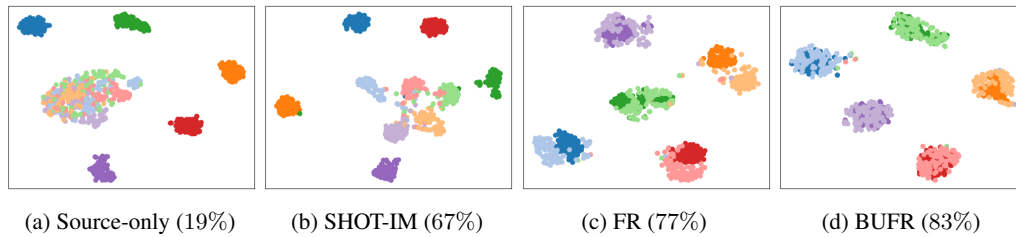


Figure 3: t-SNE (Van der Maaten & Hinton, 2008) visualization of features for 5 classes of the EMNIST-DA crystals shift. Dark colours show the source data, light the target. Model accuracies are shown in parentheses.

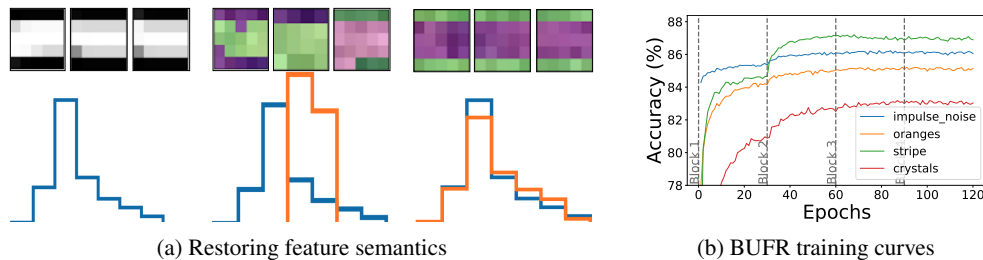


Figure 4: (a) Activation distributions (bottom) and maximally-activating image patches (top) for a specific filter in the first layer of a CNN. *Left*: Source model, source data (white digit, black backgr.). *Centre*: Source model, target data (pink digit, green backgr.). *Right*: Target model (adapted with BUFR), target data. (b) BUFR training curves on selected EMNIST-DA corruptions. Dashed-grey lines indicate when the next block is unfrozen.

**Efficacy of BU training.** Figure 4b shows that, when training in a bottom-up manner, updating only the first two blocks is sufficient to resolve many measurement shifts. This confirms the previous intuition that updating only the early layers should be sufficient for many measurement shifts. BUFR exploits this by primarily updating early layers, thus preserving learnt structure in later layers (see Appendix J.3–J.4). To examine the regularization benefits of this structure preservation, we compare the accuracy of BUFR to other SFDA methods as the number of available target examples reduces. As shown in Table 9 of Appendix J.1, the performance of all competing methods drops sharply as we reduce the number of target examples. In contrast, BUFR maintains strong performance. With only 5 examples-per-class, it surpasses the performance of many methods using all 400 examples-per-class.

**Ablation study.** We also conduct an ablation study on the components of our loss from Equation 2. Table 10 of Appendix J.2 shows that, for easier tasks like CIFAR-10-C, aligning the logit distributions and using the symmetric KL divergence (over a more commonly-used asymmetric one) make little difference to performance. However, for harder tasks like CIFAR-100-C, both improve performance.

## 6 DISCUSSIONS

**Aligning the marginals may be insufficient.** Our method seeks to restore the joint feature distribution by aligning (approximations of) the marginals. While we found that this is often sufficient, it cannot be guaranteed unless the features are independent. One potential remedy is to encourage feature independence in the source domain using “disentanglement” (Bengio et al., 2013; Eastwood & Williams, 2018) methods, allowing the marginals to better capture the joint.

**Model selection.** Like most UDA & SFDA works, we use a target-domain validation set (Gulrajani & Lopez-Paz, 2021) for model selection. However, such labelled target data is rarely available in real-world setups. Potential solutions include developing benchmarks (Gulrajani & Lopez-Paz, 2021) and validation procedures (You et al., 2019) that allow more realistic model selection and comparison.

**Conclusion.** We have proposed BUFR, a method for source-free adaptation to measurement shifts. BUFR works by aligning histogram-based approximations of the marginal feature distributions on the target data with those saved on the source. We showed that, by focusing on measurement shifts, BUFR can outperform existing methods in terms of accuracy, calibration and data efficiency, while making less assumptions about the behaviour of the source model on the target data. We also highlighted issues with the entropy-minimization techniques on which existing SFDA-methods rely, namely their classification-specificity, tendency to be poorly calibrated, and vulnerability to simple but severe shifts.

## ACKNOWLEDGEMENTS

We thank Tim Hospadales, Amos Storkey, Oisín Mac Aodha, Luigi Gresele and Julius von Kügelgen for helpful discussions and comments. CE acknowledges support from The National University of Ireland via his Travelling Studentship in the Sciences. IM is supported by the Engineering and Physical Sciences Research Council (EPSRC).

## REFERENCES

- Ehab A AlBadawy, Ashirbani Saha, and Maciej A Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical Physics*, 45(3):1150–1158, 2018.
- Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, May 2011.
- Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermesen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2018.
- Emma Beede, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M. Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–12. Association for Computing Machinery, 2020.
- Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 137–144, 2007.
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
- Boris Chidlovskii, Stéphane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 451–460, 2016.
- Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.
- Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *International Conference on Intelligent Transportation Systems*, pp. 3819–3824. IEEE, 2018.
- Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 129–136, 2010.
- Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.
- 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*, pp. 16458–16468, 2021.
- James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, pp. 194–202, 1995.

Published as a conference paper at ICLR 2022

---

- John Duchi. Derivations for linear algebra and optimization, 2007. URL [https://web.stanford.edu/~jduchi/projects/general\\_notes.pdf](https://web.stanford.edu/~jduchi/projects/general_notes.pdf). Accessed: 5<sup>th</sup> October 2021.
- Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.
- Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pp. 1802–1811, 2019.
- Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pp. 1180–1189. PMLR, 2015.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020.
- Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.
- Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.
- 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*, pp. 770–778, 2016.
- Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.
- Masato Ishii and Masashi Sugiyama. Source-free domain adaptation via distributional alignment by matching batch normalization statistics. *arXiv preprint arXiv:2101.10842*, 2021.
- Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.
- Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems*, pp. 775–783, 2010.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4544–4553, 2020.



Published as a conference paper at ICLR 2022

---

- Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 615–625, 2021.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, 2013.
- Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9641–9650, 2020.
- Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations Workshop*, 2017.
- Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.
- Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 6028–6039, July 13–18 2020.
- Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pp. 97–105, 2015.
- Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018.
- C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *Machine Learning for Autonomous Driving Workshop, NeurIPS 2019*, 2019.
- Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45:521–530, 2012.
- Pietro Morerio, Riccardo Volpi, Ruggero Ragonesi, and Vittorio Murino. Generative pseudo-label refinement for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3130–3139, 2020.
- Norman Mu and Justin Gilmer. MNIST-C: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.
- Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.
- Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *International Conference on Machine Learning*, pp. 625–632, 2005.
- Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. VISDA: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2021–2026, 2018.
- Joaquin Quiñonero-Candela, Masashi Sugiyama, Neil D Lawrence, and Anton Schwaighofer. *Dataset Shift in Machine Learning*. MIT Press, 2009.
- Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2019.
- Amélie Royer and Christoph Lampert. A flexible selection scheme for minimum-effort transfer learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2191–2200, 2020.

Published as a conference paper at ICLR 2022

---

- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pp. 213–226. Springer, 2010.
- Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.
- Patrice Simard, Bernard Victorri, Yann LeCun, and John S Denker. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In *Advances in Neural Information Processing Systems*, pp. 895–903, 1991.
- Serban Stan and Mohammad Rostami. Unsupervised model adaptation for continual semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2593–2601, 2021.
- Amos J Storkey. When training and test sets are different: characterising learning transfer. In *Dataset Shift in Machine Learning*, pp. 3–28. MIT Press, 2009.
- Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pp. 443–450. Springer, 2016.
- Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.
- David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101544, 2019.
- Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1521–1528, 2011.
- Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.
- Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.
- Georg Volk, Stefan Müller, Alexander von Bernuth, Dennis Hospach, and Oliver Bringmann. Towards robust CNN-based object detection through augmentation with synthetic rain variations. In *IEEE Intelligent Transportation Systems Conference*, pp. 285–292, 2019.
- Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. TENT: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.
- Yongxin Yang, Irene Garcia Morillo, and Timothy M. Hospedales. Deep neural decision trees. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2018.
- Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. SoFA: Source-data-free feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 474–483, 2021.
- Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.
- Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 7124–7133, 2019.

Published as a conference paper at ICLR 2022

---

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, pp. 609–616, 2001.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Medicine*, 15(11):e1002683, 2018.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833, 2014.

### 3.3 Comments on the paper

**Dealing with different shift types, adaptively.** Under the assumption of measurement shift, we fix the last layer and fine-tune the rest, restoring the same high-level features or concepts by learning new low-level features. In contrast, the standard fine-tuning approach is to fine-tune the last layer and fix the rest, learning new high-level features or concepts by recombining the same low-level features. Together, these approaches suggest that different layers should be updated depending on *what* should be transferred, or, more specifically, the type of shift encountered. This in turn raises some interesting questions: *given a shift, which layers should we update? Can we determine which layers to update automatically?* By fine-tuning each layer of a network *individually* on a small set of labelled test-domain samples, and selecting the one that yields the best performance, [Royer and Lampert \(2020\)](#) found that, for many domain shifts, it can be best to fine-tune an early or intermediate layer rather than the final one. Taking this idea further, [Eastwood et al. \(2021\)](#) found that *unit-level surprise* in neural networks can be used to reveal the layer (or level of abstraction) at which a given shift is “noticed”, and that this, in turn, can be used to devise automatic, shift-dependent fine-tuning strategies.

**Storing information about the source dataset.** We address the setting of SFDA where the source dataset is not available at deployment time. To do so, we store some lightweight statistics of the source dataset. In principle, one could store any statistics of (or amount of information about) the source data distribution and thus employ prior distribution-alignment methods from the unsupervised domain adaptation (UDA) literature. This then leads to a number of questions which would be interesting to explore:

1. *How much information about the source dataset can we (or do we want to) store?*

The answers to these questions may be determined by privacy and/or storage constraints, or perhaps by their trade-offs with performance. At one extreme is keeping everything, i.e., the entire dataset. At the other end is keeping nothing, i.e., no storage of any kind. In between, one could imagine a spectrum:

- Keep a sub-sample of the training set
- Keep a statistic computed from the training set
- Keep the parameters of a simple parametric model (fit to the training set)
- Keep the parameters of a complex parametric model (fit to the training set)

2. *What are the storage-performance trade-offs?* Can we control these trade-offs in an interpretable manner, ideally along a Pareto frontier? Are there noticeable differences between the trade-offs of different methods? For fair comparison across different forms, storage should likely be measured using compressed size.
3. *What are the storage-privacy trade-offs?* How much is privacy preserved at each storage level, e.g., with just a few histogram bin counts? Or, more specifically, to what extent can the original source dataset be reconstructed at each storage level? While differential privacy is often the standard notion of dataset privacy ([Dwork et al., 2006, 2014](#)), its use of multiple queries does not cleanly map onto the SFDA setting where the dataset statistics would just be released once (no queries or updates).

# 4

## Domain Generalisation: A Probabilistic Framework

This chapter focuses on *domain generalisation* (DG, [Blanchard et al. 2011](#); [Muandet et al. 2013](#)). Here, a model is trained on data from multiple related domains (e.g., hospitals) with the goal of performing well on data from other related but unseen test domains. In general, preparation involves exploiting invariances across the training domains in the hope that these invariances also hold in unseen test domains. Prior works have sought to do so by learning models that perform well *on-average* ([Blanchard et al., 2021](#); [Zhang et al., 2021](#)) or *in-the-worst-case* ([Arjovsky et al., 2019](#); [Sagawa\\* et al., 2020](#)). However, models that perform well on average can lack robustness ([Nagarajan et al., 2021](#)), while models that perform well in the worst case can be overly conservative ([Tsipras et al., 2019](#)). To address these issues, we propose a new probabilistic framework wherein the goal is to learn models that *perform well with high probability*. In particular, by explicitly relating the training and test domains as draws from the same underlying meta-distribution, we ensure that distribution shifts seen during training inform us of *probable* shifts at test time. Then, by minimising a particular quantile of a model’s performance distribution over training domains, we learn models that perform well on unseen test domains with the corresponding probability.

To reinforce this new probabilistic perspective and objective for DG, we highlight the importance of comparing DG algorithms based on their tail or quantile performance over multiple test domains. In particular, we question the common practice ([Gulrajani and Lopez-Paz, 2020](#); [Koh et al., 2021](#)) of comparing DG algorithms in terms of average- or single-test-domain performance, since improved robustness or tail-performance is often invisible through these lenses.

Finally, we draw new fundamental connections between invariance and causality by proving that: (i) our algorithm learns a predictor with *invariant performance* over domains as the desired probability of generalisation approaches one; and (ii) this is sufficient to recover the causal predictor under weaker assumptions than prior work (Krueger et al., 2021; Peters et al., 2016).

## 4.1 Contribution

I led this project from conceptualisation to final form. In particular, I was heavily involved in coming up with the initial idea, formalising the learning objective and algorithm, drawing the connection to causality, designing the experimental analyses, running the experimental analyses, and writing the manuscript. Some of these tasks were shared with Alexander Robey, with whom I share first authorship. I was not involved in the learning theory of Section 4.2: this was the work of Shashank Singh. While I was involved with the causal-recovery theory of Section 4.3, the main theoretical result (Theorem 4.4) was the work of Shashank Singh.

## 4.2 Paper

---

# Probable Domain Generalization via Quantile Risk Minimization

---

Cian Eastwood<sup>\*1,2</sup> Alexander Robey<sup>\*3</sup> Shashank Singh<sup>1</sup>

Julius von Kügelgen<sup>1,4</sup> Hamed Hassani<sup>3</sup> George J. Pappas<sup>3</sup> Bernhard Schölkopf<sup>1</sup>

<sup>1</sup> Max Planck Institute for Intelligent Systems, Tübingen

<sup>2</sup> University of Edinburgh   <sup>3</sup> University of Pennsylvania   <sup>4</sup> University of Cambridge

## Abstract

Domain generalization (DG) seeks predictors which perform well on unseen test distributions by leveraging data drawn from multiple related training distributions or domains. To achieve this, DG is commonly formulated as an average- or worst-case problem over the set of possible domains. However, predictors that perform well on average lack robustness while predictors that perform well in the worst case tend to be overly-conservative. To address this, we propose a new probabilistic framework for DG where the goal is to learn predictors that perform well *with high probability*. Our key idea is that distribution shifts seen during training should inform us of probable shifts at test time, which we realize by explicitly relating training and test domains as draws from the same underlying meta-distribution. To achieve probable DG, we propose a new optimization problem called *Quantile Risk Minimization* (QRM). By minimizing the  $\alpha$ -quantile of predictor’s risk distribution over domains, QRM seeks predictors that perform well with probability  $\alpha$ . To solve QRM in practice, we propose the *Empirical QRM* (EQRM) algorithm and provide: (i) a generalization bound for EQRM; and (ii) conditions under which EQRM recovers the causal predictor as  $\alpha \rightarrow 1$ . In our experiments, we introduce a more holistic quantile-focused evaluation protocol for DG and demonstrate that EQRM outperforms state-of-the-art baselines on datasets from WILDS and DomainBed.

## 1 Introduction

Despite remarkable successes in recent years [1–3], machine learning systems often fail calamitously when presented with *out-of-distribution* (OOD) data [4–7]. Evidence of state-of-the-art systems failing in the face of distribution shift is mounting rapidly—be it due to spurious correlations [8–10], changing sub-populations [11–13], changes in location or time [14–16], or other naturally-occurring variations [17–23]. These OOD failures are particularly concerning in safety-critical applications such as medical imaging [24–28] and autonomous driving [29–31], where they represent one of the most significant barriers to the real-world deployment of machine learning systems [32–35].

Domain generalization (DG) seeks to improve a system’s OOD performance by leveraging datasets from multiple environments or domains at training time, each collected under different experimental conditions [36–38] (see Fig. 1a). The goal is to build a predictor which exploits invariances across the training domains in the hope that these invariances also hold in related but distinct test domains [38–41]. To realize this goal, DG is commonly formulated as an average- [36, 42, 43] or worst-case [9, 44, 45] optimization problem over the set of possible domains. However, optimizing for average performance can lack robustness to OOD data [46], while optimizing for worst-domain performance tends to lead to overly-conservative solutions, with worst-case outcomes unlikely in practice [47, 48].

---

<sup>\*</sup>Equal contribution. Correspondence to

or

Code available at: <https://github.com/cianeastwood/qrm>



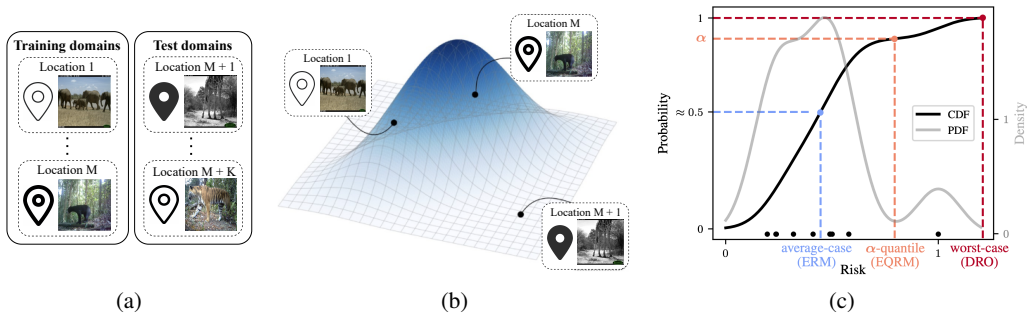


Figure 1: **Overview of Probable Domain Generalization and Quantile Risk Minimization.** (a) In domain generalization, training and test data are drawn from multiple related distributions or domains. For example, in the iWildCam dataset [50], which contains camera-trap images of animal species, the domains correspond to the different camera-traps which captured the images. (b) We relate training and test domains as draws from the same underlying (and often unknown) meta-distribution over domains  $\mathcal{Q}$ . (c) We consider a predictor’s estimated risk distribution over training domains, naturally-induced by  $\mathcal{Q}$ . By minimizing the  $\alpha$ -quantile of this distribution, we learn predictors that perform well with high probability ( $\approx \alpha$ ) rather than on average or in the worst case.

In this work, we argue that DG is neither an average-case nor a worst-case problem, but rather a probabilistic one. To this end, we propose a probabilistic framework for DG, which we call *Probable Domain Generalization* (§ 3), wherein the key idea is that distribution shifts seen during training should inform us of *probable* shifts at test time. To realize this, we explicitly relate training and test domains as draws from the same underlying meta-distribution (Fig. 1b), and then propose a new optimization problem called *Quantile Risk Minimization* (QRM). By minimizing the  $\alpha$ -quantile of predictor’s risk distribution over domains (Fig. 1c), QRM seeks predictors that perform well *with high probability* rather than on average or in the worst case. In particular, QRM leverages the key insight that this  $\alpha$ -quantile is an upper bound on the test-domain risk which holds with probability  $\alpha$ , meaning that  $\alpha$  is an interpretable conservativeness-hyperparameter with  $\alpha = 1$  corresponding to the worst-case setting.

To solve QRM in practice, we introduce the *Empirical QRM* (EQRm) algorithm (§ 4). Given a predictor’s empirical risks on the training domains, EQRm forms an estimated risk distribution using kernel density estimation (KDE, [49]). Importantly, KDE-smoothing ensures a right tail that extends beyond the largest training risk (see Fig. 1c), with this risk “extrapolation” [41] unlocking *invariant prediction* for EQRm (§ 4.1). We then provide theory for EQRm (§ 4.2, § 4.3) and demonstrate empirically that EQRm outperforms state-of-the-art baselines on real and synthetic data (§ 6).

**Contributions.** To summarize our main contributions:

- *A new probabilistic perspective and objective for DG:* We argue that predictors should be trained and tested based on their ability to perform well *with high probability*. We then propose Quantile Risk Minimization for achieving this *probable* form of domain generalization (§ 3).
- *A new algorithm:* We propose the EQRm algorithm to solve QRM in practice and ultimately learn predictors that generalize with probability  $\alpha$  (§ 4). We then provide several analyses of EQRm:
  - *Learning theory:* We prove a uniform convergence bound, meaning the empirical  $\alpha$ -quantile risk tends to the population  $\alpha$ -quantile risk given sufficiently many domains and samples (Thm. 4.1).
  - *Causality.* We prove that EQRm learns predictors with invariant risk as  $\alpha \rightarrow 1$  (Prop. 4.3), then provide conditions under which this is sufficient to recover the causal predictor (Thm. 4.4).
  - *Experiments:* We demonstrate that EQRm outperforms state-of-the-art baselines on several standard DG benchmarks, including CMNIST [9] and datasets from WILDS [12] and DomainBed [38], and highlight the importance of assessing the tail or *quantile performance* of DG algorithms (§ 6).

## 2 Background: Domain generalization

**Setup.** In domain generalization (DG), predictors are trained on data drawn from multiple related training distributions or *domains* and then evaluated on related but unseen test domains. For example, in the iWildCam dataset [50], the task is to classify animal species in images, and the domains correspond to the different camera-traps which captured the images (see Fig. 1a). More formally, we consider datasets  $D^e = \{(x_i^e, y_i^e)\}_{i=1}^{n_e}$  collected from  $m$  different training domains or *environments*  $\mathcal{E}_{\text{tr}} := \{e_1, \dots, e_m\}$ , with each dataset  $D^e$  containing data pairs  $(x_i^e, y_i^e)$  sampled i.i.d. from

$\mathbb{P}(X^e, Y^e)$ . Then, given a suitable function class  $\mathcal{F}$  and loss function  $\ell$ , the goal of DG is to learn a predictor  $f \in \mathcal{F}$  that generalizes to data drawn from a larger set of all possible domains  $\mathcal{E}_{\text{all}} \supset \mathcal{E}_{\text{tr}}$ .

**Average case.** Letting  $\mathcal{R}^e(f)$  denote the statistical risk of  $f$  in domain  $e$ , and  $\mathbb{Q}$  a distribution over the domains in  $\mathcal{E}_{\text{all}}$ , DG was first formulated [36, 37] as the following average-case problem:

$$\min_{f \in \mathcal{F}} \mathbb{E}_{e \sim \mathbb{Q}} \mathcal{R}^e(f) \quad \text{where} \quad \mathcal{R}^e(f) := \mathbb{E}_{\mathbb{P}(X^e, Y^e)}[\ell(f(X^e), Y^e)]. \quad (2.1)$$

**Worst case.** Since predictors that perform well *on average* can lack robustness [46], i.e. they can perform quite poorly on large subsets of  $\mathcal{E}_{\text{all}}$ , subsequent works [9, 22, 41, 44, 45, 51] have sought robustness by formulating DG as the following *worst-case* problem:

$$\min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}_{\text{all}}} \mathcal{R}^e(f). \quad (2.2)$$

As we only have access to data from a finite subset of  $\mathcal{E}_{\text{all}}$  during training, solving (2.2) is not just challenging but in fact impossible [41, 52, 53] without restrictions on how the domains may differ.

**Causality and invariance in DG.** Causal works on DG [9, 41, 53–55] describe domain differences using the language of causality and the notion of *interventions* [56, 57]. In particular, they assume all domains share the same underlying *structural causal model* (SCM) [56], with different domains corresponding to different interventions (see Appendix A.1 for formal definitions and a simple example). Assuming the mechanism of  $Y$  remains fixed or invariant but all  $X$ s may be intervened upon, recent works have shown that only the causal predictor has invariant: (i) predictive distributions [54], coefficients [9] or risks [41] across domains; and (ii) generalizes to arbitrary interventions on the  $X$ s [9, 54, 55]. These works then leverage some form of invariance across domains to discover causal relationships which, through the invariant mechanism assumption, generalize to new domains.

### 3 Quantile Risk Minimization

In this section we introduce *Quantile Risk Minimization* (QRM) for achieving *Probable Domain Generalization*. The core idea is to replace the worst-case perspective of (2.2) with a probabilistic one. This approach is founded on a great deal of work in classical fields such as control theory [58, 59] and smoothed analysis [60], wherein approaches that yield high-probability guarantees are used in place of worst-case approaches in an effort to mitigate conservatism and computational limitations. This mitigation is of particular interest in domain generalization since generalizing to arbitrary domains is impossible [41, 52, 53]. Thus, motivated by this classical literature, our goal is to obtain predictors that are robust *with high probability* over domains drawn from  $\mathcal{E}_{\text{all}}$ , rather than in the worst case.

**A distribution over environments.** We start by assuming the existence of a probability distribution  $\mathbb{Q}(e)$  over the set of all environments  $\mathcal{E}_{\text{all}}$ . For instance, in the context of medical imaging,  $\mathbb{Q}$  could represent a distribution over potential changes to a hospital’s setup or simply a distribution over candidate hospitals. Given that such a distribution  $\mathbb{Q}$  exists<sup>2</sup>, we can think of the risk  $\mathcal{R}^e(f)$  as a *random variable* for each  $f \in \mathcal{F}$ , where the randomness is engendered by the draw of  $e \sim \mathbb{Q}$ . This perspective gives rise to the following analogue of the optimization problem in (2.2):

$$\min_{f \in \mathcal{F}} \text{ess sup}_{e \sim \mathbb{Q}} \mathcal{R}^e(f) \quad \text{where} \quad \text{ess sup}_{e \sim \mathbb{Q}} \mathcal{R}^e(f) = \inf \left\{ t \geq 0 : \Pr_{e \sim \mathbb{Q}} \{ \mathcal{R}^e(f) \leq t \} = 1 \right\} \quad (3.1)$$

Here, *ess sup* denotes the *essential-supremum* operator from measure theory, meaning that for each  $f \in \mathcal{F}$ ,  $\text{ess sup}_{\mathbb{Q}} \mathcal{R}^e(f)$  is the least upper bound on  $\mathcal{R}^e(f)$  that holds for almost every  $e \sim \mathbb{Q}$ . In this way, the *ess sup* in (3.1) is the measure-theoretic analogue of the *max* operator in (2.2), with the subtle but critical difference being that the *ess sup* in (3.1) can neglect domains of measure zero under  $\mathbb{Q}$ . For example, for discrete  $\mathbb{Q}$ , (3.1) ignores domains which are impossible (i.e. have probability zero) while (2.2) does not, laying the foundation for ignoring domains which are *improbable*.

**High-probability generalization.** Although the minimax problem in (3.1) explicitly incorporates the distribution  $\mathbb{Q}$  over environments, this formulation is no less conservative than (2.2). Indeed, in many cases, (3.1) is equivalent to (2.2); see Appendix B for details. Therefore, rather than considering the worst-case problem in (3.1), we propose the following generalization of (3.1) which requires that predictors generalize with probability  $\alpha$  rather than in the worst-case:

$$\min_{f \in \mathcal{F}, t \in \mathbb{R}} t \quad \text{subject to} \quad \Pr_{e \sim \mathbb{Q}} \{ \mathcal{R}^e(f) \leq t \} \geq \alpha \quad (3.2)$$

<sup>2</sup>As  $\mathbb{Q}$  is often unknown, our analysis does not rely on using an explicit expression for  $\mathbb{Q}$ .

The optimization problem in (3.2) formally defines what we mean by *Probable Domain Generalization*. In particular, we say that a predictor  $f$  generalizes with risk  $t$  at level  $\alpha$  if  $f$  has risk at most  $t$  with probability at least  $\alpha$  over domains sampled from  $\mathbb{Q}$ . In this way, the conservativeness parameter  $\alpha$  controls the strictness of generalizing to unseen domains.

**A distribution over risks.** The optimization problem presented in (3.2) offers a principled formulation for generalizing to unseen distributional shifts governed by  $\mathbb{Q}$ . However,  $\mathbb{Q}$  is often unknown in practice and its support  $\mathcal{E}_{\text{all}}$  may be high-dimensional or challenging to define [22]. While many previous works have made progress by limiting the scope of possible shift types over domains [19, 22, 45], in practice, such structural assumptions are often difficult to justify and impossible to test. For this reason, we start our exposition of QRM by offering an alternative view of (3.2) which elucidates how a predictor’s *risk distribution* plays a central role in achieving probable domain generalization.

To begin, note that for each  $f \in \mathcal{F}$ , the distribution over domains  $\mathbb{Q}$  naturally induces<sup>3</sup> a distribution  $\mathbb{T}_f$  over the risks in each domain  $\mathcal{R}^e(f)$ . In this way, rather than considering the randomness of  $\mathbb{Q}$  in the often-unknown and (potentially) high-dimensional space of possible shifts (Fig. 1b), one can consider it in the real-valued space of risks (Fig. 1c). This is analogous to statistical learning theory, where the analysis of convergence of empirical risk minimizers (i.e., of functions) is substituted by that of a weaker form of convergence, namely that of scalar risk functionals—a crucial step for VC theory [61]. From this perspective, the statistics of  $\mathbb{T}_f$  can be thought of as capturing the sensitivity of  $f$  to different environmental shifts, summarizing the effect of different intervention types, strengths, and frequencies. To this end, (3.2) can be equivalently rewritten in terms of the risk distribution  $\mathbb{T}_f$  as follows:

$$\min_{f \in \mathcal{F}} F_{\mathbb{T}_f}^{-1}(\alpha) \quad \text{where} \quad F_{\mathbb{T}_f}^{-1}(\alpha) := \inf \left\{ t \in \mathbb{R} : \Pr_{R \sim \mathbb{T}_f} \{R \leq t\} \geq \alpha \right\}. \quad (\text{QRM})$$

Here,  $F_{\mathbb{T}_f}^{-1}(\alpha)$  denotes the inverse CDF (or quantile<sup>4</sup>) function of the risk distribution  $\mathbb{T}_f$ . By means of this reformulation, we elucidate how solving (QRM) amounts to finding a predictor with minimal  $\alpha$ -quantile risk. That is, (QRM) requires that a predictor  $f$  satisfy the probabilistic constraint for at least an  $\alpha$ -fraction of the risks  $R \sim \mathbb{T}_f$ , or, equivalently, for an  $\alpha$ -fraction of the environments  $e \sim \mathbb{Q}$ . In this way,  $\alpha$  can be used to interpolate between typical ( $\alpha = 0.5$ , median) and worst-case ( $\alpha = 1$ ) problems in an interpretable manner. Moreover, if the mean and median of  $\mathbb{T}_f$  coincide,  $\alpha = 0.5$  gives an average-case problem, with (QRM) recovering several notable objectives for DG as special cases.

**Proposition 3.1.** For  $\alpha = 1$ , (QRM) is equivalent to the worst-case problem of (3.1). For  $\alpha = 0.5$ , it is equivalent to the average-case problem of (2.1) if the mean and median of  $\mathbb{T}_f$  coincide  $\forall f \in \mathcal{F}$ :

$$\min_{f \in \mathcal{F}} \mathbb{E}_{R \sim \mathbb{T}_f} R = \min_{f \in \mathcal{F}} \mathbb{E}_{e \sim \mathbb{Q}} \mathcal{R}^e(f) \quad (3.3)$$

**Connection to DRO.** While fundamentally different in terms of objective and generalization capabilities (see § 4), we draw connections between QRM and distributionally robust optimization (DRO) in Appendix F by considering an alternative problem which optimizes the *superquantile*.

## 4 Algorithms for Quantile Risk Minimization

We now introduce the *Empirical QRM* (EQRM) algorithm for solving (QRM) in practice, akin to Empirical Risk Minimization (ERM) solving the Risk Minimization (RM) problem [63].

### 4.1 From QRM to Empirical QRM

In practice, given a predictor  $f$  and its empirical risks  $\hat{\mathcal{R}}^{e_1}(f), \dots, \hat{\mathcal{R}}^{e_m}(f)$  on the  $m$  training domains, we must form an *estimated* risk distribution  $\hat{\mathbb{T}}_f$ . In general, given no prior knowledge about the form of  $\mathbb{T}_f$  (e.g. Gaussian), we use *kernel density estimation* (KDE, [49, 64]) with Gaussian kernels and either the Gaussian-optimal rule [65] or Silverman’s rule-of-thumb [65] for bandwidth selection. Fig. 1c depicts the PDF and CDF for 10 training risks when using Silverman’s rule-of-thumb. Armed

<sup>3</sup> $\mathbb{T}_f$  can be formally defined as the push-forward measure of  $\mathbb{Q}$  through the risk functional  $\mathcal{R}^e(f)$ ; see App. B.

<sup>4</sup>In financial optimization, when concerned with a distribution over potential losses, the  $\alpha$ -quantile value is known as the *value at risk* (VaR) at level  $\alpha$  [62].

with a predictor’s estimated risk distribution  $\hat{\mathbb{T}}_f$ , we can approximately solve (QRM) using the following empirical analogue:

$$\min_{f \in \mathcal{F}} F_{\hat{\mathbb{T}}_f}^{-1}(\alpha) \quad (4.1)$$

Note that (4.1) depends only on known quantities so we can compute and minimize it in practice, as detailed in Alg. 1 of Appendix E.1.

**Smoothing permits risk extrapolation.** Fig. 2 compares the KDE-smoothed CDF (black) to the unsmoothed empirical CDF (gray). As shown, the latter places zero probability mass on risks greater than our largest training risk, thus implicitly assuming that test risks cannot be larger than training risks. In contrast, the KDE-smoothed CDF permits “risk extrapolation” [41] since its right tail extends beyond our largest training risk, with the estimated  $\alpha$ -quantile risk going to infinity as  $\alpha \rightarrow 1$  (when kernels have full support). Note that different bandwidth-selection methods encode different assumptions about right-tail heaviness and thus about projected OOD risk. In § 4.3, we discuss how, as  $\alpha \rightarrow 1$ , this KDE-smoothing allows EQRM to learn predictors with invariant risk over domains. In Appendix C, we discuss different bandwidth-selection methods for EQRM.

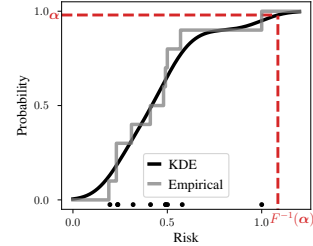


Figure 2: Risk CDFs.

## 4.2 Theory: Generalization bound

We now give a simplified version of our main generalization bound—Thm. D.1—which states that, given sufficiently many domains and samples, the empirical  $\alpha$ -quantile risk is a good estimate of the population  $\alpha$ -quantile risk. In contrast to previous results for DG, we bound the *proportion of test domains* for which a predictor performs well, rather than the average error [36, 42], and make no assumptions about the shift type, e.g. covariate shift [37]. The full version, stated and proved in Appendix D, provides specific finite-sample bounds on  $\epsilon_1$  and  $\epsilon_2$  below, depending on the hypothesis class  $\mathcal{F}$ , the empirical estimator  $F_{\hat{\mathbb{T}}_f}^{-1}(\alpha)$ , and the assumptions on the possible risk profiles of hypotheses  $f \in \mathcal{F}$ .

**Theorem 4.1** (Simplified form of Thm. D.1, uniform convergence). *Given  $m$  domains and  $n$  samples in each, there exist sequences  $\epsilon_1(n)$  and  $\epsilon_2(m)$ , with  $\epsilon_1(n) \rightarrow 0$  as  $n \rightarrow \infty$  and  $\epsilon_2(m) \rightarrow 0$  as  $m \rightarrow \infty$ , such that, with high probability over the training data:*

$$\sup_{f \in \mathcal{F}} \left| F_{\hat{\mathbb{T}}_f}^{-1}(\alpha - \epsilon_2) - F_{\hat{\mathbb{T}}_f}^{-1}(\alpha) \right| \leq \epsilon_1. \quad (4.2)$$

While many domains are required for this to bound be tight, i.e. for  $\alpha$  to *precisely* estimate the true quantile, our empirical results in § 6 demonstrate that EQRM performs well in practice given only a few domains. In such settings,  $\alpha$  still controls conservativeness, but with a less precise interpretation.

## 4.3 Theory: Causal recovery

We now prove that EQRM can recover the causal predictor in two parts. First, we show that, as  $\alpha \rightarrow 1$ , EQRM learns a predictor with minimal, invariant risk over domains. For Gaussian estimators of the risk distribution  $\hat{\mathbb{T}}_f$ , some intuition can be gained from Eq. (A.3) of Appendix A.2.1, noting that  $\alpha \rightarrow 1$  puts increasing weight on the sample standard deviation of risks over domains  $\hat{\sigma}_f$ , eventually forcing it to zero. For kernel density estimators, a similar intuition applies so long as the bandwidth has a certain dependence on  $\hat{\sigma}_f$ , as detailed in Appendix A.2.2. Second, we show that learning such a *minimal invariant-risk predictor* is sufficient to recover the causal predictor under weaker assumptions than prior work, namely Peters et al. [54] and Krueger et al. [41]. Together, these two parts provide the conditions under which EQRM successfully performs “causal recovery”, i.e., correctly recovers the true causal coefficients in a linear causal model of the data.

**Definition 4.2.** A predictor  $f$  is said to be an *invariant-risk predictor* if its risk is equal almost surely across domains (i.e.,  $\text{Var}_{e \sim \mathcal{Q}}[\mathcal{R}^e(f)] = 0$ ). A predictor is said to be a *minimal invariant-risk predictor* if it achieves the minimal possible risk across all possible invariant-risk predictors.

**Proposition 4.3** (EQRM learns a minimal invariant-risk predictor as  $\alpha \rightarrow 1$ , informal version of Props. A.4 and A.5). *Assume: (i)  $\mathcal{F}$  contains an invariant-risk predictor with finite training risks;*

and (ii) no arbitrarily-negative training risks. Then, as  $\alpha \rightarrow 1$ , Gaussian and kernel EQRM predictors (the latter with certain bandwidth-selection methods) converge to minimal invariant-risk predictors.

Props. A.4 and A.5 are stated and proved in Appendices A.2.1 and A.2.2 respectively. In addition, for the special case of Gaussian estimators of  $\mathbb{T}_f$ , Appendix A.2.1 relates our  $\alpha$  parameter to the  $\beta$  parameter of VREx [41, Eq. 8]. We next specify conditions under which learning such a minimal invariant-risk predictor is sufficient to recover the causal predictor.

**Theorem 4.4** (The causal predictor is the only minimal invariant-risk predictor). *Assume that: (i)  $Y$  is generated from a linear SEM,  $Y = \beta^\top X + N$ , with  $X$  observed and coefficients  $\beta \in \mathbb{R}^d$ ; (ii)  $\mathcal{F}$  is the class of linear predictors, indexed by  $\hat{\beta} \in \mathbb{R}^d$ ; (iii) the loss  $\ell$  is squared-error; (iv) the risk  $\mathbb{E}[(Y - \beta^\top X)^2]$  of the causal predictor  $\beta$  is invariant across domains; and (v) the system of equations*

$$\begin{aligned} 0 &\geq x^\top \text{Cov}_{X \sim e_1}(X, X)x + 2x^\top \text{Cov}_{N, X \sim e_1}(X, N) \\ &= \dots \\ &= x^\top \text{Cov}_{X \sim e_m}(X, X)x + 2x^\top \text{Cov}_{N, X \sim e_m}(X, N) \end{aligned} \quad (4.3)$$

has the unique solution  $x = 0$ . If  $\hat{\beta}$  is a minimal invariant-risk predictor, then  $\hat{\beta} = \beta$ .

**Assumptions (i–iii).** The assumptions that  $Y$  is drawn from a linear structural equation model (SEM) and that the loss is squared-error, while restrictive, are needed for all comparable causal recovery results [41, 54]. In fact, these assumptions are weaker than both Peters et al. [54, Thm. 2] (assume a linear Gaussian SEM for  $X$  and  $Y$ ) and Krueger et al. [41, Thm. 1] (assume a linear SEM for  $X$  and  $Y$ ).

**Assumption (iv).** The assumption that the risk of the causal predictor is invariant across domains, often called *domain homoskedasticity* [41], is necessary for any method inferring causality from the *invariance of risks* across domains. For methods based on the *invariance of functions*, namely the conditional mean  $\mathbb{E}[Y|\text{Pa}(Y)]$  [9, 66], this assumption is not required. Appendix G.1.2 compares methods based on invariant risks and to those based on invariant functions.

**Assumption (v).** In contrast to both Peters et al. and Krueger et al., we do not require specific types of interventions on the covariates. Instead, we require that a more general condition be satisfied, namely that the system of  $d$ -variate quadratic equations in (4.3) has a unique solution. Intuitively,  $\text{Cov}(X, X)$  captures how correlated the covariates are and ensures they are sufficiently uncorrelated to distinguish each of their influences on  $Y$ , while  $\text{Cov}(X, N)$  captures how correlated descendant covariates are with  $Y$  (via  $N$ ). Together, these terms capture the idea that *predicting  $Y$  from the causal covariates must result in the minimal invariant-risk*: the first inequality ensures the risk is *minimal* and the subsequent  $m - 1$  equalities that it is *invariant*. While this generality comes at the cost of abstraction, Appendix A.2.3 provides several concrete examples with different types of interventions to aid understanding and illustrate how this condition generalizes existing causal-recovery results based on invariant risks [41, 54]. Appendix A.2.3 also provides a proof of Thm. 4.4 and further discussion.

## 5 Related work

**Robust optimization in DG.** Throughout this paper, we follow an established line of work (see e.g., [9, 41, 51]) which formulates the DG problem through the lens of robust optimization [44]. To this end, various algorithms have been proposed for solving constrained [22] and distributionally robust [45] variants of the worst-case problem in (2.2). Indeed, this robust formulation has a firm foundation in the broader machine learning literature, with notable works in adversarial robustness [67–71] and fair learning [72, 73] employing similar formulations. Unlike these past works, we consider a robust but non-adversarial formulation for DG, where predictors are trained to generalize with high probability rather than in the worst case. Moreover, the majority of this literature—both within and outside of DG—relies on specific structural assumptions (e.g. covariate shift) on the types of possible interventions or perturbations. In contrast, we make the weaker and more flexible assumption of i.i.d.-sampled domains, which ultimately makes use of the observed domain-data to determine the types of shifts that are *probable*. We further discuss this important difference in § 7.

**Other approaches to DG.** Outside of robust optimization, many algorithms have been proposed for the DG setting which draw on insights from a diverse array of fields, including approaches based on tools from meta-learning [40, 43, 74–76], kernel methods [77, 78], and information theory [51]. Also prominent are works that design regularizers to generalize OOD [79–81] and works that seek



domain-invariant representations [82–84]. Many of these works employ hyperparameters which are difficult to interpret, which has no doubt contributed to the well-established model-selection problem in DG [38]. In contrast, in our framework,  $\alpha$  can be easily interpreted in terms of quantiles of the risk distribution. In addition, many of these works do not explicitly relate the training and test domains, meaning they lack theoretical results in the non-linear setting (e.g. [9, 41, 43, 85]). For those which do, they bound either average error over test domains [36, 42, 86] or worst-case error under specific shift types (e.g. covariate [22]). As argued above, the former lacks robustness while the latter can be both overly-conservative and difficult to justify in practice, where shift types are often unknown.

**High-probability generalization.** As noted in § 3, relaxing worst-case problems in favor of probabilistic ones has a long history in control theory [58, 59, 87–89], operations research [90], and smoothed analysis [60]. Recently, this paradigm has been applied to several areas of machine learning, including perturbation-based robustness [91, 92], fairness [93], active learning [94], and reinforcement learning [95, 96]. However, it has not yet been applied to domain generalization.

**Quantile minimization.** In financial optimization, the quantile and superquantile functions [62, 97, 98] are central to the literature surrounding portfolio risk management, with numerous applications spanning banking regulations and insurance policies [99, 100]. In statistical learning theory, several recent papers have derived uniform convergence guarantees in terms of alternative risk functionals besides expected risk [94, 101–103]. These results focus on functionals that can be written in terms of expectations over the loss distribution (e.g., the superquantile). In contrast, our uniform convergence guarantee (Theorem D.1) shows uniform convergence of the quantile function, which *cannot* be written as such an expectation; this necessitates stronger conditions to obtain uniform convergence, which ultimately suggest regularizing the estimated risk distribution (e.g. by kernel smoothing).

**Invariant prediction and causality.** Early work studied the problem of learning from multiple cause-effect datasets that share a functional mechanism but differ in noise distributions [39]. More generally, given (data from) multiple distributions, one can try to identify components which are stable, robust, or *invariant*, and find means to transfer them across problems [104–108]. As discussed in § 2, recent works have leveraged different forms of invariance across domains to discover causal relationships which, under the invariant mechanism assumption [57], generalize to new domains [9, 41, 54, 55, 109–111]. In particular, VREx [41] leveraged *invariant risks* (like EQRM) while IRM [9] leveraged *invariant functions* or coefficients—see Appendix G.1.2 for a detailed comparison of these approaches.

## 6 Experiments

We now evaluate our EQRM algorithm on synthetic datasets (§ 6.1), real-world datasets from WILDS (§ 6.2), and few-domain datasets from DomainBed (§ 6.3). Appendix G reports further results, while Appendix E reports further experimental details.

### 6.1 Synthetic datasets

**Linear regression.** We first consider a linear regression dataset based on the following linear SCM:

$$X_1 \leftarrow N_1, \quad Y \leftarrow X_1 + N_Y, \quad X_2 \leftarrow Y + N_2,$$

with  $N_j \sim \mathcal{N}(0, \sigma_j^2)$ . Here we have two features: one cause  $X_1 = X_{\text{cause}}$  and one effect  $X_2 = X_{\text{effect}}$  of  $Y$ . By fixing  $\sigma_1^2 = 1$  and  $\sigma_Y^2 = 2$  across domains but sampling  $\sigma_2 \sim \text{LogNormal}(0, 0.5)$ , we create a dataset in which  $X_2$  is more predictive of  $Y$  than  $X_1$  but less stable. Importantly, as we know the true distribution over domains  $Q(e) = \text{LogNormal}(\sigma_2^2; 0, 0.5)$ , we know the true risk quantiles. Fig. 3 depicts results for different  $\alpha$ 's with  $m = 1000$  domains and  $n = 200000$  samples in each, using the mean-squared-error (MSE) loss. Here we see that: **A**: for each true quantile (x-axis), the corresponding  $\alpha$  has the lowest risk (y-axis), confirming that the empirical  $\alpha$ -quantile risk is a good estimate of the population  $\alpha$ -quantile risk; **B**: As  $\alpha \rightarrow 1$ , the estimated risk distribution of  $f_\alpha$  approaches an invariant (or Dirac delta) distribution centered on the risk of the causal predictor; **C**: the regression coefficients approach those of the causal predictor as  $\alpha \rightarrow 1$ , trading predictive performance for robustness; and **D**: reducing the number of domains  $m$  reduces the accuracy of the estimated  $\alpha$ -quantile risks. In Appendix G.1, we additionally: (i) depict the risk CDFs corresponding to plot B above, and discuss how they depict the predictors' risk-robustness curves (G.1.1); and (ii) discuss the solutions of EQRM on datasets in which  $\sigma_1^2$ ,  $\sigma_2^2$  and/or  $\sigma_Y^2$  change over domains, compared to existing invariance-seeking algorithms like IRM [9] and VREx [41] (G.1.2).

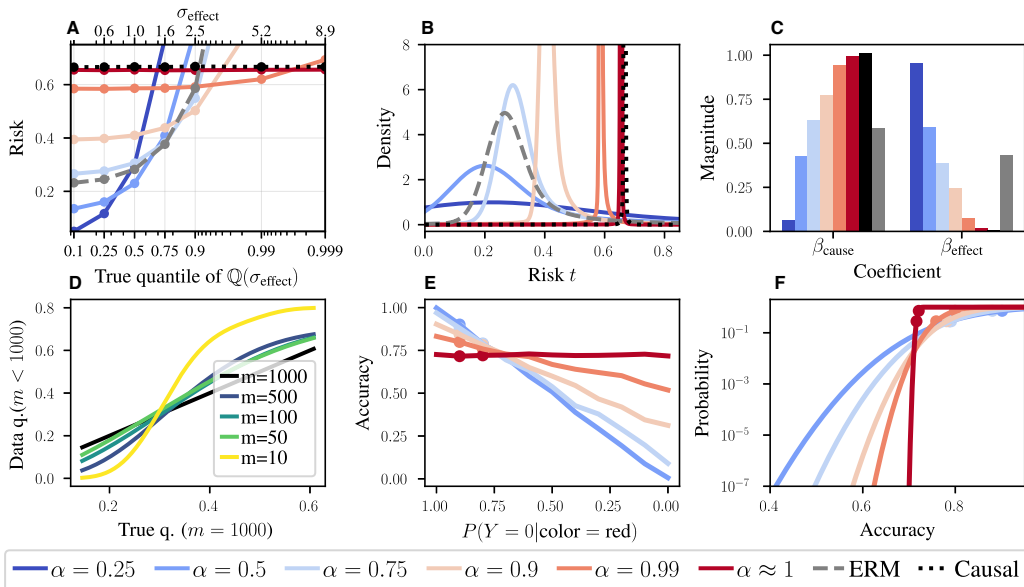


Figure 3: **EQRm on a toy linear regression dataset (A–D) and on ColoredMNIST (E–F).** **A:** Test risk at different quantiles or degrees of “OODness”. For each quantile (x-axis), the corresponding  $\alpha$  has the lowest risk (y-axis). **B:** Estimated risk distributions (corresponding CDFs in Appendix G.1.1). **C:** Regression coefficients approach those of the causal predictor ( $\beta_{\text{cause}} = 1, \beta_{\text{effect}} = 0$ ) as  $\alpha \rightarrow 1$ . **D:** Q-Q plot comparing the “true” risk quantiles (estimated with  $m = 1000$ ) against estimated ones ( $m < 1000$ ), for  $\alpha = 0.9$ . **E:** Performance of different  $\alpha$ ’s over increasingly OOD test domains, with dots showing training-domain accuracies. **F:** KDE-estimated accuracy-CDFs depicting accuracy-robustness curves. Larger  $\alpha$ ’s make lower accuracies less likely.

**ColoredMNIST.** We next consider the ColoredMNIST or CMNIST dataset [9]. Here, the MNIST dataset is used to construct a binary classification task (0–4 or 5–9) in which digit color (red or green) is a highly-informative but spurious feature. In particular, the two training domains are constructed such that red digits have an 80% and 90% chance of belonging to class 0, while the single test domain is constructed such that they only have a 10% chance. The goal is to learn an invariant predictor which uses only digit shape—a stable feature having a 75% chance of correctly determining the class in all 3 domains. We compare with IRM [9], GroupDRO [45], SD [112], IGA [113] and VREx [41] using: (i) random initialization (Xavier method [114]); and (ii) random initialization followed by several iterations of ERM. The ERM initialization or pretraining directly corresponds to the delicate penalty “annealing” or warm-up periods used by most penalty-based methods [9, 41, 112, 113]. For all methods, we use a 2-hidden-layer MLP with 390 hidden units, the Adam optimizer, a learning rate of 0.0001, and dropout with  $p = 0.2$ . We sweep over five penalty weights for the baselines and five  $\alpha$ ’s for EQRm. See Appendix E.2 for more experimental details. Table 1 shows that: (i) all methods struggle without ERM pretraining, explaining the need for penalty-annealing strategies in previous works and corroborating the results of [115, Table 1]; (ii) with ERM pretraining, EQRm matches or outperforms baseline methods, even approaching oracle performance (that of ERM trained on grayscale digits). These results suggest ERM pretraining as an effective strategy for DG methods.

In addition, Fig. 3 depicts the behavior of EQRm with different  $\alpha$ ’s. Here we see that: **E:** increasing  $\alpha$  leads to more consistent performance across domains, eventually forcing the model to ignore color and focus on shape for invariant-risk prediction; and **F:** a predictor’s (estimated) accuracy-CDF depicts its accuracy-robustness curve, just as its risk-CDF depicts its risk-robustness curve. Note that  $\alpha = 0.5$  gives the best worst-case (i.e. worst-domain) risk over the two training domains—the preferred solution of DRO [45]—while  $\alpha \rightarrow 1$  sacrifices risk for increased invariance or robustness.

## 6.2 Real-world datasets

We now evaluate our methods on the real-world or *in-the-wild* distribution shifts of WILDS [12]. We focus our evaluation on iWildCam [50] and OGB-MolPCBA [116, 117]—two large-scale classification datasets which have numerous test domains and thus facilitate a comparison of the test-domain risk distributions and their quantiles. Additional comparisons (e.g. using average accuracy) can be found in Appendix G.3. Our results demonstrate that, across two distinct data types (images and molecular graphs), EQRm offers superior tail or quantile performance.

Table 1: CMNIST test accuracy.

Algorithm	Initialization	
	Rand.	ERM
ERM	27.9 ± 1.5	27.9 ± 1.5
IRM	52.5 ± 2.4	69.7 ± 0.9
GrpDRO	27.3 ± 0.9	29.0 ± 1.1
SD	49.4 ± 1.5	70.3 ± 0.6
IGA	50.7 ± 1.4	57.7 ± 3.3
V-REx	55.2 ± 4.0	71.6 ± 0.5
EQRm	53.4 ± 1.7	71.4 ± 0.4
Oracle	72.1 ± 0.7	

Table 2: EQRm test risks on iWildCam.

Alg.	Mean risk	Quantile risk						
		0.0	0.25	0.50	0.75	0.90	0.99	1.0
ERM	1.31	0.015	0.42	0.76	2.25	2.73	4.99	5.25
IRM	1.53	0.098	0.52	1.24	1.86	2.36	6.95	7.46
GroupDRO	1.73	0.091	0.68	1.65	2.18	3.36	5.29	5.54
CORAL	1.27	0.024	0.45	0.73	2.12	2.66	4.50	4.98
EQRm <sub>0.25</sub>	2.03	0.024	0.46	2.70	3.01	3.48	5.03	5.26
EQRm <sub>0.50</sub>	1.11	<b>0.004</b>	0.24	0.68	1.71	2.15	4.04	4.11
EQRm <sub>0.75</sub>	1.05	0.009	<b>0.21</b>	0.68	1.50	2.35	4.88	5.45
EQRm <sub>0.90</sub>	<b>0.98</b>	0.047	0.28	<b>0.63</b>	<b>1.26</b>	<b>1.81</b>	4.11	4.48
EQRm <sub>0.99</sub>	0.99	0.12	0.35	0.64	1.30	2.00	<b>3.44</b>	<b>3.55</b>

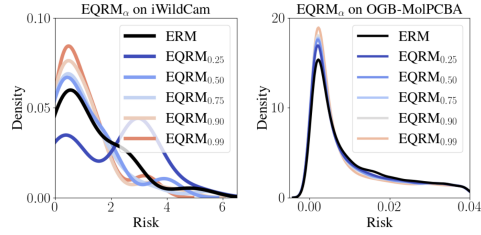


Figure 4: Test-domain risk distributions.

Table 3: EQRm test risks on OGB-MolPCBA.

Alg.	Mean risk	Quantile risk						
		0.0	0.25	0.50	0.75	0.90	0.99	1.0
ERM	<b>0.051</b>	0.0	0.004	0.017	0.060	0.13	0.49	16.04
IRM	0.073	0.098	0.52	1.24	1.86	2.36	6.95	7.46
GroupDRO	0.21	0.091	0.68	1.65	2.18	3.36	5.29	<b>5.54</b>
CORAL	0.055	0.0	0.12	0.32	1.23	2.01	5.76	7.44
EQRm <sub>0.25</sub>	0.054	0.0	0.003	0.016	0.059	0.13	0.48	15.46
EQRm <sub>0.50</sub>	0.052	0.0	0.003	0.015	0.059	0.13	0.48	11.33
EQRm <sub>0.75</sub>	0.052	0.0	0.003	0.015	0.059	0.13	0.47	12.15
EQRm <sub>0.90</sub>	0.052	0.0	0.003	0.015	0.059	0.12	0.47	10.81
EQRm <sub>0.99</sub>	0.053	0.0	0.003	<b>0.014</b>	<b>0.055</b>	<b>0.11</b>	<b>0.46</b>	7.16

**iWildCam.** We first consider the iWildCam image-classification dataset, which has 243 training domains and 48 test domains. Here, the label  $Y$  is one of 182 different animal species and the domain  $e$  is the camera trap which captured the image. In Table 2, we observe that  $EQRm_\alpha$  does indeed tend to optimize the  $\alpha$ -risk quantile, with larger  $\alpha$ s during training resulting in lower test-domain risks at the corresponding quantiles. In the left pane of Fig. 4, we plot the (KDE-smoothed) test-domain risk distribution for ERM and EQRm. Here we see a clear trend: as  $\alpha$  increases, the tails of the risk distribution tend to drop below ERM, which corroborates the superior quantile performance reported in Table 2. Note that, in Table 2, EQRm tends to record lower *average* risks than ERM. This has several plausible explanations. First, the number of testing domains (48) is relatively small, which could result in a biased sample with respect to the training domains. Second, the test domains may not represent i.i.d. draws from  $Q$ , as WILDS [12] test domains tend to be more challenging.

**OGB-MolPCBA.** We next consider the OGB-MolPCBA (or OGB) dataset, which is a molecular graph-classification benchmark containing 44,930 training domains and 43,793 test domains with an average of 3.6 samples per domain. Table 3 shows that ERM achieves the lowest *average* test risk on OGB, in contrast to the iWildCam results, while  $EQRm_\alpha$  still achieves stronger quantile performance. Of particular note is the fact that our methods significantly outperform ERM with respect to worst-case performance (columns/quantiles labeled 1.0); when  $QRM_\alpha$  is run with large values of  $\alpha$ , we reduce the worst-case risk by more than a factor of two. In Fig. 4, we again see that the risk distributions of  $EQRm_\alpha$  have lighter tails than that of ERM.

**A new evaluation protocol for DG.** The analysis provided in Tables 2-3 and Fig. 4 diverges from the standard evaluation protocol in DG [12, 38]. Rather than evaluating an algorithm’s performance *on average* across test domains, we seek to understand *the distribution of its performance*—particularly in the tails by means of the quantile function. This new evaluation protocol lays bare the importance of multiple test domains in DG benchmarks, allowing predictors’ risk distributions to be analyzed and compared. Indeed, as shown in Tables 2-3, solely reporting a predictor’s average or worst risk over test domains can be misleading when assessing its ability to generalize OOD, indicating that the performance of DG algorithms was likely never “lost”, as reported in [38], but rather invisible through the lens of average performance. This underscores the necessity of incorporating tail- or quantile-risk measures into a more holistic evaluation protocol for DG, ultimately providing a more nuanced and complete picture. In practice, which measure is preferred will depend on the application. For example, medical applications could have a human-specified robustness-level or quantile-of-interest.

### 6.3 DomainBed datasets

Finally, we consider the benchmark datasets of DomainBed [38], in particular VLCS [118], PACS [119], OfficeHome [120], TerraIncognita [5] and DomainNet [121]. As each of these datasets contain just 4 or 6 domains, it is not possible to meaningfully compare tail or quantile performance. Nonetheless, in line with much recent work, and to compare EQRm to a range of standard baselines on few-domain datasets, Table 4 reports DomainBed results in terms of the average performance



Table 4: DomainBed results. Model selection: training-domain validation set.

Algorithm	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet	Avg
ERM	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
IRM	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	61.6
GroupDRO	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	60.9
Mixup	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	63.4
MLDG	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	63.6
CORAL	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	<b>64.6</b>
ARM	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	61.7
VREx	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9	61.9
EQRm	77.8 ± 0.6	86.5 ± 0.2	67.5 ± 0.1	47.8 ± 0.6	41.0 ± 0.3	64.1

across each choice of test domain. While EQRM outperforms most baselines, including ERM, we reiterate that comparing algorithms solely in terms of average performance can be misleading (see final paragraph of § 6.2). Full implementation details are given in Appendix E.3, with further results in Appendix G.2 (additional baselines, per-dataset results, and test-domain model selection).

## 7 Discussion

**Interpretable model selection.**  $\alpha$  approximates the probability with which our predictor will generalize with risk below the associated  $\alpha$ -quantile value. Thus,  $\alpha$  represents an interpretable parameterization of the risk-robustness trade-off. Such interpretability is critical for model selection in DG, and for practitioners with application-specific requirements on performance and/or robustness.

**The assumption of i.i.d. domains.** For  $\alpha$  to approximate the probability of generalizing, training and test domains must be i.i.d.-sampled. While this is rarely true in practice—e.g. hospitals have shared funders, service providers, etc.—we can better satisfy this assumption by subscribing to a new data collection process in which we collect training-domain data which is representative of how the underlying system tends to change. For example: (i) randomly select 100 US hospitals; (ii) gather and label data from these hospitals; (iii) train our system with the desired  $\alpha$ ; (iv) deploy our system to all US hospitals, where it will be successful with probability  $\approx \alpha$ . While this process may seem expensive, time-consuming and vulnerable (e.g. to new hospitals), it offers a promising path to machine learning systems which *generalize with high probability*. Moreover, it is worth noting the alternative: prior works achieve generalization by assuming that only particular types of shifts can occur, e.g. covariate shifts [22, 122, 123], label shifts [123, 124], concept shifts [125], measurement shifts [19], mean shifts [126], shifts which leave the mechanism of  $Y$  invariant [9, 39, 41, 54], etc. In real-world settings, where the underlying shift mechanisms are often unknown, such assumptions are both difficult to justify and impossible to test. Future work could look to relax the i.i.d.-domains assumption by leveraging knowledge of domain dependencies (e.g. time).

**The wider value of risk distributions.** As demonstrated in § 6, a predictor’s risk distribution has value beyond quantile-minimization—it estimates the probability associated with each level of risk. Thus, regardless of the algorithm used, risk distributions can be used to analyze trained predictors.

## 8 Conclusion

We have presented Quantile Risk Minimization for achieving *Probable* Domain Generalization, i.e., learning predictors that perform well *with high probability* rather than *on-average* or *in the worst case*. After explicitly relating training and test domains as draws from the same underlying meta-distribution, we proposed to learn predictors with minimal  $\alpha$ -quantile risk. We then introduced the EQRM algorithm, for which we proved a generalization bound and recovery of the causal predictor as  $\alpha \rightarrow 1$ . Finally, in our experiments, we introduced a more holistic quantile-focused evaluation protocol for DG, and demonstrated that EQRM outperforms state-of-the-art baselines on several DG benchmarks.

## Acknowledgments and Disclosure of Funding

We thank Chris Williams, Ian Mason and Krikamol Muandet for providing feedback on an earlier draft, as well as Lars Lorch, David Krueger, Francesco Locatello and members of the MPI Tübingen causality group for helpful discussions and comments. We also thank Minsu Kim for catching an error in an earlier proof of Theorem D.1. This work was supported by the German Federal Ministry

of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A, 01IS18039B; and by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

## References

- [1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015. [1](#)
- [2] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, 2016.
- [3] 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. *Nature*, 596(7873):583–589, 2021. [1](#)
- [4] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011. [1](#)
- [5] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, pages 456–473, 2018. [9](#)
- [6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.
- [7] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020. [1](#)
- [8] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Medicine*, 15(11), 2018. [1](#)
- [9] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 2019. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#), [10](#), [20](#), [23](#), [36](#), [38](#), [39](#), [40](#)
- [10] Timothy Niven and Hung Yu Kao. Probing neural network comprehension of natural language arguments. In *Association for Computational Linguistics*, pages 4658–4664, 2020. [1](#)
- [11] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv:2008.04859*, 2020. [1](#)
- [12] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, 2021. [2](#), [8](#), [9](#), [36](#), [45](#)
- [13] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *World Wide Web Conference*, pages 491–500, 2019. [1](#)
- [14] Matthew C Hansen, Peter V Potapov, Rebecca Moore, Matt Hancher, Svetlana A Turubanova, Alexandra Tyukavina, David Thau, Stephen V Stehman, Scott J Goetz, Thomas R Loveland, et al. High-resolution global maps of 21st-century forest cover change. *Science*, 342(6160): 850–853, 2013. [1](#)
- [15] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Computer Vision and Pattern Recognition*, pages 6172–6180, 2018.

- [16] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? In *International Conference on Computer Vision*, pages 9661–9669, 2021. 1
- [17] Samil Karahan, Merve Kilinc Yildirim, Kadir Kirtac, Ferhat Sukru Rende, Gultekin Butun, and Hazim Kemal Ekenel. How image degradations affect deep cnn-based face recognition? In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5, 2016. 1
- [18] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20:1–25, 2019.
- [19] Cian Eastwood, Ian Mason, Christopher K. I. Williams, and Bernhard Schölkopf. Source-free adaptation to measurement shift via bottom-up feature restoration. In *International Conference on Learning Representations*, 2021. 4, 10
- [20] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.
- [21] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.
- [22] Alexander Robey, George J. Pappas, and Hamed Hassani. Model-based domain generalization. In *Advances in Neural Information Processing Systems*, 2021. 3, 4, 6, 7, 10, 28
- [23] Allan Zhou, Fahim Tajwar, Alexander Robey, Tom Knowles, George J Pappas, Hamed Hassani, and Chelsea Finn. Do deep networks transfer invariances across classes? *arXiv preprint arXiv:2203.09739*, 2022. 1
- [24] Jorge Jovicich, Silvester Czanner, Xiao Han, David Salat, Andre van der Kouwe, Brian Quinn, Jenni Pacheco, Marilyn Albert, Ronald Killiany, Deborah Blacker, et al. MRI-derived measurements of human subcortical, ventricular and intracranial brain volumes: reliability effects of scan sessions, acquisition sequences, data analyses, scanner upgrade, scanner vendors and field strengths. *Neuroimage*, 46(1):177–192, 2009. 1
- [25] Ehab A AlBadawy, Ashirbani Saha, and Maciej A Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical Physics*, 45(3):1150–1158, 2018.
- [26] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101–544, 2019.
- [27] Emma Beede, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruanviboonsuk, and Laura M. Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–12. Association for Computing Machinery, 2020.
- [28] Christian Wachinger, Anna Rieckmann, Sebastian Pölsterl, Alzheimer’s Disease Neuroimaging Initiative, et al. Detect and correct bias in multi-site neuroimaging datasets. *Medical Image Analysis*, 67:101879, 2021. 1
- [29] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *International Conference on Intelligent Transportation Systems*, pages 3819–3824, 2018. 1
- [30] Georg Volk, Stefan Müller, Alexander von Bernuth, Dennis Hospach, and Oliver Bringmann. Towards robust CNN-based object detection through augmentation with synthetic rain variations. In *International Conference on Intelligent Transportation Systems*, pages 285–292, 2019.

- [31] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *Machine Learning for Autonomous Driving Workshop, NeurIPS 2019*, 2019. 1
- [32] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016. 1
- [33] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.
- [34] Gustav Mårtensson, Daniel Ferreira, Tobias Granberg, Lena Cavallin, Ketil Oppedal, Alessandro Padovani, Irena Rektorova, Laura Bonanni, Matteo Pardini, Milica G Kramberger, et al. The reliability of a deep learning model in clinical out-of-distribution MRI data: a multicohort study. *Medical Image Analysis*, 66:101714, 2020.
- [35] Daniel C Castro, Ian Walker, and Ben Glocker. Causality matters in medical imaging. *Nature Communications*, 11(1):1–10, 2020. 1
- [36] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems*, volume 24, 2011. 1, 3, 5, 7
- [37] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013. 3, 5
- [38] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020. 1, 2, 7, 9, 28, 36
- [39] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris M Mooij. On causal and anticausal learning. In *ICML*, 2012. 7, 10
- [40] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, pages 3490–3497, 2018. 6
- [41] 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*, volume 139, pages 5815–5826, 2021. 1, 2, 3, 5, 6, 7, 8, 10, 20, 21, 23, 36, 38, 39, 40
- [42] Gilles Blanchard, Aniket Anand Deshmukh, Ürun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1):46–100, 2021. 1, 5, 7
- [43] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678, 2021. 1, 6, 7
- [44] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009. 1, 3, 6
- [45] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019. 1, 3, 4, 6, 8, 37, 38
- [46] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2021. 1, 3
- [47] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. 1

- [48] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Adversarial training can hurt generalization. In *ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019. 1
- [49] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. 2, 4
- [50] Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iWildCam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021. 2, 8
- [51] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 6
- [52] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010. 3
- [53] Rune Christiansen, Niklas Pfister, Martin Emil Jakobsen, Nicola Gnecco, and Jonas Peters. A causal framework for distribution generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3, 20
- [54] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012, 2016. 3, 5, 6, 7, 10, 20, 23, 24, 25, 39
- [55] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018. 3, 7, 20
- [56] Judea Pearl. *Causality*. Cambridge University Press, 2009. 3, 20
- [57] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017. 3, 7
- [58] Marco C Campi and Simone Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008. 3, 7
- [59] Federico Alessandro Ramponi. Consistency of the scenario approach. *SIAM Journal on Optimization*, 28(1):135–162, 2018. 3, 7
- [60] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004. 3, 7
- [61] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 1999. 4
- [62] Darrell Duffie and Jun Pan. An overview of value at risk. *Journal of derivatives*, 4(3):7–49, 1997. 4, 7
- [63] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998. 4
- [64] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837, 1956. 4
- [65] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26. CRC Press, 1986. 4, 22, 29, 36
- [66] Mingzhang Yin, Yixin Wang, and David M Blei. Optimization-based causal estimation from heterogenous environments. *arXiv preprint arXiv:2109.11990*, 2021. 6, 23, 25
- [67] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 6



- [68] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- [69] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.
- [70] Alexander Robey, Luiz Chamon, George J Pappas, Hamed Hassani, and Alejandro Ribeiro. Adversarial robustness with semi-infinite constrained learning. *Advances in Neural Information Processing Systems*, 34:6198–6215, 2021.
- [71] Jia-Jie Zhu, Christina Kouridi, Yassine Nemmour, and Bernhard Schölkopf. Adversarially robust kernel smoothing. *arXiv preprint arXiv:2102.08474*, 2021. 6
- [72] Natalia L Martinez, Martin A Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. Blind pareto fairness and subgroup robustness. In *International Conference on Machine Learning*, pages 7492–7501. PMLR, 2021. 6
- [73] Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. Minimax group fairness: Algorithms and experiments. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 66–76, 2021. 6
- [74] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018. 6
- [75] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.
- [76] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9624–9633, 2021. 6
- [77] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349, 2021. 6
- [78] Aniket Anand Deshmukh, Yunwen Lei, Srinagesh Sharma, Urun Dogan, James W Cutler, and Clayton Scott. A generalization error bound for multi-class domain generalization. *arXiv preprint arXiv:1905.10392*, 2019. 6
- [79] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107, 2020. 6
- [80] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in Neural Information Processing Systems*, 33:3118–3129, 2020.
- [81] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021. 6
- [82] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 7
- [83] 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)*, pages 624–639, 2018.
- [84] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pages 124–140. Springer, 2020. 7

- [85] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. 7
- [86] Vikas Garg, Adam Tauman Kalai, Katrina Ligett, and Steven Wu. Learn to expect the unexpected: Probably approximately correct domain generalization. In *International Conference on Artificial Intelligence and Statistics*, pages 3574–3582. PMLR, 2021. 7
- [87] Roberto Tempo, Giuseppe Calafiore, and Fabrizio Dabbene. *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer, 2013. 7
- [88] Lars Lindemann, Nikolai Matni, and George J Pappas. Stl robustness risk over discrete-time stochastic processes. *arXiv preprint arXiv:2104.01503*, 2021.
- [89] Lars Lindemann, Alena Rodionova, and George J. Pappas. Temporal robustness of stochastic signals. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2022. 7
- [90] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021. 7, 37
- [91] Alexander Robey, Luiz FO Chamon, George J Pappas, and Hamed Hassani. Probabilistically robust learning: Balancing average- and worst-case performance. *arXiv preprint arXiv:2202.01136*, 2022. 7
- [92] Leslie Rice, Anna Bair, Huan Zhang, and J Zico Kolter. Robustness between the worst and average case. *Advances in Neural Information Processing Systems*, 34, 2021. 7
- [93] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. *arXiv preprint arXiv:2007.01162*, 2020. 7
- [94] Sebastian Curi, Kfir Y Levy, Stefanie Jegelka, and Andreas Krause. Adaptive sampling for stochastic risk-averse learning. *Advances in Neural Information Processing Systems*, 33: 1036–1047, 2020. 7, 37
- [95] Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Safe policies for reinforcement learning via primal-dual methods. *IEEE Transactions on Automatic Control*, 2022. 7
- [96] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017. 7
- [97] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000. 7, 37
- [98] Pavlo Krokmal, Jonas Palmquist, and Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of risk*, 4:43–68, 2002. 7
- [99] David Wozabal. Value-at-risk optimization using the difference of convex algorithm. *OR spectrum*, 34(4):861–883, 2012. 7
- [100] Philippe Jorion. *Value at risk: the new benchmark for controlling market risk*. Irwin Professional Pub., 1997. 7
- [101] Jaeho Lee, Sejun Park, and Jinwoo Shin. Learning bounds for risk-sensitive learning. *Advances in Neural Information Processing Systems*, 33:13867–13879, 2020. 7
- [102] Justin Khim, Liu Leqi, Adarsh Prasad, and Pradeep Ravikumar. Uniform convergence of rank-weighted learning. In *International Conference on Machine Learning*, pages 5254–5263. PMLR, 2020.
- [103] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021. 7

- [104] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013. 7
- [105] E. Bareinboim and J. Pearl. Transportability from multiple environments with limited experiments: Completeness results. In *Advances in Neural Information Processing Systems 27*, pages 280–288, 2014.
- [106] Kun Zhang, Mingming Gong, and Bernhard Schölkopf. Multi-source domain adaptation: A causal view. In *Twenty-ninth AAAI Conference on Artificial Intelligence*, 2015.
- [107] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, pages 2839–2848. PMLR, 2016.
- [108] B. Huang, K. Zhang, J. Zhang, R. Sanchez-Romero, C. Glymour, and B. Schölkopf. Behind distribution shift: Mining driving forces of changes and causal arrows. In *IEEE 17th International Conference on Data Mining (ICDM 2017)*, pages 913–918, 2017. 7
- [109] Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018. 7
- [110] Niklas Pfister, Peter Bühlmann, and Jonas Peters. Invariant causal prediction for sequential data. *Journal of the American Statistical Association*, 114(527):1264–1276, 2019.
- [111] Juan L Gamella and Christina Heinze-Deml. Active invariant causal prediction: Experiment selection through stability. *Advances in Neural Information Processing Systems*, 33:15464–15475, 2020. 7
- [112] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021. 8
- [113] Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. <https://openreview.net/forum?id=FzGiUKN4aBp>, 2020. 8
- [114] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. PMLR, 2010. 8
- [115] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. *arXiv preprint arXiv:2203.15516*, 2022. 8, 36
- [116] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 33:22118–22133, 2020. 8
- [117] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. 8
- [118] Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 9
- [119] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 9
- [120] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 9
- [121] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 9



- [122] Joaquin Quiñero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. MIT Press, 2008. 10, 28
- [123] Amos J Storkey. When training and test sets are different: characterising learning transfer. In *Dataset Shift in Machine Learning*, pages 3–28. MIT Press, 2009. 10
- [124] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, pages 3122–3130, 2018. 10
- [125] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45:521–530, 2012. 10
- [126] Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):215–246, 2021. 10
- [127] Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pages 322–348. PMLR, 2020. 28
- [128] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588, 1997. 29
- [129] Ichiro Takeuchi, Quoc V. Le, Timothy D. Sears, and Alexander J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7(45):1231–1264, 2006.
- [130] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. *Advances in Neural Information Processing Systems*, 22, 2009. 29
- [131] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer, 2003. 30
- [132] Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The annals of Probability*, pages 1269–1283, 1990. 30
- [133] Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer, 2004. 32
- [134] Ronald A DeVore and George G Lorentz. *Constructive approximation*, volume 303. Springer Science & Business Media, 1993. 33
- [135] Keith Ball et al. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31(1-58):26, 1997. 33
- [136] JM Blair, CA Edwards, and J Howard Johnson. Rational Chebyshev approximations for the inverse of the error function. *Mathematics of Computation*, 30(136):827–830, 1976. 36
- [137] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 36
- [138] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 36
- [139] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 36
- [140] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017. 36
- [141] Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning. *arXiv preprint arXiv:2102.05371*, 2021. 37
- [142] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. 38

### 4.3 Comments on the paper

**The need for new DG benchmarks with multiple test domains.** The analyses provided in Tables 2-3 and Fig. 4 diverge from the standard evaluation protocol in DG (Gulrajani and Lopez-Paz, 2020; Koh et al., 2021). Rather than evaluating an algorithm’s performance on average across test domains, we seek to understand *the distribution of its performance*—particularly in the tails by means of the quantile function. This new evaluation protocol lays bare the importance of multiple test domains in DG benchmarks, allowing predictors’ performance distributions to be analysed and compared. Indeed, as shown in Tables 2-3, solely reporting a predictor’s average or worst risk over test domains can be misleading when assessing its ability to generalize OOD, indicating that the performance of DG algorithms was likely never “lost”, as reported by Gulrajani and Lopez-Paz (2020), but rather invisible through the lens of average performance. To facilitate this new, more holistic evaluation protocol for DG, new benchmark datasets are needed containing multiple test domains, ultimately allowing performance distributions to be computed and compared. While some datasets with multiple test domains exist, e.g., the iWildCam camera-trap dataset and OGB-MolPCBA molecular dataset, the community would benefit from a concerted effort to collect these datasets into an easy-to-use benchmark suite with more insightful performance measures.

**Probabilistic robustness across *and within* domains.** We sought predictors that are probably robust across domains, e.g., across hospitals. To do so, we optimized the quantile performance across domains, where the performance of each domain was an average over samples in that domain (i.e., the expected risk). However, one may also seek robustness *within* a domain, e.g., across patients within each hospital. To do so, one could characterise the domain performance itself using a particular quantile performance across samples within that domain, leading to a nested QRM problem.

**Can we do better than a fixed performance-robustness trade-off?** We choose a fixed trade-off between performance and robustness at training time via the interpretable probability-of-generalization parameter  $\alpha$ . In effect, this trade-off determines how much we use features that are informative but unreliable, eventually discarding all such features as  $\alpha \rightarrow 1$ . However, one may ask whether it’s possible to change how we use these features at test time, rather than discarding them at training time. Recent

works have provided some empirical evidence in support of this hypothesis, showing that simply retraining the last layer of an ERM-trained model can outperform more robust feature-learning methods on spurious correlation benchmarks ([Kirichenko et al., 2022](#); [Rosenfeld et al., 2022](#)). However, these works require labelled test-domain data which is generally not available. In the next chapter, we thus explore how this can be done without labels, using only the predictions of a robust model.

# 5

## Domain Generalisation: Harnessing Spurious Features

This chapter also focuses on *domain generalisation* (DG, [Blanchard et al. 2011](#); [Muan-det et al. 2013](#)) where models are trained on data from multiple related environments or domains (e.g., hospitals) with the goal of performing well on data from unseen test domains. In general, preparation involves exploiting invariances across the training domains in the hope that they hold in test domains. In particular, many prior works sought robustness by discarding “spurious” or *unstable* features whose relationship with the label changes across domains, restricting the model to features with an invariant or *stable* relationship with the label across domains ([Arjovsky et al., 2019](#); [Krueger et al., 2021](#); [Peters et al., 2016](#)). However, these unstable features often carry *complementary* information about the label that could boost performance if used correctly in the test domain. Thus, perhaps we don’t need to discard these features at all but rather *use them in the right way*.

Our main contribution is to show that it is possible to learn how to use these unstable features in the test domain *without labels*. In particular, we prove that predictions based on stable features provide sufficient guidance for doing so, provided that stable and unstable features are conditionally independent given the label.

Based on this theoretical insight, we propose the Stable Feature Boosting (SFB) algorithm for: (i) learning a predictor that separates stable and conditionally-independent unstable features on the training domains; and (ii) using the stable-feature predictions to adapt the unstable-feature predictions in the test domain. Theoretically, we prove that SFB can learn an asymptotically-optimal predictor in the test domain without labels. Empirically, we demonstrate the effectiveness of SFB on real and synthetic datasets.

## **5.1 Contribution**

I led this project from conceptualisation to final form. In particular, I was heavily involved in coming up with the initial idea, formalising the learning objective and algorithm, designing the experimental analyses, running the experimental analyses, and writing the manuscript. Shashank Singh, with whom I share first authorship, led the theory of Section 4 and helped with some of the above tasks.

## **5.2 Paper**

---

# Spuriousity Didn't Kill the Classifier: Using Invariant Predictions to Harness Spurious Features

---

Cian Eastwood<sup>\*1,2</sup>    Shashank Singh<sup>\*1</sup>  
 Andrei L. Nicolicioiu<sup>1</sup>    Marin Vlastelica<sup>1</sup>    Julius von Kügelgen<sup>1,3</sup>    Bernhard Schölkopf<sup>1</sup>

<sup>1</sup> Max Planck Institute for Intelligent Systems, Tübingen

<sup>2</sup> University of Edinburgh    <sup>3</sup> University of Cambridge

## Abstract

To avoid failures on out-of-distribution data, recent works have sought to extract features that have an invariant or *stable* relationship with the label across domains, discarding “spurious” or *unstable* features whose relationship with the label changes across domains. However, unstable features often carry *complementary* information that could boost performance if used correctly in the test domain. In this work, we show how this can be done *without test-domain labels*. In particular, we prove that pseudo-labels based on stable features provide sufficient guidance for doing so, provided that stable and unstable features are conditionally independent given the label. Based on this theoretical insight, we propose Stable Feature Boosting (SFB), an algorithm for: (i) learning a predictor that separates stable and conditionally-independent unstable features; and (ii) using the stable-feature predictions to adapt the unstable-feature predictions in the test domain. Theoretically, we prove that SFB can learn an asymptotically-optimal predictor without test-domain labels. Empirically, we demonstrate the effectiveness of SFB on real and synthetic data.

## 1 Introduction

Machine learning systems can be sensitive to distribution shift [26]. Often, this sensitivity is due to a reliance on “spurious” features whose relationship with the label changes across domains, ultimately leading to degraded performance in the test domain of interest [21]. To avoid this pitfall, recent works on domain or out-of-distribution (OOD) generalization have sought predictors which only make use of features that have a *stable* or invariant relationship with the label across domains, discarding the spurious or *unstable* features [45, 1, 35, 15]. However, despite their instability, spurious features can often provide additional or *complementary* information about the target label. Thus, if a predictor could be adjusted to use spurious features optimally in the test domain, it would boost performance substantially. That is, perhaps we don't need to discard spurious features at all but rather *use them in the right way*.

As a simple but illustrative example, consider the CoLoRmNIST or CMNIST dataset [1]. This transforms the original MNIST dataset into a binary classification task (digit in 0–4 or 5–9) and then: (i) flips the label with probability 0.25, meaning that, across all 3 domains, digit shape correctly determines the label with probability 0.75; and (ii) colorizes the digit such that digit color (red or green) is a more informative but spurious feature (see Fig. 1a). Prior work focused on learning an invariant predictor that uses only shape and avoids using color—a spurious feature whose relationship with the label changes across domains. However, as shown in Fig. 1b, the invariant predictor is suboptimal in test domains where color can be used in a domain-specific manner to improve performance. We thus ask: when and how can such informative but spurious features be safely harnessed *without labels*?

---

\*Equal contribution. Correspondence to

or shashankssingh44@gmail.com.

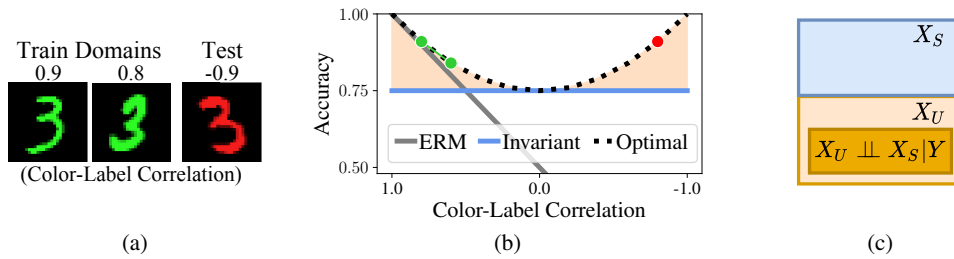


Figure 1: **Invariant (stable) and spurious (unstable) features.** (a) Illustrative images from CMNIST [1]. (b) CMNIST accuracies (y-axis) over test domains of decreasing color-label correlation (x-axis). The ‘Oracle’ uses both invariant (shape) *and* spurious (color) features optimally in the test domain, boosting performance over an invariant model (orange region). We show how this can be done *without test-domain labels*. (c) Generally, invariant models use only the *stable* component  $X_S$  of  $X$ , discarding the spurious or *unstable* component  $X_U$ . We prove that predictions based on  $X_S$  can be used to safely harness a sub-component of  $X_U$  (dark-orange region).

Our main contribution lies in answering this question, showing when and how it is possible to safely harness spurious or *unstable* features without test-domain labels. In particular, we prove that predictions based on stable features provide sufficient guidance for doing so, provided that stable and unstable features are conditionally independent given the label (see Fig. 1c).

**Structure and contributions.** The rest of this paper is organized as follows. We first discuss related work in § 2, providing context and high-level motivation for our approach. In § 3, we then explain how stable and unstable features can be extracted, how unstable features can be harnessed *with* test-domain labels, and the questions/challenges that arise when trying to harness unstable features *without* test-domain labels. In § 4, we present our main theoretical contributions which provide precise answers to these questions, before using these insights to propose the Stable Feature Boosting (SFB) algorithm in § 5. In § 6, we present our experimental results, before ending with a discussion and concluding remarks in § 7. Our contributions can be summarized as follows:

- **Algorithmic:** We propose Stable Feature Boosting (SFB), the first algorithm for using invariant predictions to safely harness spurious features *without test-domain labels*.
- **Theoretical:** SFB is grounded in a novel theoretical result (Thm 4.4) giving sufficient conditions for provable test-domain adaptation without labels. Under these conditions, Thm 4.6 shows that, given enough unlabeled data, SFB learns the Bayes-optimal adapted classifier in the test domain.
- **Experimental:** Our experiments on synthetic and real-world data demonstrate the effectiveness of SFB—even in scenarios where it is unclear if its assumptions are fully satisfied.

## 2 Related Work

**Domain generalization, robustness and invariant prediction.** A fundamental starting point for work in domain generalization is the observation that certain “stable” features, often direct causes of the label, may have an invariant relationship with the label across domains [45, 1, 67, 55, 40, 78, 14]. However, such stable or invariant predictors often discard highly informative but unstable information. Rothenhäusler et al. [51] show that we may need to trade off stability and predictiveness, while Eastwood et al. [15] seek such a trade-off via an interpretable probability-of-generalization parameter. The current work is motivated by the idea that one might avoid such a trade-off by changing how unstable features are used at test time, rather than discarding them at training time.

**Test-domain adaptation without labels (unsupervised domain adaptation).** In the source-free and test-time domain adaptation literature, it is common to adapt to new domains using a model’s own pseudo-labels [20, 36, 39, 71, 30]—see Rusak et al. [52] for a recent review. In contrast, we: (i) use one (stable) model to provide reliable/robust pseudo-labels and another (unstable) model to adapt domain-specific information; and (ii) propose a bias correction step that provably ensures an accurate, well-calibrated unstable model ( $\Pr[Y|X_U]$ ) as well as an optimal joint/combined model ( $\Pr[Y|X_S, X_U]$ ). Beyond this literature, Bui et al. [12] propose a meta-learning approach for exploiting unstable/domain-specific features. However, they use unstable features *in the same way* in the test domain, which, by definition, is not robust and can degrade performance. Sun et al. [63] share the goal of exploiting unstable features to go “beyond invariance”. However, in contrast to our approach, they require labels for the unstable features (rarely available) and only address label shifts.

Table 1: **Comparison with related work.** \*QRM [15] uses an interpretable hyperparameter  $\alpha \in [0, 1]$  to balance the probability of robust generalization and using more information from  $X$ .

Method	Components of $X$ Used			Robust	No test-domain labels
	Stable	Complementary	All		
ERM [65]	✓	✓	✓	✗	✓
IRM [1]	✓	✗	✗	✓	✓
QRM [15]	✓	✓*	✓*	✓*	✓
DARE [50]	✓	✓	✓	✓	✗
ACTIR [31]	✓	✓	✗	✓	✗
SFB (Ours)	✓	✓	✗	✓	✓

**Test-domain adaptation with labels (few-shot fine-tuning).** Fine-tuning part of a model using a small number of labeled test-domain examples is a common way to deal with distribution shift [16, 17, 13]. More recently, it has been shown that simply retraining the last layer of an ERM-trained model outperforms more robust feature-learning methods on spurious correlation benchmarks [50, 32, 74]. Similar to our approach, Jiang and Veitch [31] separate stable and conditionally-independent unstable features and then adapt their use of the latter in the test domain. However, in contrast to our approach, theirs requires test-domain labels. In addition, they assume data is drawn from an anti-causal generative model, which is strictly stronger than our “complementarity” assumption (see § 4).

Table 1 summarizes related work while App. H discusses further related work.

### 3 Problem Setup: Extracting and Harnessing Unstable Features

**Setup.** We consider the problem of domain generalization (DG) [8, 42, 24] where predictors are trained on data from multiple training domains and with the goal of performing well on data from unseen test domains. More formally, we consider datasets  $D^e = \{(X_i^e, Y_i^e)\}_{i=1}^{n_e}$  collected from  $m$  different training domains or *environments*  $\mathcal{E}_{\text{tr}} := \{E_1, \dots, E_m\}$ , with each dataset  $D^e$  containing data pairs  $(X_i^e, Y_i^e)$  sampled i.i.d. from  $\mathbb{P}(X^e, Y^e)$ .<sup>2</sup> The goal is then to learn a predictor  $f(X)$  that performs well on a larger set  $\mathcal{E}_{\text{all}} \supset \mathcal{E}_{\text{tr}}$  of possible domains.

**Average performance: use all features.** The first approaches to DG sought predictors that perform well *on average* over domains [8, 42] using empirical risk minimization (ERM, [66]). However, predictors that perform well on average can lack robustness [43, 49], potentially performing quite poorly on large subsets of  $\mathcal{E}_{\text{all}}$ . In particular, minimizing the average error leads predictors to make use of any features that are informative about the label (on average), including “spurious” or “shortcut” [21] features whose relationship with the label is subject to change across domains. In test domains where these feature-label relationships change in new or more severe ways than observed during training, this usually leads to significant performance drops or even complete failure [73, 4].

**Worst-case or robust performance: use only stable features.** To improve robustness, subsequent works sought predictors that only use *stable or invariant* features, i.e., those that have a stable or invariant relationship with the label across domains [45, 1, 47, 70, 58]. For example, Arjovsky et al. [1] do so by enforcing that the classifier on top of these features is optimal for all domains simultaneously. We henceforth use *stable features* and  $X_S$  to refer to these features, and stable predictors to refer to predictors which use only these features. Analogously, we use *unstable features* and  $X_U$  to refer to features with an unstable or “spurious” relationship with the label across domains. Note that  $X_S$  and  $X_U$  partition the components of  $X$  which are informative about  $Y$ , as depicted in Fig. 1c, and that formal definitions of  $X_S$  and  $X_U$  are provided in § 4.

#### 3.1 Harnessing unstable features with labels

A stable predictor  $f_S$  is unlikely to be the best predictor in any given domain. As illustrated in Fig. 1b, this is because it excludes unstable features  $X_U$  which are informative about  $Y$  and can boost performance *if used in an appropriate, domain-specific manner*. Assuming we can indeed learn a stable predictor with prior methods, we start by showing how  $X_U$  can be harnessed *with test-domain labels*.

<sup>2</sup>We drop the domain superscript  $e$  when referring to random variables from any environment.



**Boosting the stable predictor.** We describe boosted joint predictions  $f^e(X)$  in domain  $e$  as some combination  $C$  of stable predictions  $f_S(X)$  and domain-specific unstable predictions  $f_U^e(X)$ , i.e.,  $f^e(X) = C(f_S(X), f_U^e(X))$ . To allow us to adapt only the  $X_U$ - $Y$  relation, we decompose the stable  $f_S(X) = h_S(\Phi_S(X))$  and unstable  $f_U^e(X) = h_U^e(\Phi_U(X))$  predictions into feature extractors  $\Phi$  and classifiers  $h$ .  $\Phi_S$  extracts stable components  $X_S = \Phi_S(X)$  of  $X$ ,  $\Phi_U$  extracts unstable components  $X_U = \Phi_U(X)$  of  $X$ ,  $h_S$  is a classifier learned on top of  $\Phi_S$  (shared across domains), and  $h_U^e$  is a *domain-specific* unstable classifier learned on top of  $\Phi_U$  (one per domain). Putting these together,

$$f^e(X) = C(f_S(X), f_U^e(X)) = C(h_S(\Phi_S(X)), h_U^e(\Phi_U(X))) = C(h_S(X_S), h_U^e(X_U)), \quad (3.1)$$

where  $C : [0, 1] \times [0, 1] \rightarrow [0, 1]$  is a *combination function* that combines the stable and unstable predictions. For example, Jiang and Veitch [31, Eq. 2.1] add stable  $p_S$  and unstable  $p_U$  predictions in logit space, i.e.,  $C(p_S, p_U) = \sigma(\text{logit}(p_S) + \text{logit}(p_U))$ . Since it is unclear, *a priori*, how to choose  $C$ , we will leave it unspecified until Thm. 4.4 in § 4, where we derive a principled choice.

**Adapting with labels.** Given a new domain  $e$  and labels  $Y^e$ , we can boost performance by adapting  $h_U^e$ . Specifically, letting  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  be a loss function (e.g., cross-entropy) and  $R^e(f) = \mathbb{E}_{(X,Y)} [\ell(Y, f(X)) | E = e]$  the risk of predictor  $f : \mathcal{X} \rightarrow \mathcal{Y}$  in domain  $e$ , we can adapt  $h_U^e$  to solve:

$$\min_{h_U} R^e(C(h_S \circ \Phi_S, h_U \circ \Phi_U)) \quad (3.2)$$

### 3.2 Harnessing unstable features *without labels*

We now consider the main question of this work—can we reliably harness  $X_U$  *without* test-domain labels? We could, of course, simply select a *fixed* unstable classifier  $h_U^e$  by relying solely on the training domains (e.g., by minimizing average error), and hope that this works for the test-domain  $X_U$ - $Y$  relation. However, by definition of  $X_U$  being unstable, this is clearly not a robust or reliable approach—the focus of our efforts in this work, as illustrated in Table 1. As in § 3.1, we assume that we are able to learn a stable predictor  $f_S$  using prior methods, e.g., IRM [1] or QRM [15].

**From stable predictions to robust pseudo-labels.** While we don’t have labels in the test domain, we *do* have stable predictions. By definition, these are imperfect (i.e., *noisy*) but robust, and can be used to form *pseudo-labels*  $\hat{Y}_i = \arg \max_j (f_S(X_i))_j$ , with  $(f_S(X_i))_j \approx \Pr[Y_i = j | X_S]$  denoting the  $j^{\text{th}}$  entry of the stable prediction for  $X_i$ . Can we somehow use these noisy but robust pseudo-labels to guide our updating of  $h_U^e$ , and, ultimately, our use of  $X_U$  in the test domain?

**From joint to unstable-only risk.** If we simply use our robust pseudo-labels as if they were true labels—updating  $h_U^e$  to minimize the joint risk as in Eq. (3.2)—we arrive at trivial solutions since  $f_S$  already predicts its own pseudo-labels with 100% accuracy. For example, if we follow [31, Eq. 2.1] and use the combination function  $C(p_S, p_U) = \sigma(\text{logit}(p_S) + \text{logit}(p_U))$ , then the trivial solution  $\text{logit}(h_U^e(\cdot)) = 0$  achieves 100% accuracy (and minimizes cross-entropy; see Prop. D.1 of App. D). Thus, we cannot minimize a joint loss involving  $f_S$ ’s predictions when using  $f_S$ ’s pseudo-labels. A sensible alternative is to update  $h_U^e$  to minimize the *unstable-only risk*  $R^e(h_U^e \circ \Phi_U)$ .

**More questions than answers.** While this new procedure *could* work, it raises questions about *when* it will work, or, more precisely, the conditions under which it can be used to safely harness  $X_U$ . We now summarise these questions before addressing them in § 4:

1. **Does it make sense to minimize the unstable-only risk?** In particular, when can we minimize the unstable-predictor risk *alone* or separately, and then arrive at the optimal joint predictor? This cannot always work; e.g., for independent  $X_S, X_U \sim \text{Bernoulli}(1/2)$  and  $Y = X_S \text{ XOR } X_U$ ,  $Y$  is independent of each of  $X_S$  and  $X_U$  and hence cannot be predicted from either alone.
2. **How should we combine predictions?** Is there a principled choice for the combination function  $C$  in Eq. (3.1)? In particular, is there a  $C$  that correctly weights stable and unstable predictions in the test domain? As  $X_U$  could be very strongly or very weakly predictive of  $Y$  in the test domain, this seems a difficult task. Intuitively, correctly weighting stable and unstable predictions requires them to be properly calibrated: do we have any reason to believe that, after training on  $f_S$ ’s pseudo-labels,  $h_U^e$  will be properly calibrated in the test domain?
3. **Can the student outperform the teacher?** Stable predictions likely make mistakes—indeed, this is the motivation for trying to improve them. Is it possible to correct these mistakes with  $X_U$ ? Is it

possible to learn an unstable “student” predictor that outperforms its own supervision signal or “teacher”? Perhaps surprisingly, we show that, for certain types of features, the answer is yes. In fact, even a very weak stable predictor, with performance just above chance, can be used to learn an *optimal* unstable classifier in the test domain given enough unlabeled data.

## 4 Theory: When Can We Safely Harness Unstable Features Without Labels?

Suppose we have already identified a stable feature  $X_S$  and a potentially unstable feature  $X_U$  (we will return to the question of how to learn/extract  $X_S$  and  $X_U$  themselves in § 5). In this section, we analyze the problem of using  $X_S$  to leverage  $X_U$  in the test domain without labels. We first reduce this to a special case of the so-called “marginal problem” in probability theory, i.e., the problem of identifying a joint distribution based on information about its marginals. In the special case where two variables are conditionally independent given a third, we show this problem can be solved exactly. This solution, which may be of interest beyond the context of domain generalization/adaptation, motivates our test-domain adaptation algorithm (Alg. 1), and forms the basis of Thm. 4.6 which shows that Alg. 1 converges to the best possible classifier given enough unlabeled data.

We first pose a population-level model of our domain generalization setup. Let  $E$  be a random variable denoting the *environment*. Given an environment  $E$ , we have that the stable feature  $X_S$ , unstable feature  $X_U$  and label  $Y$  are distributed according to  $P_{X_S, X_U, Y|E}$ . We can now formalize the three key assumptions underlying our approach, starting with the notion of a stable feature, motivated in § 3:

**Definition 4.1** (Stable and Unstable Features).  $X_S$  is a stable feature with respect to  $Y$  if  $P_{Y|X_S}$  does not depend on  $E$ ; equivalently, if  $Y$  and  $E$  are conditionally independent given  $X_S$  ( $Y \perp\!\!\!\perp E | X_S$ ). Conversely,  $X_U$  is an unstable feature with respect to  $Y$  if  $P_{Y|X_U}$  depends on  $E$ ; equivalently, if  $Y$  and  $E$  are conditionally dependent given  $X_U$  ( $Y \not\perp\!\!\!\perp E | X_U$ ).

Next, we state our complementarity assumption, which we will show justifies the approach of separately learning the relationships  $X_S$ - $Y$  and  $X_U$ - $Y$  and then combining them:

**Definition 4.2** (Complementary Features).  $X_S$  and  $X_U$  are complementary features with respect to  $Y$  if  $X_S \perp\!\!\!\perp X_U | (Y, E)$ ; i.e., if  $X_S$  and  $X_U$  share no redundant information beyond  $Y$  and  $E$ .

Finally, to provide a useful signal for test-domain adaptation, the stable feature needs to help predict the label in the test domain. Formally, we assume:

**Definition 4.3** (Informative Feature).  $X_S$  is said to be informative of  $Y$  in environment  $E$  if  $X_S \not\perp\!\!\!\perp Y | E$ ; i.e.,  $X_S$  is predictive of  $Y$  within the environment  $E$ .

We will discuss the roles of these assumptions after stating our main result (Thm. 4.4) that uses them. To keep our results as general as possible, we avoid assuming a particular causal generative model, but the above conditional (in)dependence assumptions can be interpreted as constraints on such a causal model. App. D.2 formally characterizes the set of causal models that are consistent with our assumptions and shows that our setting generalizes those of prior works [49, 68, 31, 69].

**Reduction to the marginal problem with complementary features.** By Defn. 4.1, we have the same stable relationship  $P_{Y|X_S, E} = P_{Y|X_S}$  in training and test domains. Now, suppose we have used the training data to learn this stable relationship and thus know  $P_{Y|X_S}$ . Also suppose that we have enough unlabeled data from test domain  $E$  to learn  $P_{X_S, X_U|E}$ , and recall that our ultimate goal is to predict  $Y$  from  $(X_S, X_U)$  in test domain  $E$ . Since the rest of our discussion is conditioned on  $E$  being the test domain, we omit  $E$  from the notation. Now note that, if we could express  $P_{Y|X_S, X_U}$  in terms of  $P_{Y|X_S}$  and  $P_{X_S, X_U}$ , we could then use  $P_{Y|X_S, X_U}$  to optimally predict  $Y$  from  $(X_S, X_U)$ . Thus, our task thus becomes to reconstruct  $P_{Y|X_S, X_U}$  from  $P_{Y|X_S}$  and  $P_{X_S, X_U}$ . This is an instance of the classical “marginal problem” from probability theory [28, 29, 19], which asks under which conditions we can recover the joint distribution of a set of random variables given information about its marginals. In general, although one can place bounds on the conditional distributions  $P_{Y|X_U}$  and  $P_{Y|X_S, X_U}$ , they cannot be completely inferred from  $P_{Y|X_S}$  and  $P_{X_S, X_U}$  [19]. However, the following section demonstrates that, *under the additional assumptions that  $X_S$  and  $X_U$  are complementary and  $X_S$  is informative*, we can exactly recover  $P_{Y|X_U}$  and  $P_{Y|X_S, X_U}$  from  $P_{Y|X_S}$  and  $P_{X_S, X_U}$ .

#### 4.1 Solving the marginal problem with complementary features

We now present our main result which shows how to reconstruct  $P_{Y|X_S, X_U}$  from  $P_{Y|X_S}$  and  $P_{X_S, X_U}$  when  $X_S$  and  $X_U$  are complementary and  $X_S$  is informative. To simplify notation, we assume the label  $Y$  is binary and defer the multi-class extension to App. C.

**Theorem 4.4** (Solution to the marginal problem with binary labels and complementary features). *Consider three random variables  $X_S$ ,  $X_U$ , and  $Y$ , where (i)  $Y$  is binary ( $\{0, 1\}$ -valued), (ii)  $X_S$  and  $X_U$  are complementary features for  $Y$  (i.e.,  $X_S \perp\!\!\!\perp X_U | Y$ ), and (iii)  $X_S$  is informative of  $Y$  ( $X_S \not\perp\!\!\!\perp Y$ ). Then, the joint distribution of  $(X_S, X_U, Y)$  can be written in terms of the joint distributions of  $(X_S, Y)$  and  $(X_S, X_U)$ . Specifically, if  $\hat{Y}|X_S \sim \text{Bernoulli}(\Pr[Y = 1|X_S])$  is a pseudo-label<sup>3</sup> and*

$$\epsilon_0 := \Pr[\hat{Y} = 0|Y = 0] \quad \text{and} \quad \epsilon_1 := \Pr[\hat{Y} = 1|Y = 1] \quad (4.1)$$

are the accuracies of the pseudo-labels on classes 0 and 1, respectively. Then, we have:

$$\epsilon_0 + \epsilon_1 > 1, \quad (4.2)$$

$$\Pr[Y = 1|X_U] = \frac{\Pr[\hat{Y} = 1|X_U] + \epsilon_0 - 1}{\epsilon_0 + \epsilon_1 - 1}, \quad \text{and} \quad (4.3)$$

$$\Pr[Y = 1|X_S, X_U] = \sigma(\text{logit}(\Pr[Y = 1|X_S]) + \text{logit}(\Pr[Y = 1|X_U]) - \text{logit}(\Pr[Y = 1])). \quad (4.4)$$

Intuitively, suppose we generate pseudo-labels  $\hat{Y}$  based on feature  $X_S$  and train a model to predict  $\hat{Y}$  using feature  $X_U$ . For complementary  $X_S$  and  $X_U$ , Eq. (4.3) shows how to transform this into a prediction of the *true* label  $Y$ , correcting for differences between  $\hat{Y}$  and  $Y$ . Crucially, given the conditional distribution  $P_{Y|X_S}$  and observations of  $X_S$ , we can estimate class-wise pseudo-label accuracies  $\epsilon_0$  and  $\epsilon_1$  in Eq. (4.3) even without new labels  $Y$  (see App. A.1, Eq. (A.2)). Finally, Eq. (4.4) shows how to weight predictions based on  $X_S$  and  $X_U$ , justifying the combination function

$$C_p(p_S, p_U) = \sigma(\text{logit}(p_S) + \text{logit}(p_U) - \text{logit}(p)) \quad (4.5)$$

in Eq. (3.1), where  $p = \Pr[Y = 1]$  is a constant independent of  $X_S$  and  $X_U$ . We now sketch the proof of Thm. 4.4, elucidating the roles of informativeness and complementarity (full proof in App. A.1).

*Proof Sketch of Thm. 4.4.* We prove Eq. (4.2), Eq. (4.3), and Eq. (4.4) in order.

**Proof of Eq. (4.2):** The informativeness condition (iii) is equivalent to the pseudo-labels having predictive accuracy above random chance; formally, App. A.1 shows:

**Lemma 4.5.**  $\epsilon_0 + \epsilon_1 > 1$  if and only if  $X_S$  is informative of  $Y$  (i.e.,  $X_S \not\perp\!\!\!\perp Y$ ).

Together with Eq. (4.3), it follows that *any* dependence between  $X_S$  and  $Y$  allows us to fully learn the relationship between  $X_U$  and  $Y$ , affirmatively answering our question from § 3: *Can the student outperform the teacher?* While a stronger relationship between  $X_S$  and  $Y$  is still helpful, it only improves the (unlabeled) *sample complexity* of learning  $P_{Y|X_U}$  and not *consistency* (Thm. 4.6 below), mirroring related results in the literature on learning from noisy labels [44, 7, 75]. In particular, a weak relationship corresponds to  $\epsilon_0 + \epsilon_1 \approx 1$ , increasing the variance of the bias-correction in Eq. (4.3). With a bit more work, one can formalize this intuition to show that our approach has a relative statistical efficiency of  $\epsilon_0 + \epsilon_1 - 1 \in [0, 1]$ , compared to using true labels  $Y$ .

**Proof of Eq. (4.3):** The key observation behind the bias-correction (Eq. (4.3)) is that, due to complementarity ( $X_S \perp\!\!\!\perp X_U | Y$ ) and the fact that the pseudo-label  $\hat{Y}$  depends only on  $X_S$ ,  $\hat{Y}$  is conditionally independent of  $X_U$  given the true label  $Y$  ( $\hat{Y} \perp\!\!\!\perp X_U | Y$ ); formally:

$$\begin{aligned} \Pr[\hat{Y} = 1|X_U] &= \Pr[\hat{Y} = 1|Y = 0, X_U] \Pr[Y = 0|X_U] \\ &\quad + \Pr[\hat{Y} = 1|Y = 1, X_U] \Pr[Y = 1|X_U] \quad (\text{Law of Total Probability}) \\ &= \Pr[\hat{Y} = 1|Y = 0] \Pr[Y = 0|X_U] \\ &\quad + \Pr[\hat{Y} = 1|Y = 1] \Pr[Y = 1|X_U] \quad (\text{Complementarity}) \\ &= (\epsilon_0 + \epsilon_1 - 1) \Pr[Y = 1|X_U] + 1 - \epsilon_0. \quad (\text{Definitions of } \epsilon_0 \text{ and } \epsilon_1) \end{aligned}$$

<sup>3</sup>Our *stochastic* pseudo-labels differ from hard ( $\hat{Y} = 1\{\Pr[Y = 1|X_S] > 1/2\}$ ) pseudo-labels often used in practice [20, 36, 52]. By capturing irreducible error in  $Y$ , stochastic pseudo-labels ensure  $\Pr[Y|X_U]$  is well-calibrated, allowing us to combine  $\Pr[Y|X_S]$  and  $\Pr[Y|X_U]$  in Eq. (4.4).

---

**Algorithm 1:** Bias-corrected adaptation procedure. Multi-class version given by Algorithm 2.

---

**Input:** Calibrated stable classifier  $f_S(x_S) = \Pr[Y = 1 | X_S = x_S]$ , unlabelled data  $\{(X_{S,i}, X_{U,i})\}_{i=1}^n$

**Output:** Joint classifier  $\hat{f}(x_S, x_U)$  estimating  $\Pr[Y = 1 | X_S = x_S, X_U = x_U]$

- 1 Compute soft pseudo-labels (PLs)  $\{\hat{Y}_i\}_{i=1}^n$  with  $\hat{Y}_i = f_S(X_{S,i})$
  - 2 Compute soft class-1 count  $n_1 = \sum_{i=1}^n \hat{Y}_i$
  - 3 Estimate PL accuracies  $(\hat{\epsilon}_0, \hat{\epsilon}_1) = \left( \frac{1}{n-n_1} \sum_{i=1}^n (1-\hat{Y}_i)(1-f_S(X_{S,i})), \frac{1}{n_1} \sum_{i=1}^n \hat{Y}_i f_S(X_{S,i}) \right)$  // Eq. (4.1)
  - 4 Fit unstable classifier  $\tilde{f}_U(x_U)$  to pseudo-labelled data  $\{(X_{U,i}, \hat{Y}_i)\}_{i=1}^n$  //  $\approx \Pr[\hat{Y} = 1 | X_U = x_U]$
  - 5 Bias-correct  $\hat{f}_U(x_U) \mapsto \max \left\{ 0, \min \left\{ 1, \frac{\tilde{f}_U(x_U) + \hat{\epsilon}_0 - 1}{\hat{\epsilon}_0 + \hat{\epsilon}_1 - 1} \right\} \right\}$  // Eq. (4.3),  $\approx \Pr[Y = 1 | X_U = x_U]$
  - 6 **return**  $\hat{f}(x_S, x_U) \mapsto C_{\frac{n_1}{n}}(f_S(x_S), \hat{f}_U(x_U))$  // Eq. (4.4)/(4.5),  $\approx \Pr[Y = 1 | X_S = x_S, X_U = x_U]$
- 

Here, complementarity allowed us to approximate the unknown  $\Pr[\hat{Y} = 1 | Y = 0, X_U]$  by its average  $\Pr[\hat{Y} = 1 | Y = 0] = \mathbb{E}_{X_U}[\Pr[\hat{Y} = 1 | Y = 0, X_U]]$ , which depends only on the known distribution  $P_{X_S, Y}$ . By informativeness, Lemma 4.5 allows us to divide by  $\epsilon_0 + \epsilon_1 - 1$ , giving Eq. (4.3).

**Proof of Eq. (4.4):** While the exact proof of Eq. (4.4) is a bit more algebraically involved, the key idea is simply that complementarity allows us to decompose  $\Pr[Y | X_S, X_U]$  into separately-estimatable terms  $\Pr[Y | X_S]$  and  $\Pr[Y | X_U]$ : for any  $y \in \mathcal{Y}$ ,

$$\begin{aligned} \Pr[Y = y | X_S, X_U] &\propto_{X_S, X_U} \Pr[X_S, X_U | Y = y] \Pr[Y = y] && \text{(Bayes' Rule)} \\ &= \Pr[X_S | Y = y] \Pr[X_U | Y = 1] \Pr[Y = y] && \text{(Complementarity)} \\ &\propto_{X_S, X_U} \frac{\Pr[Y = y | X_S] \Pr[Y = 1 | X_U]}{\Pr[Y = 1]}, && \text{(Bayes' Rule)} \end{aligned}$$

where,  $\propto_{X_S, X_U}$  denotes proportionality with a constant depending only on  $X_S$  and  $X_U$ , not on  $y$ . Directly estimating these constants involves estimating the density of  $(X_S, X_U)$ , which may be intractable without further assumptions. However, in the binary case, since  $1 - \Pr[Y = 1 | X_S, X_U] = \Pr[Y = 0 | X_S, X_U]$ , these proportionality constants conveniently cancel out when the above relationship is written in logit-space, as in Eq. (4.4). In the multi-class case, App. C shows how to use the constraint  $\sum_{y \in \mathcal{Y}} \Pr[Y = y | X_S, X_U] = 1$  to avoid computing the proportionality constants.  $\square$

## 4.2 A provably consistent algorithm for unsupervised test-domain adaptation

Having learned  $P_{Y|X_S}$  from the training domain(s), Thm. 4.4 implies we can learn  $P_{Y|X_S, X_U}$  in the test domain by learning  $P_{X_S, X_U}$ —the latter only requiring *unlabeled* test-domain data. This motivates our Alg. 1 for test-domain adaptation, which is a finite-sample version of the bias-correction and combination equations (Eqs. (4.3) and (4.4)) in Thm. 4.4. Alg. 1 comes with the following guarantee:

**Theorem 4.6** (Consistency Guarantee, Informal). *Assume (i)  $X_S$  is stable, (ii)  $X_S$  and  $X_U$  are complementary, and (iii)  $X_S$  is informative of  $Y$  in the test domain. As  $n \rightarrow \infty$ , if  $\tilde{f}_U \rightarrow \Pr[\hat{Y} = 1 | X_U]$  then  $\hat{f} \rightarrow \Pr[Y = 1 | X_S, X_U]$ .*

In words, as the amount of unlabeled data from the test domain increases, if the unstable classifier on Line 4 of Alg. 1 learns to predict the pseudo-label  $\hat{Y}$ , then the joint classifier output by Alg. 1 learns to predict the true label  $Y$ . Convergence in Thm. 4.6 occurs  $P_{X_S, X_U}$ -a.e., both weakly (in prob.) and strongly (a.s.), depending on the convergence of  $\tilde{f}_U$ . Formal statements and proofs are in Appendix B.

## 5 Algorithm: Stable Feature Boosting (SFB)

Using theoretical insights from § 4, we now propose Stable Feature Boosting (SFB): an algorithm for safely harnessing unstable features without test-domain labels. We first describe learning a stable predictor and extracting complementary unstable features from the training domains. We then describe how to use these with Alg. 1, adapting our use of the unstable features to the test domain.

**Training domains: Learning stable and complementary features.** Using the notation of Eq. (3.1), our goal on the training domains is to learn stable and unstable features  $\Phi_S$  and  $\Phi_U$ , a stable predictor  $f_S$ , and domain-specific unstable predictors  $f_U^e$  such that:

1.  $f_S$  is stable, informative, and calibrated (i.e.,  $f_S(x_S) = \Pr[Y = 1 | X_S = x_S]$ ).
2. In domain  $e$ ,  $f_U^e$  boosts  $f_S$ 's performance with complementary  $\Phi_U(X^e) \perp\!\!\!\perp \Phi_S(X^e) | Y^e$ .

To achieve these learning goals, we propose the following objective:

$$\begin{aligned} \min_{\Phi_S, \Phi_U, h_S, h_U^e} \sum_{e \in \mathcal{E}_t} R^e(h_S \circ \Phi_S) + R^e(C(h_S \circ \Phi_S, h_U^e \circ \Phi_U)) \\ + \lambda_S \cdot P_{\text{Stability}}(\Phi_S, h_S, R^e) + \lambda_C \cdot P_{\text{CondIndep}}(\Phi_S(X^e), \Phi_U(X^e), Y^e) \end{aligned} \quad (5.1)$$

The first term encourages good stable predictions  $f_S(X) = h_S(\Phi_S(X))$  while the second encourages improved domain-specific joint predictions  $f^e(X^e) = C(h_S(\Phi_S(X^e)), h_U^e(\Phi_U(X^e)))$  via a domain-specific use  $h_U^e$  of the unstable features  $\Phi_U(X^e)$ . For binary  $Y$ , the combination function  $C$  takes the simplified form of Eq. (4.5). Otherwise,  $C$  takes the more general form of Eq. (C.1).  $P_{\text{Stability}}$  is a penalty encouraging stability while  $P_{\text{CondIndep}}$  is a penalty encouraging complementarity or conditional independence, i.e.,  $\Phi_U(X^e) \perp\!\!\!\perp \Phi_S(X^e) | Y^e$ . Several approaches exist for enforcing stability [1, 35, 58, 47, 15, 67, 40, 78] (e.g., IRM [1]) and conditional independence (e.g., conditional HSIC [22]).  $\lambda_S \in [0, \infty)$  and  $\lambda_C \in [0, \infty)$  are regularization hyperparameters. While another hyperparameter  $\gamma \in [0, 1]$  could control the relative weighting of stable and joint risks, i.e.,  $\gamma R^e(h_S \circ \Phi_S)$  and  $(1 - \gamma) R^e(C(h_S \circ \Phi_S, h_U^e \circ \Phi_U))$ , we found this unnecessary in practice. Finally, note that, in principle,  $h_U^e$  could take any form and we could learn completely separate  $\Phi_S, \Phi_U$ . In practice, we simply take  $h_U^e$  to be a linear classifier and split the output of a shared  $\Phi(X) = (\Phi_S(X), \Phi_U(X))$ .

**Post-hoc calibration.** As noted in § 4.2, the stable predictor  $f_S$  must be properly calibrated to (i) form unbiased unstable predictions (Line 5 of Alg. 1) and (ii) correctly combine the stable and unstable predictions (Line 6 of Alg. 1). Thus, after optimizing the objective (5.1), we apply a post-processing step (e.g., temperature scaling [25]) to calibrate  $f_S$ .

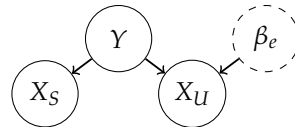
**Test-domain adaptation without labels.** Given a stable predictor  $f_S = h_S \circ \Phi_S$  and complementary features  $\Phi_U(X)$ , we now adapt the unstable classifier  $h_U^e$  in the test domain to safely harness (or make optimal use of)  $\Phi_U(X)$ . To do so, we use the bias-corrected adaptation algorithm of Alg. 1 (or Alg. 2 for the multi-class case) which takes as input the stable classifier  $h_S^4$  and unlabelled test-domain data  $\{\Phi_S(x_i), \Phi_U(x_i)\}_{i=1}^{n_e}$ , outputting a joint classifier adapted to the test domain.

## 6 Experiments

We now evaluate the performance of our algorithm on synthetic and real-world datasets requiring out-of-distribution generalization. App. E contains full details on these datasets and a depiction of their samples (see Fig. 4). In the experiments below, SFB uses IRM [1] for  $P_{\text{Stability}}$  and the conditional-independence proxy of Jiang and Veitch [31, §3.1] for  $P_{\text{CondIndep}}$ , with App. F.1.2 giving results with other stability penalties. App. F contains further results, including ablation studies (F.1.1) and results on additional datasets (F.2). In particular, App. F.2 contains results on the Came1yon17 medical dataset [3] from the WILDS package [33], where we find that all methods perform similarly *when properly tuned* (see discussion in App. F.2). Code is available at: <https://github.com/cianeastwood/sfb>.

**Synthetic data.** We consider two synthetic datasets: anti-causal (AC) data and cause-effect data with direct  $X_S$ - $X_U$  dependence (CE-DD). AC data satisfies the structural equations

$$\begin{aligned} Y &\leftarrow \text{Rad}(0.5); \\ X_S &\leftarrow Y \cdot \text{Rad}(0.75); \\ X_U &\leftarrow Y \cdot \text{Rad}(\beta_e), \end{aligned}$$



where the input  $X = (X_S, X_U)$  and  $\text{Rad}(\beta)$  denotes a Rademacher random variable that

<sup>4</sup>Note: while Sections 3 and 5 use  $h$  for the classifier and  $f = h \circ \Phi$  for the classifier-representation composition, Section 4 and Alg. 1 use  $f$  for the classifier, since no representation  $\Phi$  is being learned.

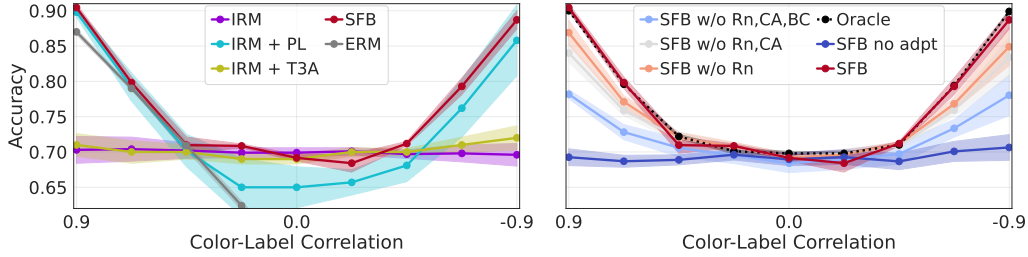


Table 2: Synthetic &amp; PACS test-domain accuracies over 100 &amp; 5 seeds each.

Algorithm	Synthetic			PACS		
	AC	CE-DD	P	A	C	S
ERM	9.9 ± 0.1	11.6 ± 0.7	93.0 ± 0.7	79.3 ± 0.5	74.3 ± 0.7	65.4 ± 1.5
ERM + PL	9.9 ± 0.1	11.6 ± 0.7	93.7 ± 0.4	79.6 ± 1.5	74.1 ± 1.2	63.1 ± 3.1
IRM [1]	74.9 ± 0.1	69.6 ± 1.3	93.3 ± 0.3	78.7 ± 0.7	75.4 ± 1.5	65.6 ± 2.5
IRM + PL	74.9 ± 0.1	69.6 ± 1.3	94.1 ± 0.7	78.9 ± 2.9	75.1 ± 4.6	62.9 ± 4.9
ACTIR [31]	74.8 ± 0.4	43.5 ± 2.6	94.8 ± 0.1	<b>82.5 ± 0.4</b>	<b>76.6 ± 0.6</b>	62.1 ± 1.3
SFB no adpt.	74.7 ± 1.2	74.9 ± 3.6	93.7 ± 0.6	78.1 ± 1.1	73.7 ± 0.6	69.7 ± 2.3
SFB	<b>89.2 ± 2.9</b>	<b>88.6 ± 1.4</b>	<b>95.8 ± 0.6</b>	80.4 ± 1.3	<b>76.6 ± 0.6</b>	<b>71.8 ± 2.0</b>

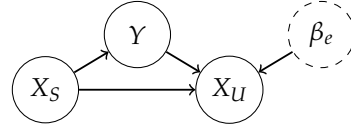
Table 3: CMNIST test accuracies over 10 seeds.

Algorithm	Test Acc.
ERM	27.9 ± 1.5
IRM [1]	69.7 ± 0.9
SFB no adpt.	70.6 ± 1.8
SFB	<b>88.1 ± 1.8</b>
Oracle no adpt.	72.1 ± 0.7
Oracle	89.9 ± 0.1

Figure 2: CMNIST accuracies (y-axis) over test domains of decreasing color-label correlation (x-axis). Empirical versions of Fig. 1b. *Left*: SFB vs. baseline methods. *Right*: Ablations showing SFB without (w/o) bias correction (BC), calibration (CA) and multiple pseudo-labeling rounds (Rn). Numerical results in Table 7 of App. F.1.3.

is  $-1$  with probability  $1 - \beta$  and  $+1$  with probability  $\beta$ . Following [31, §6.1], we create two training domains with  $\beta_e \in \{0.95, 0.7\}$ , one validation domain with  $\beta_e = 0.6$  and one test domain with  $\beta_e = 0.1$ . CE-DD data is generated according to the structural equations

$$\begin{aligned} X_S &\leftarrow \text{Bern}(0.5); \\ Y &\leftarrow \text{XOR}(X_S, \text{Bern}(0.75)); \\ X_U &\leftarrow \text{XOR}(\text{XOR}(Y, \text{Bern}(\beta_e)), X_S), \end{aligned}$$



where  $\text{Bern}(\beta)$  denotes a Bernoulli random variable that is 1 with probability  $\beta$  and 0 with probability  $1 - \beta$ . Note that  $X_S \not\perp\!\!\!\perp X_U | Y$ , since  $X_S$  directly influences  $X_U$ . Following [31, App. B], we create two training domains with  $\beta_e \in \{0.95, 0.8\}$ , one validation domain with  $\beta_e = 0.2$ , and one test domain with  $\beta_e = 0.1$ . For both datasets, the idea is that, during training, prediction based on the stable  $X_S$  results in lower accuracy (75%) than prediction based on the unstable  $X_U$ . Thus, models optimizing for prediction accuracy only—and not stability—will use  $X_U$  and ultimately end up with only 10% in the test domain. Importantly, while the stable predictor achieves 75% accuracy in the test domain, this can be improved to 90% if  $X_U$  is used correctly. Following [31], we use a simple 3-layer network for both datasets and choose hyperparameters using the validation-domain performance: see App. G.2 for further implementation details.

On the AC dataset, Table 2 shows that ERM performs poorly as it misuses  $X_U$ , while IRM, ACTIR, and SFB-no-adpt. do well by using only  $X_S$ . Critically, only SFB (with adaptation) is able to harness  $X_U$  in the test domain *without labels*, leading to a near-optimal performance boost.

On the CE-DD dataset, Table 2 again shows that ERM performs poorly while IRM and SFB-no-adpt. do well by using only the stable  $X_S$ . However, we now see that ACTIR performs poorly since its assumption of anti-causal structure no longer holds. This highlights another key advantage of SFB over ACTIR: any stability penalty can be used, including those with weaker assumptions than ACTIR’s anti-causal structure (e.g., IRM). Perhaps more surprisingly, SFB (with adaptation) performs well despite the complementarity assumption  $X_S \perp\!\!\!\perp X_U | Y$  being violated. One explanation for this is that complementarity is only weakly violated in the test domain. Another is that complementarity is not *necessary* for SFB, with some weaker, yet-to-be-determined condition(s) sufficing. In App. I, we provide a more detailed explanation and discussion of this observation.

**ColorMNIST.** We now consider the ColorMNIST dataset [1], described in § 1 and Fig. 1a. We follow the experimental setup of Eastwood et al. [15, §6.1]; see App. G.3 for details. Table 3 shows that: (i) SFB learns a stable predictor (“no adpt.”) with performance comparable to other stable/invariant

methods like IRM [1]; and (ii) only SFB (with adaptation) is capable of harnessing the spurious color feature in the test domain *without labels*, leading to a near-optimal boost in performance. Note that “Oracle no adpt.” refers to an ERM model trained on grayscale images, while “Oracle” refers to an ERM model trained on labeled test-domain data. Table 6 of App. F.1.3 compares to additional baseline methods, including V-REx [35], EQRM [15], Fishr [48] and more. Fig. 2 gives more insight by showing performance across test domains of varying color-label correlation. On the left, we see that SFB outperforms ERM and IRM, as well as additional adaptive baseline methods in IRM + pseudo-labeling (PL, [36]) and IRM + T3A [30] (see App. G.1 for details). On the right, ablations show that: (i) bias-correction (BC), post-hoc calibration (CA), and multiple rounds of pseudo-labeling (Rn) improve adaptation performance; and (ii) without labels, SFB harnesses the spurious color feature near-optimally in test domains of varying color-label correlation—the original goal we set out to achieve in Fig. 1b. Further results and ablations are provided in App. F.1.

**PACS.** Table 2 shows that SFB’s stable (“no adpt.”) performance is comparable to that of the other stable/invariant methods (IRM, ACTIR). One exception is the sketch domain (S)—the most severe shift based on performance drop—where SFB’s stable predictor performs best. Another is on domains A and C, where ACTIR performs better than SFB’s stable predictor. Most notable, however, is: (i) the consistent performance boost that SFB gets from unsupervised adaptation; and (ii) SFB performing best or joint-best on 3 of the 4 domains. These results suggest SFB can be useful on real-world datasets where it is unclear if complementarity holds. In App. I, we discuss why this may be the case.

## 7 Conclusion & Future Work

This work demonstrated, both theoretically and practically, how to adapt our usage of spurious features to new test domains using only a stable, complementary training signal. By using invariant predictions to safely harness complementary spurious features, our proposed Stable Feature Boosting algorithm can provide significant performance gains compared to only using invariant/stable features or using unadapted spurious features—without requiring any true labels in the test domain.

**Stable and calibrated predictors.** Perhaps the greatest challenge in applying SFB in practice is the need for a stable and calibrated predictor. While stable features may be directly observable in some cases (e.g., using prior knowledge of causal relationships between the domain, features, and label, as in Prop. D.2), they often need to be extracted from high-dimensional observations (e.g., images). Several methods for stable-feature extraction have recently been proposed [1, 35, 58, 70, 15], with future improvements likely to benefit SFB. Calibrating complex predictors like deep neural networks is also an active area of research [18, 25, 72, 59], with future improvements likely to benefit SFB.

**Weakening the complementarity condition.** SFB also assumes that stable and unstable features are complementarity, i.e., conditionally independent given the label. This assumption is implicit in the causal generative models assumed by prior work [49, 68, 31], and future work may look to weaken it. However, our experimental results suggest that SFB may be robust to violations of complementarity in practice: on our synthetic data where complementarity does not hold (CE-DD) and real data where we have no reason to believe it holds (PACS), SFB still outperformed baseline methods. We discuss potential reasons for this in App. I and hope that future work can identify weaker sufficient conditions.

**Exploiting newly-available test-domain features without labels.** While we focused on domain generalization (DG) and the goal of (re)learning how to use the same spurious features (e.g., color) in a new way, our solution to the “marginal problem” in § 4.1 can be used to exploit a completely new set of (complementary) features in the test domain that weren’t available in the training domains. For example, given a stable predictor of diabetes based on causal features (e.g., age, genetics), SFB could exploit new unlabeled data containing previously-unseen effect features (e.g., glucose levels). We hope future work can explore such uses of SFB.

## Acknowledgments and Disclosure of Funding

The authors thank Chris Williams and Ian Mason for providing feedback on an earlier draft, as well as the MPI Tübingen causality group for helpful discussions and comments. This work was supported by the Tübingen AI Center (FKZ: 01IS18039B) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC number 2064/1 – Project number 390727645. The authors declare no competing interests.

## References

- [1] Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2020). Invariant risk minimization. arXiv:1907.02893. [Cited on pages 1, 2, 3, 4, 8, 9, 10, 26, 27, 29, and 31.]
- [2] Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? *Advances in Neural Information Processing Systems*, 27. [Cited on page 19.]
- [3] Bandi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M., Hermsen, M., Bejnordi, B. E., Lee, B., Paeng, K., Zhong, A., et al. (2018). From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560. [Cited on pages 8, 27, 28, and 29.]
- [4] Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, pages 456–473. [Cited on page 3.]
- [5] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, 79(1):151–175. [Cited on page 32.]
- [6] Bickel, S., Brückner, M., and Scheffer, T. (2009). Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9). [Cited on page 26.]
- [7] Blanchard, G., Flaska, M., Handy, G., Pozzi, S., and Scott, C. (2016). Classification with asymmetric label noise: Consistency and maximal denoising. *Electronic Journal of Statistics*, 10:2780–2824. [Cited on pages 6, 19, and 32.]
- [8] Blanchard, G., Lee, G., and Scott, C. (2011). Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems*, volume 24. [Cited on page 3.]
- [9] Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J. (2007). Learning bounds for domain adaptation. *Advances in Neural Information Processing Systems*, 20. [Cited on page 32.]
- [10] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. [Cited on page 32.]
- [11] Buciluă, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. [Cited on page 19.]
- [12] Bui, M.-H., Tran, T., Tran, A., and Phung, D. (2021). Exploiting domain-specific features to enhance domain generalization. In *Advances in Neural Information Processing Systems*, volume 34. [Cited on page 2.]
- [13] Eastwood, C., Mason, I., and Williams, C. (2021). Unit-level surprise in neural networks. In *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*. [Cited on page 3.]
- [14] Eastwood, C., Mason, I., Williams, C., and Schölkopf, B. (2022a). Source-free adaptation to measurement shift via bottom-up feature restoration. In *International Conference on Learning Representations*. [Cited on page 2.]
- [15] Eastwood, C., Robey, A., Singh, S., von Kügelgen, J., Hassani, H., Pappas, G. J., and Schölkopf, B. (2022b). Probable domain generalization via quantile risk minimization. In *Advances in Neural Information Processing Systems*. [Cited on pages 1, 2, 3, 4, 8, 9, 10, 29, and 31.]
- [16] Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611. [Cited on page 3.]
- [17] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. [Cited on page 3.]
- [18] Flach, P. A. (2016). Classifier calibration. In *Encyclopedia of machine learning and data mining*. Springer US. [Cited on page 10.]



- [19] Fréchet, M. (1951). Sur les tableaux de corrélation dont les marges sont données. *Ann. Univ. Lyon, 3<sup>e</sup> serie, Sciences, Sect. A*, 14:53–77. [Cited on page 5.]
- [20] Galstyan, A. and Cohen, P. R. (2008). Empirical comparison of “hard” and “soft” label propagation for relational classification. In *Inductive Logic Programming: 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, Revised Selected Papers 17*, pages 98–111. Springer. [Cited on pages 2 and 6.]
- [21] Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673. [Cited on pages 1 and 3.]
- [22] Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory: 16th International Conference, ALT 2005, Singapore, October 8-11, 2005. Proceedings 16*, pages 63–77. Springer. [Cited on page 8.]
- [23] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5. [Cited on page 26.]
- [24] Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*. [Cited on pages 3, 27, 30, and 31.]
- [25] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. [Cited on pages 8, 10, 28, 31, and 32.]
- [26] Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*. [Cited on page 1.]
- [27] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*. [Cited on page 19.]
- [28] Hoeffding, W. (1940). Masstabinvariante korrelationstheorie. *Schriften des Mathematischen Instituts und Instituts für Angewandte Mathematik der Universität Berlin*, 5:181–233. [Cited on page 5.]
- [29] Hoeffding, W. (1941). Masstabinvariante korrelationsmasse für diskontinuierliche verteilungen. *Archiv für mathematische Wirtschafts- und Sozialforschung*, 7:49–70. [Cited on page 5.]
- [30] Iwasawa, Y. and Matsuo, Y. (2021). Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems*. [Cited on pages 2, 10, 30, and 31.]
- [31] Jiang, Y. and Veitch, V. (2022). Invariant and transportable representations for anti-causal domain shifts. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*. [Cited on pages 3, 4, 5, 8, 9, 10, 25, 26, 27, 30, 31, and 32.]
- [32] Kirichenko, P., Izmailov, P., and Wilson, A. G. (2022). Last layer re-training is sufficient for robustness to spurious correlations. In *Advances in Neural Information Processing Systems*. [Cited on page 3.]
- [33] Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B. A., Haque, I. S., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. (2021). WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*. [Cited on pages 8, 27, and 32.]
- [34] Krogel, M.-A. and Scheffer, T. (2004). Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning*, 57:61–81. [Cited on page 32.]

- [35] Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (REx). In *International Conference on Machine Learning*, volume 139, pages 5815–5826. [Cited on pages 1, 8, 10, 29, and 31.]
- [36] Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3. [Cited on pages 2, 6, 10, and 30.]
- [37] Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017a). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. [Cited on page 27.]
- [38] Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. (2017b). Learning from noisy labels with distillation. In *Proceedings of the IEEE international conference on computer vision*, pages 1910–1918. [Cited on page 32.]
- [39] Liang, J., Hu, D., and Feng, J. (2020). Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039. [Cited on page 2.]
- [40] Makar, M., Packer, B., Moldovan, D., Blalock, D., Halpern, Y., and D’Amour, A. (2022). Causally motivated shortcut removal using auxiliary labels. In *International Conference on Artificial Intelligence and Statistics*, pages 739–766. PMLR. [Cited on pages 2, 8, and 29.]
- [41] Mansour, Y., Mohri, M., and Rostamizadeh, A. (2008). Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21. [Cited on page 33.]
- [42] Muandet, K., Balduzzi, D., and Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. [Cited on page 3.]
- [43] Nagarajan, V., Andreassen, A., and Neyshabur, B. (2021). Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*. [Cited on page 3.]
- [44] Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. *Advances in neural information processing systems*, 26. [Cited on pages 6, 19, and 32.]
- [45] Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012. [Cited on pages 1, 2, and 3.]
- [46] Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., and Lajoie, G. (2021). Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272. [Cited on page 29.]
- [47] Puli, A. M., Zhang, L. H., Oermann, E. K., and Ranganath, R. (2022). Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*. [Cited on pages 3, 8, and 29.]
- [48] Rame, A., Dancette, C., and Cord, M. (2022). Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. [Cited on pages 10 and 29.]
- [49] Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. (2018). Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342. [Cited on pages 3, 5, 10, 25, and 26.]
- [50] Rosenfeld, E., Ravikumar, P., and Risteski, A. (2022). Domain-adjusted regression or: ERM may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*. [Cited on page 3.]

- [51] Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. (2021). Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):215–246. [Cited on page 2.]
- [52] Rusak, E., Schneider, S., Pachitariu, G., Eck, L., Gehler, P. V., Bringmann, O., Brendel, W., and Bethge, M. (2022). If your data distribution shifts, use self-learning. *Transactions on Machine Learning Research*. [Cited on pages 2 and 6.]
- [53] Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks. In *International Conference on Learning Representations*. [Cited on page 29.]
- [54] Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5:197–227. [Cited on page 32.]
- [55] Schölkopf, B. (2022). Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 765–804. Association for Computing Machinery. [Cited on pages 2 and 26.]
- [56] Scott, C., Blanchard, G., and Handy, G. (2013). Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference on learning theory*, pages 489–511. PMLR. [Cited on page 32.]
- [57] Shi, Y., Seely, J., Torr, P., N, S., Hannun, A., Usunier, N., and Synnaeve, G. (2022a). Gradient matching for domain generalization. In *International Conference on Learning Representations*. [Cited on page 29.]
- [58] Shi, Y., Seely, J., Torr, P., Siddharth, N., Hannun, A., Usunier, N., and Synnaeve, G. (2022b). Gradient matching for domain generalization. In *International Conference on Learning Representations*. [Cited on pages 3, 8, 10, and 29.]
- [59] Silva Filho, T., Song, H., Perello-Nieto, M., Santos-Rodriguez, R., Kull, M., and Flach, P. (2023). Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning*, pages 1–50. [Cited on page 10.]
- [60] Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*. [Cited on page 32.]
- [61] Sugiyama, M. and Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press. [Cited on page 26.]
- [62] Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5). [Cited on page 26.]
- [63] Sun, Q., Murphy, K., Ebrahimi, S., and D’Amour, A. (2022). Beyond invariance: Test-time label-shift adaptation for distributions with "spurious" correlations. *arXiv preprint arXiv:2211.15646*. [Cited on page 2.]
- [64] Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560. [Cited on page 32.]
- [65] Vapnik, V. (1991). Principles of risk minimization for learning theory. *Advances in Neural Information Processing Systems*, 4. [Cited on pages 3 and 26.]
- [66] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New York, NY. [Cited on page 3.]
- [67] Veitch, V., D’Amour, A., Yadlowsky, S., and Eisenstein, J. (2021). Counterfactual invariance to spurious correlations: Why and how to pass stress tests. In *Advances in Neural Information Processing Systems*. [Cited on pages 2, 8, and 29.]
- [68] von Kügelgen, J., Mey, A., and Loog, M. (2019). Semi-generative modelling: Covariate-shift adaptation with cause and effect features. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1361–1369. PMLR. [Cited on pages 5, 10, 25, and 26.]

- [69] von Kügelgen, J., Mey, A., Loog, M., and Schölkopf, B. (2020). Semi-supervised learning, causality, and the conditional cluster assumption. In *Conference on Uncertainty in Artificial Intelligence*, pages 1–10. PMLR. [Cited on pages 5 and 26.]
- [70] Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. (2021). On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227. [Cited on pages 3, 10, and 29.]
- [71] Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021a). Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*. [Cited on page 2.]
- [72] Wang, D.-B., Feng, L., and Zhang, M.-L. (2021b). Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820. [Cited on page 10.]
- [73] Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Medicine*, 15(11). [Cited on page 3.]
- [74] Zhang, J., Lopez-Paz, D., and Bottou, L. (2022). Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*. [Cited on pages 3 and 31.]
- [75] Zhang, M., Lee, J., and Agarwal, S. (2021). Learning from noisy labels with no change to the training process. In *International Conference on Machine Learning*, pages 12468–12478. PMLR. [Cited on pages 6, 19, and 32.]
- [76] Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. (2019). On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR. [Cited on page 32.]
- [77] Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. (2018). Adversarial multiple source domain adaptation. *Advances in Neural Information Processing Systems*, 31. [Cited on page 32.]
- [78] Zheng, J. and Makar, M. (2022). Causally motivated multi-shortcut identification & removal. *Advances in Neural Information Processing Systems*. [Cited on pages 2, 8, and 29.]

### 5.3 Comments on the paper

**Doing better than a fixed performance-robustness trade-off.** In Chapter 4, we chose a fixed trade-off between performance and robustness at training time via the probability-of-generalisation parameter  $\alpha$ . This trade-off determined how much we used spurious but informative features, eventually discarding all such features as  $\alpha \rightarrow 1$ . In this paper, we showed that, rather than discarding such features at training time, we can change how we use them at test time *without labels* and still be provably robust.

**Exploiting new test-domain features without labels.** Extending the idea of using complementary features “in a new way”, we could in fact exploit a completely new set of features in the test domain that weren’t available in the training domains—so long as they satisfy complementarity. For example, if we leveraged data from multiple hospitals to learn a stable predictor of diabetes that only used causes (e.g., BMI, number of pregnancies, etc.), then, upon deployment to a new test hospital, we could exploit newly-available effect features without any labels (e.g., glucose levels).

**Weakening the complementarity condition.** Perhaps the most significant limitation of this work is the assumption of complementarity, i.e., that the stable and unstable features are conditionally independent given the label. While complementarity is implicit in the causal generative models assumed by prior works (Rojas-Carulla et al., 2018; von Kügelgen et al., 2019), it would be interesting to investigate whether or not weaker conditions suffice for SFB to succeed. One point of encouragement is that, in the related context of co-training, a similar condition was initially assumed and then weakened in subsequent work (Abney, 2002; Balcan et al., 2004; Blum and Mitchell, 1998; Wang and Zhou, 2010).

**Regularised feature-learning: a challenging optimisation problem.** To learn stable or invariant predictors, many works—including ours—add regularisation terms to the ERM objective (Arjovsky et al., 2019; Krueger et al., 2021). Unfortunately, in practice, the resulting optimisation problem can be significantly more challenging than ERM when using deep neural networks (Gulrajani and Lopez-Paz, 2020; Rosenfeld et al., 2022; Zhang et al., 2022). This is particularly true for SFB since it adds two different regularisation terms to the ERM objective (for stability and conditional independence). As a result, we found SFB to be particularly difficult to optimise compared to both

ERM and IRM. To address these difficulties, and make SFB easier to use in practice, it would be interesting to investigate:

1. **Pre-trained networks.** Recent works found that *ERM pretraining* can significantly boost the performance of more robust or regularised DG algorithms (see, e.g., Table 1 of Chapter 4 and Table 1 of Zhang et al. 2022). In addition, Zhang et al. (2022) proposed *Rich Feature Construction (RFC) pretraining* to extract a rich set of potentially-useful features, showing that this further stabilises the training of DG algorithms (see Figure 3 of their paper) and, as a result, leads to improved performance (see Table 3 of their paper).
2. **Pre-trained and frozen networks.** To further stabilise the training of DG algorithms when using deep networks, recent works have frozen all but the final linear layer after ERM or RFC pretraining (Rosenfeld et al., 2022; Zhang et al., 2022). While last-layer fine-tuning has long been used for transfer learning with deep networks (Girshick et al., 2014; Zeiler and Fergus, 2014), it has only recently been investigated for stabilising the training of DG algorithms. In particular, Rosenfeld et al. (2022) show that this stabilised training translates into reliable performance gains over ERM (see Table 1 of their paper), positing that: (i) previous observations on DomainBed (Gulrajani and Lopez-Paz, 2020), where no DG algorithm reliably beat ERM, were primarily due to more difficult optimisation problems<sup>1</sup>; and (ii) the current bottleneck for DG is not feature learning, since this can be done well with ERM or RFC, but rather robust regression.
3. **Separate networks for stable and unstable predictors.** Motivated by feature reuse and a fair comparison to baselines (in terms of the number of parameters), we used a single, shared feature extractor  $\Phi$  for our stable and unstable predictors. However, in light of the training instabilities that we encountered, it would be interesting to investigate completely-separate feature extractors  $\Phi_S$  and  $\Phi_U$ , choosing the network architectures carefully such that the total number of parameters matches that of single-network baselines. The hypothesis here is that separate feature extractors are easier to train as there is no competition or interference between the gradients.

---

<sup>1</sup>While we agree that this played a significant role, we believe that only comparing the *average* performance of these algorithms played just as significant a role, as discussed in § 4.3.





# 6

## Disentangled Representation Learning

This chapter focuses on representation learning, in particular, *disentangled representation learning*. One of the primary goals of representation learning is to learn representations of complex data that make it easier for downstream tasks to extract useful information (Bengio et al., 2013). In the context of this thesis, this can be viewed as an extreme setting for distribution shift in which the *task changes or shifts* at test time. With this view in mind, disentangled representation learning can be seen as preparing for an unknown test-time task by recovering and separating the data’s underlying factors of variation, discarding as little information as possible (Desjardins et al., 2012; Kulkarni et al., 2015). To better facilitate the learning and comparison of methods for disentangled representation learning, prior works have proposed evaluation frameworks for disentangled representations. We build on one such framework, that of Eastwood and Williams (2018), by first connecting it to identifiability and then extending it to contain new complementary measures of representation quality which better correlate with downstream performance.

Our main idea is that the *functional capacity required to use a representation* is an important but thus-far neglected aspect of representation quality, which we quantify using an explicitness or ease-of-use (E) score. In contrast to prior “mixing-based” measures of disentanglement, such as the D and C scores of the DCI framework (Eastwood and Williams, 2018), our E score directly measures a representation’s ease-of-use—often the most desirable property for representations.



## 6.1 Contribution

I led this project from conceptualisation to final form. In particular, I was heavily involved in coming up with the initial idea, formalising the extended framework, designing the experimental analyses, and writing the manuscript. In addition to helping with these tasks, Andrei Liviu Nicolicioiu ran the experiments while Julius von Kügelgen led the formal connection to identifiability. Both share first authorship.

## 6.2 Paper

# DCI-ES: AN EXTENDED DISENTANGLEMENT FRAMEWORK WITH CONNECTIONS TO IDENTIFIABILITY

Cian Eastwood<sup>\*1,2</sup>, Andrei Liviu Nicolicioiu<sup>\*1</sup>, Julius von Kügelgen<sup>\*1,3</sup>,  
Armin Kekić<sup>1</sup>, Frederik Träuble<sup>1</sup>, Andrea Dittadi<sup>1,4</sup>, and Bernhard Schölkopf<sup>1</sup>

<sup>1</sup>Max Planck Institute for Intelligent Systems, Tübingen, Germany

<sup>2</sup>School of Informatics, University of Edinburgh

<sup>3</sup>Department of Engineering, University of Cambridge

<sup>4</sup>Technical University of Denmark

## ABSTRACT

In representation learning, a common approach is to seek representations which disentangle the underlying factors of variation. Eastwood & Williams (2018) proposed three metrics for quantifying the quality of such disentangled representations: disentanglement (D), completeness (C) and informativeness (I). In this work, we first connect this DCI framework to two common notions of linear and nonlinear identifiability, thereby establishing a formal link between disentanglement and the closely-related field of independent component analysis. We then propose an extended DCI-ES framework with two new measures of representation quality—*explicitness* (E) and *size* (S)—and point out how D and C can be computed for black-box predictors. Our main idea is that the *functional capacity required to use a representation* is an important but thus-far neglected aspect of representation quality, which we quantify using explicitness or *ease-of-use* (E). We illustrate the relevance of our extensions on the MPI3D and Cars3D datasets.

## 1 INTRODUCTION

A primary goal of representation learning is to learn representations  $r(\mathbf{x})$  of complex data  $\mathbf{x}$  that “make it easier to extract useful information when building classifiers or other predictors” (Bengio et al., 2013). *Disentangled* representations, which aim to recover and separate (or, more formally, *identify*) the underlying factors of variation  $\mathbf{z}$  that generate the data as  $\mathbf{x} = g(\mathbf{z})$ , are a promising step in this direction. In particular, it has been argued that such representations are not only interpretable (Kulkarni et al., 2015; Chen et al., 2016) but also make it easier to extract useful information for downstream tasks by recombining previously-learned factors in novel ways (Lake et al., 2017).

While there is no single, widely-accepted definition, many evaluation protocols have been proposed to capture different notions of disentanglement based on the relationship between the learnt representation or *code*  $\mathbf{c} = r(\mathbf{x})$  and the ground-truth data-generative factors  $\mathbf{z}$  (Higgins et al., 2017; Eastwood & Williams, 2018; Ridgeway & Mozer, 2018; Kim & Mnih, 2018; Chen et al., 2018; Suter et al., 2019; Shu et al., 2020). In particular, the metrics of Eastwood & Williams (2018)—*disentanglement* (D), *completeness* (C) and *informativeness* (I)—estimate this relationship by learning a *probe*  $f$  to predict  $\mathbf{z}$  from  $\mathbf{c}$  and can be used to relate many other notions of disentanglement (see Locatello et al. 2020, § 6).

In this work, we extend this DCI framework in several ways. Our main idea is that *the functional capacity required to recover  $\mathbf{z}$  from  $\mathbf{c}$  is an important but thus-far neglected aspect of representation quality*. For example, consider the case of recovering  $\mathbf{z}$  from: (i) a noisy version thereof; (ii) raw, high-dimensional data (e.g. images); and (iii) a linearly-mixed version thereof, with each  $c_i$  containing the same amount of information about each  $z_j$  (precise definition in § 6.1). The noisy version (i) will do quite well with just linear capacity, but is fundamentally limited by the noise corruption; the raw data (ii) will likely do quite poorly with linear capacity, but eventually outperform (i) given sufficient capacity; and the linearly-mixed version (iii) will perfectly recover  $\mathbf{z}$  with just linear capacity, yet achieve the worst-possible disentanglement score of  $D = 0$ . Motivated by this observation, we introduce a measure of *explicitness* or *ease-of-use* based a representation’s *loss-capacity curve* (see Fig. 1).

<sup>\*</sup>Equal contribution.

Published as a conference paper at ICLR 2023

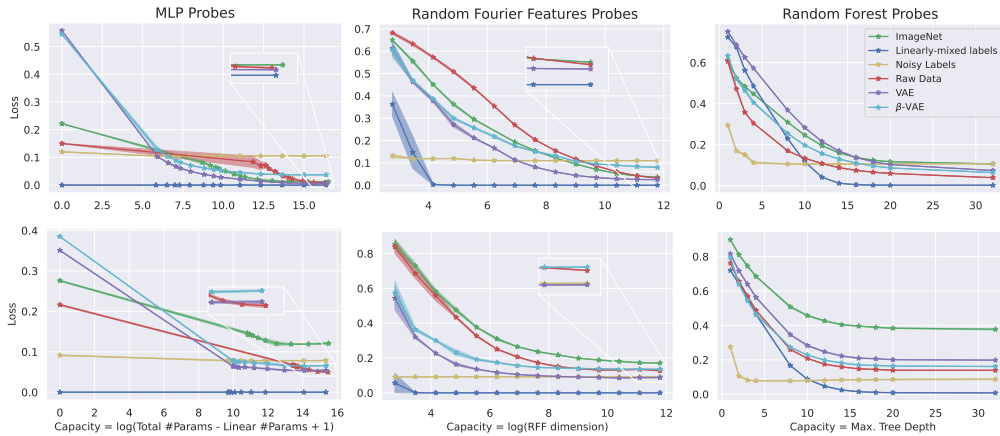


Figure 1: **Loss-capacity curves.** Empirical loss-capacity curves (see § 4.1) for various representations (see legend), datasets (top: MPI3D-Real, bottom: Cars3D), and probe types (left: multi-layer perceptrons / MLPs, middle: Random Fourier Features / RFFs, right: Random Forests / RFs). The loss was first averaged over factors  $z_j$ , and then means and 95% confidence intervals were computed over 3 random seeds. Details in § 6.

**Structure and contributions.** First, we connect the DCI metrics to two common notions of linear and nonlinear identifiability (§ 3). Next, we propose an extended DCI-ES framework (§ 4) in which we: (i) introduce two new complementary measures of representation quality—*explicitness* (E), derived from a representation’s *loss-capacity curve*, and *size* (S); and then (ii) elucidate a means to compute the D and C scores for arbitrary black-box probes (e.g., MLPs). Finally, in our experiments (§ 6), we use our extended framework to compare different representations on the MPI3D-Real (Gondal et al., 2019) and Cars3D (Reed et al., 2015) datasets, illustrating the practical usefulness of our E score through its strong correlation with downstream performance.

## 2 BACKGROUND

Given a synthetic dataset of observations  $x = g(z)$  along with the corresponding  $K$ -dimensional data-generating factors  $z \in \mathbb{R}^K$ , the DCI framework quantitatively evaluates an  $L$ -dimensional data representation or *code*  $c = r(x) \in \mathbb{R}^L$  using two steps: (i) train a probe  $f$  to predict  $z$  from  $c$ , i.e.,  $\hat{z} = f(c) = f(r(x)) = f(r(g(z)))$ ; and then (ii) quantify  $f$ ’s prediction error and its deviation from the ideal one-to-one mapping, namely a permutation matrix (with extra “dead” units in  $c$  whenever  $L > K$ ).<sup>1</sup> For step (i), Eastwood & Williams (2018) use Lasso (Tibshirani, 1996) or Random Forests (RFs, Breiman 2001) as linear or nonlinear predictors, respectively, for which it is straightforward to read-off suitable “relative feature importances”.

**Definition 2.1.**  $R \in \mathbb{R}^{L \times K}$  is a matrix of relative importances for predicting  $z$  from  $c$  via  $\hat{z} = f(c)$  if  $R_{ij}$  captures some notion of the contribution of  $c_i$  to predicting  $z_j$  s.t.  $\forall i, j: R_{ij} \geq 0$  and  $\sum_{i=1}^L R_{ij} = 1$ .

For step (ii), Eastwood & Williams use  $R$  and the prediction error to define and quantify three desiderata of disentangled representations: *disentanglement* (D), *completeness* (C), and *informativeness* (I).

**Disentanglement.** Disentanglement (D) measures the average number of data-generating factors  $z_j$  that are captured by any single code  $c_i$ . The score  $D_i$  is given by  $D_i = 1 - H_K(P_i)$ , where  $H_K(P_i) = -\sum_{k=1}^K P_{ik} \log_K P_{ik}$  denotes the entropy of the distribution  $P_i$  over *row*  $i$  of  $R$ , with  $P_{ij} = R_{ij} / \sum_{k=1}^K R_{ik}$ . If  $c_i$  is only important for predicting a single  $z_j$ , we get a perfect score of  $D_i = 1$ . If  $c_i$  is equally important for predicting all  $z_j$  (for  $j = 1, \dots, K$ ), we get the worst score of  $D_i = 0$ . The overall score  $D$  is then given by the weighted average  $D = \sum_{i=1}^L \rho_i D_i$ , with  $\rho_i = \frac{1}{K} \sum_{k=1}^K R_{ik}$ .

**Completeness.** Completeness (C) measures the average number of code variables  $c_i$  required to capture any single  $z_j$ ; it has also been called *compactness* (Ridgeway & Mozer, 2018). The score  $C_j$  in capturing  $z_j$  is given by  $C_j = (1 - H_L(\tilde{P}_j))$ , where  $H_L(\tilde{P}_j) = -\sum_{\ell=1}^L \tilde{P}_{\ell j} \log_L \tilde{P}_{\ell j}$  denotes the

<sup>1</sup>W.l.o.g., it can be assumed that  $z_i$  and  $c_j$  are normalised to have mean zero and variance one for all  $i, j$ , for otherwise such normalisation can be “absorbed” into  $g(\cdot)$  and  $r(\cdot)$ .

entropy of the distribution  $\tilde{P}_j$  over *column*  $j$  of  $\mathbf{R}$ , with  $\tilde{P}_{ij} = R_{ij}$ . If a single  $c_i$  contributes to  $z_j$ 's prediction, we get a perfect score of  $C_j = 1$ . If all  $c_i$  equally contribute to  $z_j$ 's prediction (for  $i = 1, \dots, L$ ), we get the worst score of  $C_j = 0$ . The overall completeness score is given by  $C = \frac{1}{K} \sum_{j=1}^K C_j$ .

**Remark 2.2.** Together,  $D$  and  $C$  quantify the degree of ‘‘mixing’’ between  $c$  and  $z$ , i.e., the deviation from a one-to-one mapping. They are reported separately as they capture distinct criteria.

**Informativeness.** The informativeness ( $I$ ) of representation  $c$  about data-generative factor  $z_j$  is quantified by the prediction error, i.e.,  $I_j = 1 - \mathbb{E}[\ell(z_j, f_j(c))]$ , where  $\ell$  is an appropriate loss function.<sup>2</sup> Note that  $I_j$  depends on the capacity of  $f_j$ , as depicted in Fig. 1. Thus, for  $I_j$  to accurately capture the informativeness of  $c$  about  $z_j$ ,  $f_j$  must have sufficient capacity to extract *all* of the information in  $c$  about  $z_j$ . This capacity-informativeness dependency motivates a separate measure of representation *explicitness* in § 4.1. The overall informativeness score is given by  $I = \frac{1}{K} \sum_{j=1}^K I_j$ .

### 3 CONNECTION TO IDENTIFIABILITY

The goal of learning a data representation which recovers the underlying data-generating factors is closely related to blind source separation and independent component analysis (ICA, Comon 1994; Hyvärinen & Pajunen 1999; Hyvärinen et al. 2019). Whether a given learning algorithm provably achieves this goal up to acceptable ambiguities, subject to certain assumptions on the data-generating process, is typically formalised using the notion of *identifiability*. Two common types of identifiability for linear and nonlinear settings, respectively, are the following.

**Definition 3.1.** We say that  $c = r(\mathbf{x}) = r(g(\mathbf{z}))$  identifies  $\mathbf{z}$  up to sign and permutation if  $c = \mathbf{P}\mathbf{z}$  for some signed permutation matrix  $\mathbf{P}$  (i.e.,  $|\mathbf{P}|$  is a permutation).

**Definition 3.2.** We say  $c$  identifies  $\mathbf{z}$  up to permutation and element-wise reparametrisation if there exists a permutation  $\pi$  of  $\{1, \dots, K\}$  and invertible scalar-functions  $\{h_k\}_{k=1}^K$  s.t.  $\forall j: c_j = h_j(z_{\pi(j)})$ .

We now establish theoretical connections between the DCI framework and these identifiability types.

**Proposition 3.3.** If  $D = C = 1$  and  $K = L$  (i.e.,  $\dim(c) = \dim(\mathbf{z})$ ), then  $\mathbf{R}$  is a permutation matrix.

All proofs are provided in Appendix A. Using Prop. 3.3, we can establish links to identifiability, provided the inferred representation  $c$  perfectly predicts the true data-generating factors  $\mathbf{z}$ , i.e.,  $I = 1$ .

**Corollary 3.4.** Under the same conditions as Prop. 3.3, if  $\mathbf{z} = \mathbf{W}^\top c$  (so that  $I = 1$ ) for some  $\mathbf{W}$  with  $R_{ij} = \frac{|w_{ij}|}{\sum_{i=1}^L |w_{ij}|}$ , then  $c$  identifies  $\mathbf{z}$  up to permutation and sign (Defn. 3.1).

For nonlinear  $f$ , we give a more general statement for suitably-chosen feature-importance matrices  $\mathbf{R}$ .

**Corollary 3.5.** Under the same conditions as Prop. 3.3, let  $\mathbf{z} = f(c)$  (so that  $I = 1$ ) with  $f$  an invertible and differentiable nonlinear function, and let  $\mathbf{R}$  be a matrix of relative feature importances for  $f$  (Defn. 2.1) with the property that  $R_{ij} = 0$  if and only if  $f_j$  does not depend on  $c_i$ , i.e.,  $\|\partial_i f_j\|_2 = 0$ . Then  $c$  identifies  $\mathbf{z}$  up to permutation and element-wise reparametrisation (Defn. 3.2).

**Remark 3.6.** While the *if* part of Corollary 3.5 holds for most feature importance measures, the *only if* part, in general, does not: not using a feature  $c_i$  is typically a *sufficient* condition for  $R_{ij} = 0$ , but it need not be a *necessary* condition (as required for Corollary 3.5). E.g., measures based on *average* performance may not satisfy this since a feature may not contribute on average, but still be used—sometimes helping and sometimes hurting performance (see § 7 for further discussion). In contrast, Gini importances, as used in random forests, *do* satisfy the necessary condition. While the non-invertibility of random forests prevents an explicit link to identifiability (typically studied for continuous features), they can still be a principled choice in practice (where features are often categorical).

**Summary.** We have established that the learnt representation  $c$  identifies the ground-truth  $\mathbf{z}$  up to:

- sign and permutation if  $D = C = I = 1$  and  $f$  is linear;
- permutation and element-wise reparametrisation if  $D = C = I = 1$  and  $R_{ij} = 0 \Leftrightarrow \|\partial_i f_j\|_2 = 0$ .

<sup>2</sup>Here we deviate from Eastwood & Williams (who had  $I_j = \mathbb{E}[\ell(z_j, f_j(c))]$ ) such that 1 is now the best score.

## 4 EXTENDED DCI-ES FRAMEWORK

Motivated by our theoretical insights from § 3—considering different probe function classes provides links to different types of identifiability—and the empirically-observed performance differences between representations trained with different-capacity probes shown in Fig. 1, we now propose several extensions of the DCI framework.

### 4.1 EXPLICITNESS (E)

We first introduce a new complementary notion of disentanglement based on the functional capacity required to recover or predict  $z$  from  $c$ . The key idea is to measure the *explicitness* or *ease-of-use* (E) of a representation using its *loss-capacity curve*.

**Notation.** Let  $\mathcal{F}$  be a probe function class (e.g., MLPs or RFs), let  $f_j^* \in \arg \min_{f \in \mathcal{F}} \mathbb{E}[\ell(z_j, f(c))]$  be a minimum-loss probe for factor  $z_j$  on a held-out data split<sup>3</sup>, and let  $\text{Cap}(\cdot)$  be a suitable capacity measure on  $\mathcal{F}$ —e.g., for RFs,  $\text{Cap}(f)$  could correspond to the maximum tree-depth of  $f$ .

**Loss-capacity curves.** A loss-capacity curve for representation  $c$ , factor  $z_j$ , and probe class  $\mathcal{F}$  displays test-set loss against probe capacity for increasing-capacity probes  $f \in \mathcal{F}$  (see Fig. 1). To plot such a curve, we must train  $T$  predictors with capacities  $\kappa_1, \dots, \kappa_T$  to predict  $z_j$ , with

$$f_j^t \in \arg \min_{f \in \mathcal{F}} \mathbb{E}[\ell(z_j, f(c))] \quad \text{s.t.} \quad \text{Cap}(f) = \kappa_t. \quad (4.1)$$

Here  $\kappa_1, \dots, \kappa_T$  is a list of  $T$  increasing probe *capacities*, ideally<sup>4</sup> shared by all representations, with suitable choices for  $\kappa_1$  and  $\kappa_T$  depending on both  $\mathcal{F}$  and the dataset. For example, we may choose  $\kappa_T$  to be large enough for all representations to achieve their lowest loss and, for random forest  $f$ s, we may choose an initial tree depth of  $\kappa_1 = 1$  and then  $T - 2$  tree depths between 1 and  $\kappa_T$ .

**AULCC.** We next define the *Area Under the Loss-Capacity Curve* (AULCC) for representation  $c$ , factor  $z_j$ , and probe class  $\mathcal{F}$  as the (approximate) area between the corresponding loss-capacity curve and the loss-line of our best predictor  $\ell_j^{*,c} = \mathbb{E}[\ell(z_j, f_j^*(c))]$ . To compute this area, depicted in Fig. 2, we use the trapezoidal rule

$$\text{AULCC}(z_j, c; \mathcal{F}) = \sum_{t=2}^{t^{*,c}} \left( \frac{1}{2} \left( \ell_j^{t-1,c} + \ell_j^{t,c} \right) - \ell_j^{*,c} \right) \cdot \Delta \kappa_t,$$

where  $t^{*,c}$  denotes the index of  $c$ 's lowest-loss capacity  $\kappa_{*,c}$ ;  $\ell_j^{t,c} = \mathbb{E}[\ell(z_j, f_j^t(c))]$  the test-set loss with predictor  $f_j^t$ , see Eq. (4.1); and  $\Delta \kappa_t = \kappa_t - \kappa_{t-1}$  the size of the capacity interval at step  $t$ . If the lowest loss is achieved at the lowest capacity, i.e.  $t^{*,c} = 1$ , we set  $\text{AULCC} = 0$ .

**Explicitness.** We define the **explicitness** (E) of representation  $c$  for predicting factor  $z_j$  with predictor class  $\mathcal{F}$  as

$$E(z_j, c; \mathcal{F}) = 1 - \frac{\text{AULCC}(z_j, c; \mathcal{F})}{\frac{1}{2}(\kappa_T - \kappa_1)(\ell_j^b - \ell_j^*)},$$

where  $\ell_j^b$  is a baseline loss (e.g., that of  $\mathbb{E}[z_j]$ ) and  $\ell_j^*$  a lowest possible loss (e.g., 0) for  $\mathcal{F}$ . Here, the denominator represents the area of the light-blue triangle in Fig. 2, *normalizing* the AULCC such that  $E_j \in [-1, 1]$  so long as  $\ell_j^* < \ell_j^b$ . The best score  $E_j = 1$  means that the best loss was achieved with the lowest-capacity probe  $f_j^1$ , i.e.,  $\ell_j^{*,c} = \ell_j^{1,c}$  and  $\kappa_{*,c} = \kappa_1$ , and thus our representation  $c$  was explicit or easy-to-use for predicting  $z_j$  with  $f \in \mathcal{F}$  since there was *no surplus capacity required*

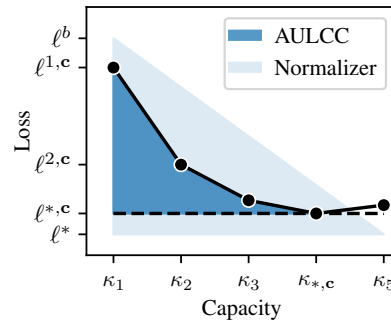


Figure 2: **Explicitness via the area under the loss-capacity curve (AULCC).** Here,  $\kappa_1, \dots, \kappa_T$  (x-axis) are a sequence of increasing function-capacities and  $\ell^{1,c}, \dots, \ell^{T,c}$  (y-axis) are the losses achieved by the corresponding optimal predictors for  $c$ . The lowest loss  $\ell^{*,c}$  is achieved at capacity  $\kappa_{*,c}$ , while  $\ell^b$  and  $\ell^*$  are baseline and best-possible losses for the probe class.

<sup>3</sup>In practice, all expectations are taken w.r.t. the corresponding empirical (train/validation/test) distributions.

<sup>4</sup>True for RFs but not input-size dependent MLPs (see § 6).

(beyond  $\kappa_1$ ) to achieve our lowest loss. In contrast,  $E_j = 0$  means that AULCC = Normalizer in Fig. 2, i.e., that the loss decreased at a linear rate from  $\ell_j^b$  to  $\ell_j^*$  with increased probe capacity. More generally, if  $\ell^{*,c} = \ell^*$ , i.e., the lowest loss for  $\mathcal{F}$  can be reached with representation  $c$ , then  $E_j < 0$  implies that the loss decreased at a *sub-linear rate* with increased capacity, while  $E_j > 0$  implies it decreased at a *super-linear rate*. The overall explicitness score is given by  $E = \frac{1}{K} \sum_{j=1}^K E_j$ .

**E vs. I.** While the informativeness score  $I_j$  captures the (total) amount of information in  $c$  about  $z_j$ , the explicitness score  $E_j$  captures the *ease-of-use* of this information. In particular, while  $I_j$  is quantified by the *lowest prediction error with any capacity*  $\ell^{*,c}$ , corresponding to a single point on  $c$ 's loss-capacity curve,  $E_j$  is quantified by the *area under this curve*.

**A fine-grained picture of identifiability.** Compared to the commonly-used mean correlation coefficient (MCC) or Amari distance (Amari et al., 1996; Yang & Amari, 1997), the  $D, C, I, E$  scores represent empirical measures which: (i) easily extend to mismatches in dimensionalities, i.e.,  $L > K$ ; and (ii) provide a more fine-grained picture of identifiability (violations), for if the initial probe capacity  $\kappa_1$  is linear and  $R$  satisfies Corollary 3.5, we have that:

- $D=C=I=E=1 \implies$  identified up to sign and permutation (Defn. 3.1);
- $D=C=I=1 \implies$  identified up to permutation and element-wise reparametrisation (Defn. 3.2);
- $I=E=1 \implies$  identified up to invertible linear transformation (cf. Khemakhem et al., 2020).

Thus, if  $D=C=I=E=1$  does not hold exactly, which score deviates the most from 1 may provide valuable insight into the type of identifiability violation.

**Probe classes.** As emphasized above, whether or not a representation  $c$  is explicit or easy-to-use for predicting factor  $z_j$  depends on the class of probe  $\mathcal{F}$  used, e.g., MLPs or RFs. More generally, the explicitness of a representation depends on the way in which it is used in downstream applications, with different downstream uses or probe classes resulting in different definitions of explicit or easy-to-use information. We thus conduct experiments with different probe classes in § 6.

#### 4.2 SIZE (S)

We next introduce a measure of representation size (S), motivated by the observation that larger representations tend to be both more informative and more explicit (see Tab. 1, more details below). Reporting S thus allows size-informativeness and size-explicitness trade-offs to be analysed.

**A measure of size.** We measure representation *size* (S) relative to the ground-truth as:

$$S = \frac{K}{L} = \frac{\dim(\mathbf{z})}{\dim(\mathbf{c})}.$$

When  $L \geq K$ , as often the case, we have  $S \in (0, 1]$  with the perfect score being  $S = 1$ . However, if we also consider the  $L < K$  case, which would likely sacrifice some informativeness, we have  $S \in (1, K]$ .

**Larger representations are often more informative.** When  $L < K$ , it is intuitive that larger representations are more informative—they can simply preserve more information about  $\mathbf{z}$ . When  $L > K$ , however, it is also common for larger representations to be more informative, perhaps due to an easier optimization landscape (Frankle & Carbin, 2019; Golubeva et al., 2021). Tab. 1 illustrates this point, where AE-5 denotes an autoencoder with  $L = 5$ . Note that  $K = 7$  for MPI3D-Real (see § 6).

**Larger representations are often more explicit.** The explicitness of a representation also depends on its size: larger representations tend to be more explicit, as is apparent from the second column of Tab. 1. To explain this, we plot the corresponding loss-capacity curves in Fig. 3. Here we see that the increased explicitness (i.e., smaller AULLC) of larger representations stems from a substantially lower initial loss when using a linear-capacity MLP probe. The fact that larger representations perform better with linear-capacity MLPs is unsurprising since they have more parameters.

#### 4.3 PROBE-AGNOSTIC FEATURE IMPORTANCES

Finally, to meaningfully discuss more flexible probe-function choices within the DCI-ES framework, we point out that the D and C scores can be computed for arbitrary black-box probes  $f$  by using *probe-agnostic* feature-importance measures. In particular, in our experiments (§ 6), we use SAGE (Covert et al., 2020) which summarises each feature's importance based on its contribution to



Published as a conference paper at ICLR 2023

Representation	I	E	S
AE-5	0.75	0.74	1.4
AE-7	0.92	0.71	1.0
AE-10	0.99	0.72	0.7
AE-100	1.0	0.90	0.07
AE-500	1.0	0.93	0.01

Table 1: I, E and S scores for auto-encoders of various sizes on MPI3D-Real with MLP probes.

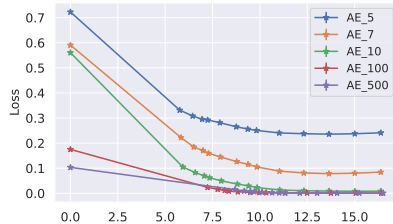


Figure 3: Loss-capacity curves for auto-encoders of various sizes on MPI3D-Real with MLP probes.

predictive performance, making use of Shapley values (Shapley, 1953) to account for complex feature interactions. Such probe-agnostic measures allow the  $D$  and  $C$  scores to be computed for probes with no inherent or built-in notion of feature importance (e.g., MLPs), thereby generalising the Lasso and RF examples of Eastwood & Williams (2018, § 4.3). While SAGE has several practical advantages over other probe-agnostic methods (see, e.g., Covert et al., 2020, Table 1), it may not satisfy the conditions required to link the  $D$  and  $C$  scores to different identifiability equivalence classes (see Remark 3.6). Future work may explore alternative methods which do, e.g., by looking at a feature’s mean *absolute* attribution value (Lundberg & Lee, 2017) since, intuitively, absolute contributions do not allow for a cancellation of positive and negative attribution on average (cf. Remark 3.6).

## 5 RELATED WORK

**Explicit representations.** Eastwood & Williams (2018, § 2) noted that the informativeness score with a linear probe quantifies the amount of information in  $c$  about  $z$  that is “explicitly represented”, while Ridgeway & Mozer (2018, § 3) proposed a measure of “explicitness” which simply reports the informativeness score with a linear probe. In contrast, our DCI-ES framework differentiates between the amount of information in  $c$  about  $z$  (*informativeness*) and the ease-of-use of this information (*explicitness*). This allows a more fine-grained analysis of the relationship between  $c$  and  $z$ , both theoretically (distinguishing between more identifiability equivalence classes; § 3) and empirically (§ 6).

**Loss-capacity curves.** Plotting loss against model complexity or capacity has long been used in statistical learning theory, e.g., for studying the bias-variance trade-off (Hastie et al., 2009, Fig. 7.1). More recently, such loss-capacity curves have been used to study the double-descent phenomenon of neural networks (Belkin et al., 2019; Nakkiran et al., 2021) as well as the scaling laws of large language models (Kaplan et al., 2020). However, they have yet to be used for assessing the quality or explicitness of representations.

**Loss-data curves.** Whitney et al. (2020) use loss-data curves, which plot loss against dataset size, to assess representations. They measure the quality of a representation by the *sample complexity* of learning probes that achieve low loss on a task of interest. Loss-data curves are also studied under the term *learning curves* in standard/purely supervised-learning settings (see, e.g., Viering & Loog, 2021, for a recent review). In contrast, we focus on *functional complexity* and the task of predicting the data-generative factors  $z$ , and then discuss the functional complexity for other tasks  $y$  in § 7.

## 6 EXPERIMENTS

### 6.1 SETUP

**Data.** We perform our analysis of loss-capacity curves on the MPI3D-Real (Gondal et al., 2019) and Cars3D (Reed et al., 2015) datasets. MPI3D-Real contains  $\approx 1M$  real-world images of a robotic arm holding different objects with seven annotated ground-truth factors: object colour (6), object shape (6), object size (2), camera height (3), background colour (3) and two degrees of rotations of the arm ( $40 \times 40$ ); numbers in brackets indicate the number of possible values for each factor. Cars3D contains  $\approx 17.5k$  rendered images of cars with three annotated ground-truth factors: camera elevation (4), azimuth (24) and car type (183).

**Representations.** We use the following synthetic baselines and standard models as representations:

- *Noisy labels*:  $\mathbf{c} = \mathbf{z} + \boldsymbol{\epsilon}$ , with  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.01 \cdot \mathbf{I}_K)$ .
- *Linearly-mixed labels*:  $\mathbf{c} = \mathbf{W}\mathbf{z}$ , with  $W_{ij} = \frac{1}{LK} + \epsilon_{ij}$  and  $\epsilon_{ij} \sim \mathcal{N}(0, 0.001)$  to achieve “uniform mixing” (each  $z_j$  evenly-distributed across the  $c_i$ s) while also ensuring the invertibility of  $\mathbf{W}$  a.s.
- *Raw data (pixels)*:  $\mathbf{c} = \mathbf{x} = g(\mathbf{z})$ .
- *Others*: We also use VAEs (Kingma & Welling, 2014) with 10 latents ( $L=10$ ),  $\beta$ -VAEs (Higgins et al. 2017,  $L=10$ ); and an ImageNet-pretrained ResNet18 (He et al. 2016,  $L=512$ ).

**Probes.** We use MLPs, RFs and Random Fourier Features (RFFs, Rahimi & Recht 2007) to predict  $\mathbf{z}$  from  $\mathbf{c}$ , with RFFs having a linear classifier on top. For MLPs, we start with linear probes (no hidden layers) then increase capacity by adding two hidden layers and varying their widths from  $2 \times K$  to  $512 \times K$ . We then measure capacity based on the number of “extra” parameters beyond that of the linear probe, and compute feature importances using SAGE with permutation-sampling estimators and marginal sampling of masked values (see <https://github.com/iancovert/sage>). For RFs, we use ensembles of 100 trees, control capacity by varying the maximum depth between 1 and 32, and compute feature importances using Gini importance. For RFFs, we control capacity by exponentially increasing the number of random features from  $2^4$  to  $2^{17}$ , and compute feature importances using SAGE.

**Implementation details.** We split the data into training, validation and test sets of size 295k, 16k, and 726k respectively for `MPI3D-Real` and 12.6k, 1.4k, 3.4k for `Cars3d`. We use the validation split for hyperparameter selection and report results on the test split. We train MLP probes using the Adam (Kingma & Ba, 2015) optimizer for 100 epochs. We use mean-square error and cross-entropy losses for continuous and discrete factors  $z_j$ , respectively. To compute  $\bar{E}_j$ , we use the baseline losses of  $\mathbb{E}[z_j]$  and a random classifier for continuous and discrete  $z_j$ , respectively. Further details can be found in our open-source code: <https://github.com/andreinicolicioiu/DCI-ES>.

## 6.2 EVALUATION RESULTS: CURVES AND SCORES

**Loss-capacity curves.** Fig. 1 depicts loss-capacity curves for the three probes and two datasets, averaged over factors  $z_j$ . In all six plots, the noisy-labels baseline performs well with low-capacity and then is surpassed by other representations given sufficient capacity, as expected. Note that the linearly-mixed-labels baseline immediately achieves  $\approx 0$  loss with MLP probes but not with RFF or RF probes, supporting the idea that the explicitness or ease-of-use of a representation depends on the way in which it is used. Also note that, with MLP probes and  $\log(\text{excess \#params})$  as the capacity measure, larger input representations are afforded more parameters with a linear probe and thus are more expressive. This further explains why larger representations are often more explicit, and highlights the difficulty of measuring the capacity of MLPs—an active area of research in its own right, which we discuss in § 7. Finally, in Appendix B.2, we investigate the effect of dataset size by plotting loss-capacity curves for different dataset sizes, observing that larger datasets have smaller performance gaps between: (i) synthetic and learned representations; and (ii) small and large representations (see Fig. 10).

**DCI-ES scores.** Tab. 2 reports the corresponding DCI-ES scores, along with some oracle scores for MLPs. Note that: (i) the GT labels  $\mathbf{z}$  get perfect scores of 1 for all metrics; (ii) by attaining very low D and C scores but near-perfect E scores, the linearly-mixed labels expose the key difference between mixing-based (D,C) and functional-capacity-based (E) measures of the *simplicity of the  $\mathbf{c}$ - $\mathbf{z}$  relationship*; (iii) larger representations (ImageNet-pretr, raw data) tend to be more explicit than smaller ones (VAE,  $\beta$ -VAE), with  $S$  and  $E$  together capturing this size-explicitness trade-off; and (iv)  $\beta$ -VAE achieves better mixing-based scores (D,C) but similar E scores compared to the VAE, illustrating that these two “disentanglement” notions are indeed orthogonal and complementary.

## 6.3 DOWNSTREAM RESULTS: SCORE CORRELATIONS

**Setup.** To illustrate the practical usefulness of our explicitness score, we calculate its correlation with downstream performance when using low-capacity probes. Using `MPI-3D`, we create 14 synthetic downstream tasks: 7 regression tasks with  $y^i = M^i \mathbf{z}$  and  $M_{jk}^i \sim U(0, 1)$ , and 7 classification tasks with  $y^i = \mathbb{1}_{\{z_i > m_i\}}$  and  $m_i$  the median value of factor  $z_i$ . For representations, we use AEs, VAEs and  $\beta$ -VAEs, 2 latent dimensionalities (i.e.  $\dim(\mathbf{c})$ ) of 10 and 50, and 5 random seeds—resulting in a total of 30 different representations  $\mathbf{c}$ . To compute the correlations, we first compute the DCIE scores as before, training MLP and RF probes  $f$  to predict  $\mathbf{z}$  from  $\mathbf{c}$ , i.e.  $\hat{z}_j = f_j(\mathbf{c})$ , and then compute the down-



Published as a conference paper at ICLR 2023

Table 2: **DCI-ES scores for different probes, datasets and representations.** Empirical scores using MLP, RFF and RF probes trained on the MPI3D-Real and Cars3D datasets, as well as theoretical/oracle scores for some simple representations with MLPs (MLP\*). We show averages over 3 random seeds; standard deviations were all  $< 0.05$ . Note that which representation is deemed “best” depends on the application of interest—some are more disentangled, some more informative, some more explicit, etc.

Representation	Probe	MPI3D					CARS3D				
		D	C	I	E	S	D	C	I	E	S
GT Labels $z$	MLP*	1	1	1	1	1	1	1	1	1	1
Noisy labels	MLP*	1	1	0.9	1	1.0	1	1	0.9	1	1.0
	MLP	0.97	0.97	0.89	0.99	1.0	0.99	0.99	0.92	0.99	1.0
	RFF	0.97	0.97	0.88	0.99	1.0	1.0	1.0	0.91	1.0	1.0
	RF	0.93	0.93	0.89	0.98	1.0	0.95	0.95	0.92	0.99	1.0
Linearly-mixed labels	MLP*	0	0	1	1	1.0	0	0	1	1	1.0
	MLP	0.13	0.22	1.0	1.0	1.0	0.21	0.22	1.0	1.0	1.0
	RFF	0.11	0.21	1.0	0.94	1.0	0.19	0.19	1.0	1.0	1.0
	RF	0.17	0.21	1.0	0.72	1.0	0.08	0.12	0.99	0.78	1.0
VAE	MLP	0.15	0.14	0.99	0.71	0.7	0.18	0.11	0.95	0.80	0.3
	RFF	0.13	0.14	0.97	0.69	0.7	0.16	0.11	0.91	0.87	0.3
	RF	0.10	0.10	0.93	0.65	0.7	0.14	0.09	0.80	0.81	0.3
$\beta$ -VAE	MLP	0.46	0.41	0.96	0.74	0.7	0.27	0.23	0.94	0.78	0.3
	RFF	0.41	0.38	0.92	0.71	0.7	0.31	0.23	0.86	0.86	0.3
	RF	0.39	0.35	0.94	0.76	0.7	0.20	0.17	0.84	0.83	0.3
ImgNet-pretr	MLP	0.16	0.10	0.99	0.82	0.01	0.22	0.07	0.88	0.86	0.006
	RFF	0.15	0.13	0.96	0.58	0.01	0.24	0.10	0.83	0.65	0.006
	RF	0.35	0.20	0.89	0.78	0.01	0.20	0.09	0.62	0.83	0.006
Raw data	MLP	0.22	0.16	0.99	0.82	0.001	0.39	0.27	0.95	0.84	0.0002
	RFF	0.37	0.14	0.97	0.44	0.001	0.32	0.24	0.87	0.64	0.0002
	RF	0.84	0.41	0.96	0.80	0.001	0.53	0.31	0.86	0.82	0.0002

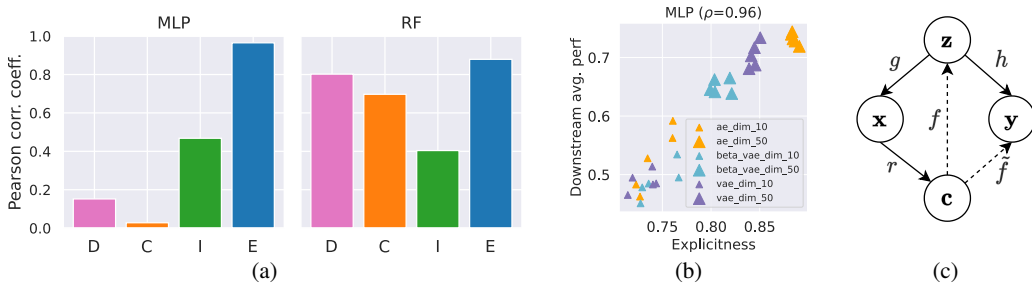


Figure 4: (a) Correlation coefficients  $\rho$  between DCIE scores and downstream performance with low-capacity probes. (b) E vs. downstream performance with linear MLPs. (c) DCIE scores are computed by predicting  $z$  from  $c$  with probes  $f$ , then downstream tasks  $y = h(z)$  are solved by predicting  $y$  from  $c$  with low-capacity probes  $\tilde{f}$ .

stream performance by training new low-capacity MLP and RF probes  $\tilde{f}$  to predict  $y$  from  $c$ , i.e.  $\hat{y}^i = \tilde{f}_i(c)$  (see Fig. 4c). For MLP probes, low capacity means linear. For RF probes, low capacity means the maximum tree depth is 10. Next, we average the downstream performances across all 14 tasks before computing the correlation coefficient between this average and each of the D, C, I, and E scores.

**Analysis.** Figs. 4a and 4b show that E is strongly correlated with downstream performance when using both MLP ( $\rho = 0.96$ ,  $p = 8e-18$ ) and RF probes ( $\rho = 0.88$ ,  $p = 2e-10$ ). In contrast, mixing-based disentanglement scores (D, C) exhibit much weaker correlations with MLP probes, corroborating the results of Truble et al. (2022, Fig. 8) who also found a weak correlation between D and downstream performance on reinforcement learning tasks with MLPs. See App. B.1 for further details and results.

Published as a conference paper at ICLR 2023

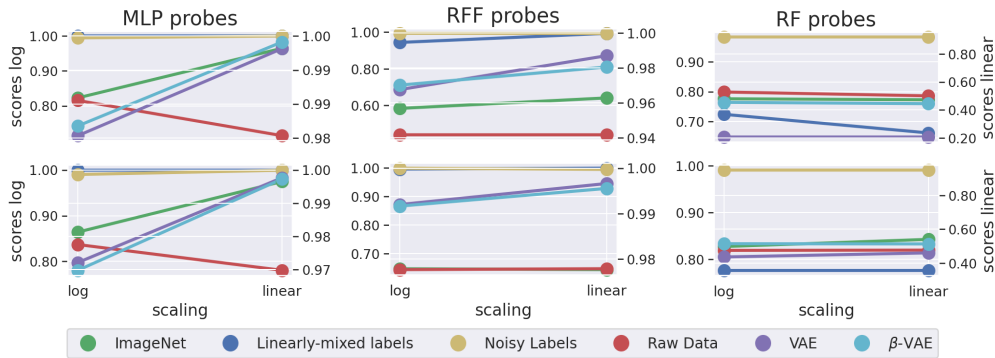


Figure 5: Explicitness scores on MPI3D-Real (top) and Cars3D (bottom) for different representations (see legend). Each pair of points represents the score with logarithmic (left) and linear (right) capacity-scaling.

## 7 DISCUSSION

**Why connect disentanglement and identifiability?** Connecting prediction-based evaluation in the disentanglement literature to the more theoretical notion of identifiability has several benefits. Firstly, it provides a concrete link between two often-separate communities. Secondly, it endows the often empirically-driven or practice-focused disentanglement metrics with a solid and well-studied theoretical foundation. Thirdly, compared to the commonly-used MCC or Amari distance, it provides the ICA or identifiability community with more fine-grained empirical measures, as discussed in § 4.1.

**Measuring probe capacity.** Our measure of explicitness  $E$  depends strongly on the choice of capacity measure for a probe or function class. For some probes like RFs or RFFs, there exist natural measures of capacity. However, for other probes like MLPs, coming up with a good capacity measure is itself an important and active area of research (Jiang et al., 2020; Dziugaite et al., 2020). Another difficulty arises from choosing a capacity *scale*, with different scales (e.g., log, linear, etc.) leading to loss-capacity curves with different shapes, areas and thus explicitness scores. To investigate the extent of this issue, i.e., the sensitivity of our explicitness measure to the choice of capacity scale, Fig. 5 compares the explicitness scores when using logarithmic and linear scaling. Here we see that the *ranking* essentially remains the same except for the raw-data representation with MLP probes.

**Measuring feature importance.** Similarly, the choice of feature-importance measure has a strong influence on the  $D$  and  $C$  scores, with some probes having natural or in-built measures (e.g., random forests) and others not (e.g., MLPs). For the latter, we proposed the use of probe-agnostic feature-importance measures like SAGE, and specified the conditions (Corollary 3.5) that importance measures must satisfy if the resulting  $D$  and  $C$  scores are to be connected to identifiability. As with probe capacity, coming up with good measures of feature importance is its own orthogonal field of study (e.g., model explainability), with future advances likely to improve the DCI-ES framework.

**What about explicitness for other tasks  $y$ ?** While we focused on the explicitness or ease-of-use of a representation for predicting the data-generative factors  $z$ , one may also be interested in its ease-of-use for other tasks/labels  $y$ . While it is often implicitly assumed that the ease-of-use for predicting  $z$  correlates with the ease-of-use for common tasks of interest (e.g., object classification, segmentation, etc.), future work could directly evaluate the explicitness of a representation for particular tasks  $y$ . For example, one could consider the entire loss-capacity curve when benchmarking self-supervised representations on ImageNet, rather than just linear-probe performance (a single slice). Future work could also explore the trade-off between *explicit but task-specific* and *implicit but task-agnostic* representations.

## 8 CONCLUSION

We have presented DCI-ES—an extended disentanglement framework with two new complementary measures of representation quality—and proven its connections to identifiability. In particular, we have advocated for additionally measuring the explicitness ( $E$ ) of a representation by the functional capacity required to use it, and proposed to quantify this explicitness using a representation’s loss-capacity curve. Together with the size ( $S$ ) of a representation, we believe that our extended DCI-ES framework allows for a more fine-grained and nuanced benchmarking of representation quality.

Published as a conference paper at ICLR 2023

---

#### ACKNOWLEDGMENTS

The authors would like to thank Chris Williams, Francesco Locatello, Nasim Rahaman, Sidak Singh and Yash Sharma for helpful discussions and comments. This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A, 01IS18039B; and by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

#### REFERENCES

- Shun-ichi Amari, Andrzej Cichocki, Howard Hua Yang, et al. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems*, pp. 757–763, 1996. [Cited on page 5.]
- Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. [Cited on page 6.]
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. [Cited on page 1.]
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. [Cited on page 2.]
- Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2615–2625, 2018. [Cited on page 1.]
- Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2180–2188, 2016. [Cited on page 1.]
- Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. [Cited on page 3.]
- Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17212–17223, 2020. [Cited on pages 5 and 6.]
- Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy. In search of robust measures of generalization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11723–11733, 2020. [Cited on page 9.]
- Cian Eastwood and Christopher K I Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. [Cited on pages 1, 2, 3, and 6.]
- Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. [Cited on page 5.]
- Anna Golubeva, Guy Gur-Ari, and Behnam Neyshabur. Are wider nets better given the same number of parameters? In *International Conference on Learning Representations*, 2021. [Cited on page 5.]
- Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems*, 32, 2019. [Cited on pages 2 and 6.]
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer, 2009. [Cited on page 6.]

Published as a conference paper at ICLR 2023

---

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pp. 770–778, 2016. [Cited on page 7.]
- Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner.  $\beta$ -VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. [Cited on pages 1 and 7.]
- Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999. [Cited on page 3.]
- Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868. PMLR, 2019. [Cited on page 3.]
- Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. [Cited on page 9.]
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. [Cited on page 6.]
- 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*, pp. 2207–2217. PMLR, 2020. [Cited on page 5.]
- Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658, 2018. [Cited on page 1.]
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. [Cited on page 7.]
- Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014. [Cited on page 7.]
- Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2539–2547, 2015. [Cited on page 1.]
- Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017. [Cited on page 1.]
- 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. *Journal of Machine Learning Research*, 21(209):1–62, 2020. [Cited on page 1.]
- Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4768–4777, 2017. [Cited on page 6.]
- Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 1(12), 2021. [Cited on page 6.]
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2007. [Cited on page 7.]
- Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1252–1260, 2015. [Cited on pages 2 and 6.]
- Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, volume 31, pp. 185–194, 2018. [Cited on pages 1, 2, and 6.]

Published as a conference paper at ICLR 2023

---

- Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953. [Cited on page 6.]
- Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*, 2020. [Cited on page 1.]
- 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*, pp. 6056–6065, 2019. [Cited on page 1.]
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. [Cited on page 2.]
- Frederik Träuble, Andrea Dittadi, Manuel Wuthrich, 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*, 2022. [Cited on page 8.]
- Tom Viering and Marco Loog. The shape of learning curves: a review. *arXiv preprint arXiv:2103.10948*, 2021. [Cited on page 6.]
- William F Whitney, Min Jae Song, David Brandfonbrener, Jaan Altosaar, and Kyunghyun Cho. Evaluating representations by the complexity of learning low-loss predictors. *arXiv preprint arXiv:2009.07368*, 2020. [Cited on page 6.]
- Howard Hua Yang and Shun-ichi Amari. Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information. *Neural Computation*, 9(7):1457–1482, 1997. [Cited on page 5.]

## 6.3 Comments on the paper

**Measuring probe capacity.** Our measure of explicitness  $E$  depends strongly on the choice of capacity measure for a probe or function class. For some probes like RFs or RFFs, there exist natural measures of capacity. However, for other probes like MLPs, coming up with a good capacity measure is itself an important and active area of research (Dziugaite et al., 2020; Jiang et al., 2020). While we used the number of parameters as a simple measure of the available or upper-bound probe capacity, future work may explore the use of more sophisticated measures of probe capacity, such as those based on the *used* or *effective* capacity of a trained MLP (Hanin and Rolnick, 2019; Maddox et al., 2020).

**Explicitness for other tasks  $\mathbf{y}$ .** We focused on the explicitness or ease-of-use of a representation for predicting the data-generative factors  $\mathbf{z}$  (in line with the disentanglement and identifiability literature). However, one may also be interested in its ease-of-use for other tasks or labels  $\mathbf{y}$ . While it is often implicitly assumed that the ease-of-use for predicting  $\mathbf{z}$  correlates with the ease-of-use for common tasks of interest, such as object classification or segmentation, future work could directly evaluate the explicitness of a representation for particular tasks  $\mathbf{y}$ . For example, one could consider the entire loss-capacity curve when benchmarking self-supervised representations on ImageNet, rather than just linear-probe performance (corresponding to a single slice of the loss-capacity curve). It would also be interesting to explore the inherent trade-off between *explicit but task-specific* and *implicit but task-agnostic* representations.

**An evaluation framework for causal representation learning.** Suter et al. (2019) proposed a disentanglement measure based on representation’s robustness to interventions on the data-generative factors  $z_j$  (see their Defns. 2&3), which can also be used to construct a matrix of feature importances  $R$  (see their Figs. 8–12). Similar ideas may help extend the DCI-ES framework to the evaluation of *causal representations* (Schölkopf et al., 2021). This would provide more fine-grained empirical measures than the commonly-used alternatives, such as the mean correlation coefficient (MCC) and the Amari distance (Amari et al., 1996; Yang and Amari, 1997).



# 7

## Conclusions

In this thesis, we explored a number of ways in which machine learning systems can be prepared for distribution shift. In particular, we explored four of my works which sought to prepare for an inevitable distribution shift.

First, in Chapter 3, we explored source-free domain adaptation. Here we showed how to prepare for and resolve measurement shift—one particular type of distribution shift. By storing and re-aligning lightweight statistics of the feature distribution, we saw improved accuracy, calibration, and data efficiency compared to the dominant prior approach of entropy minimisation.

Next, in Chapter 4, we explored a probabilistic framework for domain generalisation, showing how to build machine learning systems that are robust to distribution shift with a desired probability—so long as the collected training-domain data is representative of the shifts we are likely to see at test time. In particular, by minimising a particular quantile of a model’s performance distribution over training domains, we can learn models that perform well on unseen test domains with the corresponding probability. We also highlighted the importance of comparing domain-generalisation algorithms based on their tail or quantile performance since improved robustness is often invisible through the lens of average performance.

After that, in Chapter 5, we explored how invariant predictions could be used to harness spurious features in the test domain *without labels*, reliably boosting the performance of domain generalisation algorithms. In particular, we showed that invariant predictions provide sufficient guidance for doing so, provided that the invariant/stable and spurious/unstable features are conditionally independent given the label. Based on this theoretical insight, we then proposed the Stable Feature Boosting (SFB) algorithm and demonstrated its effectiveness on real and synthetic datasets.



Finally, in Chapter 6, we explored disentangled representations and showed how to prepare for unknown downstream tasks (i.e., task shifts) by learning representations that are *easy to use in terms of functional capacity*. In particular, we measured this ease-of-use or explicitness using a representation’s loss-capacity curve, and showed that explicitness measure better correlates with downstream performance than existing disentanglement measures.

## 7.1 Future directions

**Source-Free Domain Adaptation (Chapter 3).** In § 3.3 we discussed how, in retrospect, one could store any amount of information about the source dataset and use this to adapt—from lightweight histogram bin-counts to the entire source dataset. It would be interesting to explore storage-performance trade-offs by comparing methods in terms of their storage requirements and resulting performance.

In addition, with the SFDA setting often motivated by privacy constraints, it would be interesting to explore just how private different amounts and types of storage are, starting with histogram bin counts.

Finally, we discussed the idea of automatic, shift-dependent fine-tuning strategies in which different layers are adapted depending on the type of shift encountered. Here, when determining which layers should be adapted and *how* to adapt them, it would be interesting to explore both supervised and unsupervised approaches.

**Domain Generalisation: A Probabilistic Framework (Chapter 4).** In § 4.3 we discussed the need for DG benchmarks with multiple test domains, allowing the community to compare methods based on the distribution of their performance—not just the average.

In addition, we discussed how one may seek probabilistic robustness both across and within domains, e.g., across hospitals and across patients within those hospitals, and how this could be achieved with nested quantile-minimisation problems.

**Domain Generalisation: Harnessing Spurious Features (Chapter 5).** In § 5.3 we discussed how the assumption of complementarity is perhaps the biggest limitation of our work, and how future work could investigate ways to weaken it.

In addition, we discussed the difficulty of optimising SFB over deep neural networks and how future work may look to address this by: (1) using ERM-pretrained networks;

(2) freezing all but the final layer after ERM-pretraining; and (3) using completely separate networks for the stable and unstable predictors.

**Disentangled Representations (Chapter 6).** In § 6.3 we discussed how one could explore a representation’s explicitness with respect to any task or set of labels  $\mathbf{y}$ , not just the data-generative factors  $\mathbf{z}$ , and how this could lead to an interesting investigation of explicit-but-task-specific vs. implicit-but-task-agnostic representations.

In addition, we discussed how future work may explore the use of more sophisticated measures of probe capacity for neural networks, such as the effective or used capacity, in order to improve the resulting explicitness score.

Finally, we discussed how one could use interventional robustness—rather than predictive accuracy—to quantify the relationship between the representation and data-generative factors, ultimately extending the DCI-ES framework to the evaluation of causal representations.



# Bibliography

- Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367.
- Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. (2021). Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34.
- Amari, S.-i., Cichocki, A., Yang, H. H., et al. (1996). A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems*, pages 757–763.
- Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv:1907.02893*.
- Balaji, Y., Sankaranarayanan, S., and Chellappa, R. (2018). MetaReg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31.
- Balcan, M.-F., Blum, A., and Yang, K. (2004). Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems 17*.
- Bareinboim, E. and Pearl, J. (2014). Transportability from multiple environments with limited experiments: Completeness results. In *Advances in Neural Information Processing Systems 27*, pages 280–288.
- Beede, E., Baylor, E., Hersch, F., Iurchenko, A., Wilcox, L., Ruamviboonsuk, P., and Vardoulakis, L. M. (2020). A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–12. Association for Computing Machinery.
- Beery, S., Agarwal, A., Cole, E., and Birodkar, V. (2021). The iWildCam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*.
- Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, 79(1):151–175.

- Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 137–144.
- Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). Robust optimization. In *Robust Optimization*. Princeton University Press.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.
- Bengio, Y., Deleu, T., Rahaman, N., Ke, N. R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. (2020). A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*.
- Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G., and Scott, C. (2021). Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1):46–100.
- Blanchard, G., Lee, G., and Scott, C. (2011). Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems*, volume 24.
- Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100.
- Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification. In *World Wide Web Conference*, pages 491–500.
- Bouchacourt, D., Tomioka, R., and Nowozin, S. (2018). Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.
- Chen, R. T., Li, X., Grosse, R., and Duvenaud, D. (2018). Isolating sources of disentanglement in vaes. In *Advances in Neural Information Processing Systems*, volume 31, pages 2615–2625.
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 29, pages 2180–2188.
- Dai, D. and Van Gool, L. (2018). Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *International Conference on Intelligent Transportation Systems*, pages 3819–3824.

- Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624.
- Deshmukh, A. A., Lei, Y., Sharma, S., Dogan, U., Cutler, J. W., and Scott, C. (2019). A generalization error bound for multi-class domain generalization. *arXiv preprint arXiv:1905.10392*.
- Desjardins, G., Courville, A., and Bengio, Y. (2012). Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*.
- Dou, Q., Coelho de Castro, D., Kamnitsas, K., and Glocker, B. (2019). Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32.
- Dubey, A., Ramanathan, V., Pentland, A., and Mahajan, D. (2021). Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349.
- Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer.
- Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.
- Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., and Roy, D. M. (2020). In search of robust measures of generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 11723–11733.
- Eastwood, C., Mason, I., and Williams, C. K. I. (2021). Unit-level surprise in neural networks. In *NeurIPS 2021 Workshop "I (Still) Can't Believe It's Not Better!"*, volume 163 of *Proceedings of Machine Learning Research*, pages 33–40.
- Eastwood, C., Mason, I., Williams, C. K. I., and Schölkopf, B. (2022a). Source-free adaptation to measurement shift via bottom-up feature restoration. In *The Tenth International Conference on Learning Representations*.
- Eastwood, C., Nanbo, L., and Williams, C. K. I. (2022b). Align-Deform-Subtract: an interventional framework for explaining object differences. In *ICLR 2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*.
- Eastwood, C., Nicolicioiu, A. L., Kügelgen, J. V., Kekić, A., Träuble, F., Dittadi, A., and Schölkopf, B. (2023a). DCI-ES: An extended disentanglement framework with connections to identifiability. In *The Eleventh International Conference on Learning Representations*.

- Eastwood, C., Robey, A., Singh, S., Kügelgen, J. V., Hassani, H., Pappas, G. J., and Schölkopf, B. (2022c). Probable domain generalization via quantile risk minimization. In *Advances in Neural Information Processing Systems*, volume 35, pages 17340–17358.
- Eastwood, C., Singh, S., Nicolicioiu, L. A., Von Kügelgen, J., and Schölkopf, B. (2023b). Spuriousity didn't kill the classifier: Using invariant predictions to harness spurious features. In *Advances in Neural Information Processing Systems*.
- Eastwood, C., von Kügelgen, J., Ericsson, L., Bouchacourt, D., Vincent, P., Schölkopf, B., and Ibrahim, M. (2023c). Self-supervised disentanglement by leveraging structure in data augmentations. *Preprint arXiv:2311.08815*.
- Eastwood, C. and Williams, C. K. I. (2018). A framework for the quantitative evaluation of disentangled representations. In *The Sixth International Conference on Learning Representations*.
- Gamella, J. L. and Heinze-Deml, C. (2020). Active invariant causal prediction: Experiment selection through stability. *Advances in Neural Information Processing Systems*, 33:15464–15475.
- Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030.
- Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587.
- Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. (2016). Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, pages 2839–2848. PMLR.
- Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. In *International Conference on Learning Representations*.
- Hanin, B. and Rolnick, D. (2019). Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, volume 32, pages 361–370.
- Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S. A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S. J., Loveland, T. R., et al. (2013). High-resolution global maps of 21st-century forest cover change. *Science*, 342(6160):850–853.

- Heinze-Deml, C., Peters, J., and Meinshausen, N. (2018). Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2).
- Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017).  $\beta$ -VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.
- Hosoya, H. (2019). Group-based learning of disentangled representations with generalizability for novel contents. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2506–2513.
- Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169.
- Huang, B., Zhang, K., Zhang, J., Sanchez-Romero, R., Glymour, C., and Schölkopf, B. (2017). Behind distribution shift: Mining driving forces of changes and causal arrows. In *IEEE 17th International Conference on Data Mining (ICDM 2017)*, pages 913–918.
- Huang, Z., Wang, H., Xing, E. P., and Huang, D. (2020). Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pages 124–140. Springer.
- Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.
- Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. (2020). Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*.
- Jovicich, J., Czanner, S., Han, X., Salat, D., van der Kouwe, A., Quinn, B., Pacheco, J., Albert, M., Killiany, R., Blacker, D., et al. (2009). MRI-derived measurements of human subcortical, ventricular and intracranial brain volumes: reliability effects of scan sessions, acquisition sequences, data analyses, scanner upgrade, scanner vendors and field strengths. *Neuroimage*, 46(1):177–192.
- Kim, D., Yoo, Y., Park, S., Kim, J., and Lee, J. (2021). Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628.
- Kim, H. and Mnih, A. (2018). Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658.
- Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.



- Kirichenko, P., Izmailov, P., and Wilson, A. G. (2022). Last layer re-training is sufficient for robustness to spurious correlations. In *Advances in Neural Information Processing Systems*.
- Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B. A., Haque, I. S., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. (2021). WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*.
- Krause, A., Perona, P., and Gomes, R. (2010). Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems*, pages 775–783.
- Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (REx). In *International Conference on Machine Learning*, volume 139, pages 5815–5826.
- Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. (2015). Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, volume 28, pages 2539–2547.
- Kundu, J. N., Venkat, N., Babu, R. V., et al. (2020). Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553.
- Kurmi, V. K., Subramanian, V. K., and Namboodiri, V. P. (2021). Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.
- Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3.
- Li, D. and Hospedales, T. (2020). Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer.
- Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. (2018a). Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Li, H., Wang, Y., Wan, R., Wang, S., Li, T.-Q., and Kot, A. (2020a). Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in Neural Information Processing Systems*, 33:3118–3129.

- Li, N., Eastwood, C., and Fisher, R. (2020b). Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems*, volume 33, pages 5656–5666.
- Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. (2020c). Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650.
- Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. (2018b). Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639.
- Li, Y., Wang, N., Shi, J., Liu, J., and Hou, X. (2017). Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations Workshop*.
- Li, Y., Yang, Y., Zhou, W., and Hospedales, T. (2019). Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR.
- Liang, J., Hu, D., and Feng, J. (2020). Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039.
- Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124.
- Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2020a). A sober look at the unsupervised learning of disentangled representations and their evaluation. *Journal of Machine Learning Research*, 21(209):1–62.
- Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020b). Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359.
- Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105.
- Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2018). Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*.
- Maddox, W. J., Benton, G., and Wilson, A. G. (2020). Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*.
- Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., and Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *Machine Learning for Autonomous Driving Workshop, NeurIPS 2019*.

- Morerio, P., Volpi, R., Ragonesi, R., and Murino, V. (2020). Generative pseudo-label refinement for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3130–3139.
- Motiian, S., Jones, Q., Iranmanesh, S., and Doretto, G. (2017). Few-shot adversarial domain adaptation. *Advances in Neural Information Processing Systems*, 30.
- Muandet, K., Balduzzi, D., and Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18.
- Nagarajan, V., Andreassen, A., and Neyshabur, B. (2021). Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*.
- Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012.
- Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference: Foundations and learning algorithms*. The MIT Press.
- Pfister, N., Bühlmann, P., and Peters, J. (2019). Invariant causal prediction for sequential data. *Journal of the American Statistical Association*, 114(527):1264–1276.
- Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2008). *Dataset shift in machine learning*. MIT Press.
- Reed, S. E., Zhang, Y., Zhang, Y., and Lee, H. (2015). Deep visual analogy-making. *Advances in Neural Information Processing Systems*, 28.
- Ridgeway, K. and Mozer, M. C. (2018). Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, volume 31, pages 185–194.
- Robey, A., Pappas, G. J., and Hassani, H. (2021). Model-based domain generalization. In *Advances in Neural Information Processing Systems*.
- Rocco, I., Arandjelović, R., and Sivic, J. (2018). Convolutional neural network architecture for geometric matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2553–2567.
- Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. (2018). Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342.
- Rosenfeld, E., Ravikumar, P., and Risteski, A. (2022). Domain-adjusted regression or: ERM may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*.

- Royer, A. and Lampert, C. (2020). A flexible selection scheme for minimum-effort transfer learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2191–2200.
- Sagawa\*, S., Koh\*, P. W., Hashimoto, T. B., and Liang, P. (2020). Distributionally robust neural networks. In *International Conference on Learning Representations*.
- Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München.
- Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879.
- Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634.
- Shu, R., Bui, H., Narui, H., and Ermon, S. (2018). A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*.
- Shu, R., Chen, Y., Kumar, A., Ermon, S., and Poole, B. (2020). Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*.
- Shu, Y., Cao, Z., Wang, C., Wang, J., and Long, M. (2021). Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9624–9633.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.
- Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. (2023). Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.
- Stan, S. and Rostami, M. (2021). Unsupervised model adaptation for continual semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2593–2601.
- Storkey, A. J. (2009). When training and test sets are different: characterising learning transfer. In *Dataset Shift in Machine Learning*, pages 3–28. MIT Press.
- Suter, R., Miladinovic, D., Schölkopf, B., and Bauer, S. (2019). Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065.
- Thrun, S. and Pratt, L. (1998). Learning to learn: Introduction and overview. In *Learning to Learn*, pages 3–17. Springer.

- Tseng, H.-Y., Lee, H.-Y., Huang, J.-B., and Yang, M.-H. (2020). Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*.
- Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2019). Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.
- Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New York, NY.
- von Kügelgen, J., Mey, A., and Loog, M. (2019). Semi-generative modelling: Covariate-shift adaptation with cause and effect features. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1361–1369. PMLR.
- Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021). TENT: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.
- Wang, W. and Zhou, Z.-H. (2010). A new analysis of co-training. In *International Conference on Machine Learning*, volume 2, page 3.
- Yang, H. H. and Amari, S.-i. (1997). Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information. *Neural Computation*, 9(7):1457–1482.
- Yeh, H.-W., Yang, B., Yuen, P. C., and Harada, T. (2021). SoFA: Source-data-free feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 474–483.
- Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, page 3320–3328.
- Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Medicine*, 15(11).
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833.
- Zhang, J., Lopez-Paz, D., and Bottou, L. (2022). Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, pages 26397–26411. PMLR.
- Zhang, K., Gong, M., and Schölkopf, B. (2015). Multi-source domain adaptation: A causal view. In *Twenty-ninth AAAI Conference on Artificial Intelligence*.

- Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2013). Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR.
- Zhang, M., Marklund, H., Dhawan, N., Gupta, A., Levine, S., and Finn, C. (2021). Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678.
- Zhao, S., Gong, M., Liu, T., Fu, H., and Tao, D. (2020). Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107.



# Appendix A

## Paper Appendices

This appendix contains supplementary material for each of the papers presented in this thesis, namely:

[A.1 Source-Free Domain Adaptation \(§ 3.2\)](#)

[A.2 Domain Generalisation: A Probabilistic Framework \(§ 4.2\)](#)

[A.3 Domain Generalisation: Harnessing Spurious Features \(§ 5.2\)](#)

[A.4 Disentangled Representations \(§ 6.2\)](#)

### **A.1 Source-Free Domain Adaptation (§ 3.2)**



Published as a conference paper at ICLR 2022

---

# Appendix

## Table of Contents

---

<b>A</b>	<b>Soft binning</b>	<b>16</b>
<b>B</b>	<b>FR algorithm</b>	<b>17</b>
<b>C</b>	<b>When might FR work?</b>	<b>17</b>
<b>D</b>	<b>Common UDA benchmarks are not measurement shifts</b>	<b>18</b>
<b>E</b>	<b>Further related work</b>	<b>19</b>
<b>F</b>	<b>Datasets</b>	<b>19</b>
<b>G</b>	<b>Further implementation details</b>	<b>22</b>
<b>H</b>	<b>Reliability diagrams and confidence histograms</b>	<b>23</b>
<b>I</b>	<b>Activation distributions</b>	<b>25</b>
<b>J</b>	<b>Further analysis</b>	<b>27</b>
	J.1 Efficacy of bottom-up training . . . . .	27
	J.2 Loss ablation study . . . . .	27
	J.3 Who is affected . . . . .	28
	J.4 Who moves . . . . .	28
<b>K</b>	<b>Full Results</b>	<b>29</b>
	K.1 Digit and character summary results . . . . .	29
	K.2 Online results . . . . .	30
	K.3 CAMELYON17 results . . . . .	30
	K.4 MNIST-C full results . . . . .	31
	K.5 EMNIST-DA full results . . . . .	32
	K.6 CIFAR-10-C full results . . . . .	33
	K.7 CIFAR-100-C full results . . . . .	34
	K.8 CIFAR-10-C full online results . . . . .	35
	K.9 CIFAR-100-C full online results . . . . .	36
<b>L</b>	<b>Notations</b>	<b>37</b>

---

Published as a conference paper at ICLR 2022

## A SOFT BINNING

**Function.** Let  $z \sim p_z$  be a continuous 1D variable for which we have  $n$  samples  $\{z^{(i)}\}_{i=1}^n$ . The goal is approximately parameterize  $p_z$  using  $B$  normalized bin counts  $\pi_z = [\pi_{z,1}, \dots, \pi_{z,B}]$ , where  $\pi_{z,b}$  represents the probability that  $z$  falls into bin  $b$  and  $\sum_{b=1}^B \pi_{z,b} = 1$ . We achieve this using the soft binning function of Yang et al. (2018, Section 3.1). The first step is to find the range of  $z$ , i.e. the minimum and maximum denoted  $z^{min} = \min_i z^{(i)}$  and  $z^{max} = \max_i z^{(i)}$  respectively. This will allow us to normalize the range of our samples  $z^{(i)}$  to be  $[0, 1]$  and thus ensure that binning “softness”, i.e. the degree to which mass is distributed into nearby bins, is comparable across variables with different ranges. The second step is to define  $B - 1$  uniformly-spaced and monotonically-increasing cut points (i.e. bin edges) over this normalized range  $[0, 1]$ , denoted  $\mathbf{c} = [c_1, c_2, \dots, c_{B-1}] = \frac{1}{B-2}[0, 1, 2, \dots, B-3, B-2]$ . The third step is to compute the  $B$ -dimensional vector of soft counts for a sample  $z^{(i)}$ , denoted  $\mathbf{u}(z^{(i)})$ , using soft binning vector-valued function  $\mathbf{u}$ ,

$$\mathbf{u}(z^{(i)}; z^{min}, z^{max}) = \sigma\left(\frac{\mathbf{w} \left( \frac{z^{(i)} - z^{min}}{z^{max} - z^{min}} \right) + \mathbf{w}_0}{\tau}\right), \quad (4)$$

where  $\mathbf{w} = [1, 2, \dots, B]$ ,  $\mathbf{w}_0 = [0, -c_1, -c_1 - c_2, \dots, -\sum_{j=1}^{B-1} c_j]$ ,  $\tau > 0$  is a temperature factor,  $\sigma$  is the softmax function,  $\mathbf{u}(z^{(i)})_b$  is the mass assigned to bin  $b$ , and  $\sum_{b=1}^B \mathbf{u}(z^{(i)})_b = 1$ . Note that: (i) both  $\mathbf{w}$  and  $\mathbf{w}_0$  are constant vectors for a pre-specified number of bins  $B$ ; (ii) as  $\tau \rightarrow 0$ ,  $\mathbf{u}(z^{(i)})$  tends to a one-hot vector; and (iii) the  $B - 1$  cut points  $\mathbf{c}$  result in  $B$  bins, where values  $z^{(i)} < 0$  or  $z^{(i)} > 1$  are handled sensibly by the soft binning function in order to catch new samples that lie outside the range of our original  $n$  samples (as  $\tau \rightarrow 0$ , they will appear in the leftmost or rightmost bin respectively). Finally, we get the total counts per bin by summing over the per-sample soft counts  $\mathbf{u}(z^{(i)})$ , before normalizing by the total number of samples  $n$  to get the normalized bin counts  $\pi_z$ , i.e.,  $\pi_z = \sum_{i=1}^n \frac{\mathbf{u}(z^{(i)}; z^{min}, z^{max})}{n}$ .

**Memory cost.** When using 32-bit floating point numbers for each (soft) bin count, the memory cost of soft binning is  $32 \times B \times D$  bits—depending only on the number bins  $B$  and the number of features  $D$ , and *not* on the dataset size. For concreteness, Table 5 compares the cost of storing bin counts to that of: (i) storing the whole source dataset; and (ii) storing the (weights of the) source model. As in our experiments, we assume 8 bins per feature and the following network architectures: a variation of LeNet (LeCun et al., 1998) for MNIST; ResNet-18 (He et al., 2016) for CIFAR-100; and ResNet-101 (He et al., 2016) for both VisDA-C (Peng et al., 2018) and ImageNet (Russakovsky et al., 2015).

Table 5: Storage size for different datasets and their corresponding source models.

Storage size (MB)	MNIST	CFR-100	VisDA-C	ImageNet
Source dataset	33	150	7885	138000
Source model	0.9	49	173	173
Source bin-counts	0.004	0.02	0.5	0.5

Published as a conference paper at ICLR 2022

## B FR ALGORITHM

Algorithm 1 gives the algorithm for FR at *development time*, where a source model is trained before saving approximations of the feature and logit distributions under the source data. Algorithm 2 gives the algorithm for FR at *deployment time*, where the feature-extractor is adapted such that the approximate feature and logit distributions under the target data realign with those saved on the source.

Algorithm 1: FR at <i>development time</i> .	Algorithm 2: FR at <i>deployment time</i> .
<b>Input:</b> Source model $f_s$ , labelled source data $D_s = (X_s, Y_s)$ , number of bins $B$ , number of training iterations $I$ .	<b>Input:</b> Source model $f_s$ , unlabelled target data $X_t$ , source data statistics $\mathcal{S}_s$ , number of adaptation iterations $I$ .
<pre> /* Train src model <math>f_s = h \circ g_s</math> */ for <math>i</math> in range(<math>I</math>) do   <math>L_i \leftarrow \mathcal{L}_{src}(f_s, D_s)</math>;   <math>f_s \leftarrow \text{SGD}(f_s, L_i)</math>;  /* Calc. feat.&amp;logit ranges */ <math>\mathbf{z}^{min}, \mathbf{z}^{max} \leftarrow \text{CALC\_RANGE}(f_s, X_s)</math>; <math>\mathbf{a}^{min}, \mathbf{a}^{max} \leftarrow \text{CALC\_RANGE}(f_s, X_s)</math>;  /* Calc. feat.&amp;logit bin cnts */ <math>\pi_{\mathbf{z}}^s \leftarrow \text{CALC\_BC}(f_s, X_s; \mathbf{z}^{min}, \mathbf{z}^{max}, B)</math>; <math>\pi_{\mathbf{a}}^s \leftarrow \text{CALC\_BC}(f_s, X_s; \mathbf{a}^{min}, \mathbf{a}^{max}, B)</math>;  /* Gather source stats <math>\mathcal{S}_s</math> */ <math>\mathcal{S}_s \leftarrow \{\pi_{\mathbf{z}}^s, \pi_{\mathbf{a}}^s, \mathbf{z}^{min}, \mathbf{z}^{max}, \mathbf{a}^{min}, \mathbf{a}^{max}\}</math>;  <b>Output:</b> <math>f_s, \mathcal{S}_s</math> </pre>	<pre> /* Init trgt model <math>f_t = h \circ g_t</math> */ <math>f_t \leftarrow f_s</math>;  /* Adapt trgt feat.-extractr <math>g_t</math> */ for <math>i</math> in range(<math>I</math>) do   <math>\pi_{\mathbf{z}}^t \leftarrow \text{CALC\_BC}(f_t, X_t; \mathbf{z}^{min}, \mathbf{z}^{max}, B)</math>;   <math>\pi_{\mathbf{a}}^t \leftarrow \text{CALC\_BC}(f_t, X_t; \mathbf{a}^{min}, \mathbf{a}^{max}, B)</math>;    <math>L_i \leftarrow \mathcal{L}_{tgt}(\pi_{\mathbf{z}}^s, \pi_{\mathbf{z}}^t, \pi_{\mathbf{a}}^s, \pi_{\mathbf{a}}^t)</math>;   <math>g_t \leftarrow \text{SGD}(g_t, L_i)</math>;  <b>Output:</b> <math>g_t</math> </pre>

## C WHEN MIGHT FR WORK?

**Toy example where FR will work.** Let  $L$  take two values  $\{-1, 1\}$ , and let

$$Y = L \tag{5}$$

$$X = U[L - 0.5, L + 0.5] + E, \tag{6}$$

where  $U$  denotes a uniform distribution and  $E$  a domain-specific offset (this setup is depicted in Figure 1a). Then the optimal classifier  $f : X \rightarrow Y$  can be written as  $f(X) = \text{sign}(X - E)$ . Imagine the source domain has  $E = 0$ , and the target domain has  $E = 2$ . Then all points will be initially classified as positive in the target domain, but FR will restore optimal performance by essentially “re-normalizing”  $X$  to achieve an intermediate feature representation  $Z$  with the same distribution as before (in the source domain).

**Toy example where FR will not work.** Let  $L$  be a rotationally-symmetric multivariate distribution (e.g. a standard multivariate Gaussian), and let  $X$  be a rotated version of  $L$  where the rotation depends on  $E$ . Now let  $Y = L_1$ , the first component of  $L$ . Then any projection of  $X$  will have the correct marginal distribution, hence FR will not work here as matching the marginal distributions of the intermediate feature representation  $Z$  will not be enough to yield the desired invariant representation.

**How to know if FR is suitable.** We believe it reasonable to assume that one has knowledge of the type of shifts that are likely to occur upon deployment. For example, if deploying a medical imaging system to a new hospital, one may know that the imaging and staining techniques may differ but the catchment populations are similar in e.g. cancer rate. In such cases, we can deduce that measurement shift is likely and thus FR is suitable.

Published as a conference paper at ICLR 2022

## D COMMON UDA BENCHMARKS ARE NOT MEASUREMENT SHIFTS

**Overview.** The standard approach for common UDA benchmarks like VisDA-C (Peng et al., 2018) is to first pretrain on ImageNet to gain more “general” visual features and then carefully fine-tune these features on (i) the source domain, and then (ii) the target domain, effectively making the adaptation task ImageNet  $\rightarrow$  synthetic  $\rightarrow$  real. Here, we use VisDA-C to: (i) investigate the reliance of existing methods on ImageNet pretraining; (ii) evaluate our FR and BUFR methods on domain shifts that *require* learning new features (i.e. *non* measurement shifts); and (iii) investigate the effect of label shift on our methods (which violates the assumption of measurement shift and indeed even domain shift).

**Reducing label shift.** For (iii), we first note that VisDA-C contains significant label shift. For example, 8% of examples are labelled ‘car’ in the source domain, while 19% of examples are labelled ‘car’ in the target domain. To correct for this while retaining as many examples as possible, we randomly drop examples from some classes and oversample examples from others so that all classes have 11000 examples in the source domain and 3500 examples in the target domain—this is labelled as “No label shift” in Table 6.

**Results.** In Table 6 we see that: (i) without ImageNet pre-training, all (tested) methods fail—despite similar accuracy being achieved in the source domain with or without ImageNet pre-training (compare  $\times\times$  vs.  $\checkmark\times$ ); (ii) with the standard VisDA-C setup (i.e.  $\checkmark\times$ ), AdaBN  $<$  FR  $\ll$  SHOT, as SHOT learns *new* discriminative features in the target domain; and (iii) correcting for label shift boosts the performance of FR and closes the gap with SHOT (compare  $\checkmark\times$  vs.  $\checkmark\checkmark$ ), but some gap remains as *VisDA-C is not a measurement shift but rather a more general domain shift*. Finally, we note that ImageNet pretraining makes the features in early layers quite robust, reducing the advantage of bottom-up training.

**Implementation details.** These results were achieved using a standard VisDA-C implementation/setup: we train a ResNet-101 (He et al., 2016) (optionally pre-trained on ImageNet) for 15 epochs using SGD, a learning rate of 0.001, and a batch size of 64. We additionally adopt the learning rate scheduling of (Ganin & Lempitsky, 2015; Long et al., 2018; Liang et al., 2020) in the source domain, and reduce the learning rate to 0.0001 in the target domain.

Table 6: VisDA-C results (ResNet-101). *No label shift*: examples were dropped or oversampled to correct for label shift.

Model	ImageNet pretrain	No label shift	Avg. Acc.
No corruption	$\times$	$\times$	99.8
Source-only	$\times$	$\times$	10.4
AdaBN (Li et al., 2017)	$\times$	$\times$	<b>15.9</b>
SHOT (Liang et al., 2020)	$\times$	$\times$	<b>17.1</b>
FR	$\times$	$\times$	<b>16.8</b>
BUFR	$\times$	$\times$	<b>16.2</b>
No corruption	$\checkmark$	$\times$	99.6
Source-only	$\checkmark$	$\times$	47.0
AdaBN (Li et al., 2017)	$\checkmark$	$\times$	65.2
SHOT (Liang et al., 2020)	$\checkmark$	$\times$	<b>82.9</b>
FR	$\checkmark$	$\times$	73.7
BUFR	$\checkmark$	$\times$	72.9
No corruption	$\checkmark$	$\checkmark$	99.7
Source-only	$\checkmark$	$\checkmark$	44.6
AdaBN (Li et al., 2017)	$\checkmark$	$\checkmark$	68.7
SHOT (Liang et al., 2020)	$\checkmark$	$\checkmark$	<b>85.0</b>
FR	$\checkmark$	$\checkmark$	82.8
BUFR	$\checkmark$	$\checkmark$	83.1

Published as a conference paper at ICLR 2022

## E FURTHER RELATED WORK

**Domain generalization.** Domain generalization seeks to do well in the target domain *without updating the source model*. The goal is to achieve this through suitable data augmentation, self-supervision, and inductive biases with respect to a perturbation of interest (Simard et al., 1991; Engstrom et al., 2019; Michaelis et al., 2019; Roy et al., 2019; Djolonga et al., 2021). One may view this as specifying the shifts that a model should be robust to *a priori*. Practically, however, we generally do not know what shift will occur upon deployment—there will always be unseen shifts. Furthermore, the condition that our augmented development process be sufficiently diverse is untestable—with the worst-case error still being arbitrarily high (David et al., 2010; Arjovsky et al., 2019). Permitting adaptation in the target domain is one reasonable solution to these problems.

**Common corruptions.** Previous works (Hendrycks & Dietterich, 2019) have used *common corruptions* to study the robustness of neural networks to simple transformations of the input, e.g. Gaussian noise (common in low-lighting conditions), defocus blur (camera is not properly focused or calibrated), brightness (variations in daylight intensity), and impulse noise (colour analogue of salt-and-pepper noise, caused by bit errors). We see common corruptions as one particular type of measurement shift, with all the aforementioned corruptions arising from a change in measurement system. However, not all measurement shifts are common corruptions. For example, the right column of Figure 1c depicts tissue slides from different hospitals. Here, the shift has arisen from changes in slide-staining procedures, patient populations and image acquisition (e.g. different sensing equipment). This measurement shift cannot be described in terms of simple input transformations like Gaussian noise or blurring, and thus we do not consider it a common corruption. In addition, EMNIST-DA shifts like bricks and grass use knowledge of the object type (i.e. a digit) to change the background and foreground separately (see Figure 7). We do not consider these to be common corruptions as common corruptions rarely have knowledge of the image content—e.g. blurring all pixels or adding noise randomly. In summary, we consider measurement shifts to be a superset of common corruptions, thus warranting their own definition.

**SFDA and related settings.** Table 7 compares the setting of SFDA to the related settings of fine-tuning, unsupervised domain adaptation (UDA), and domain generalization (DG).

Table 7: Source-free domain adaptation and related settings. Adapted from Wang et al. (2021).

Setting	Source data	Target data	Adapt. Loss
Fine-tuning	-	$x^t, y^t$	$L(x^t, y^t)$
UDA	$x^s, y^s$	$x^t$	$L(x^s, y^s) + L(x^s, x^t)$
Domain gen.	$x^s, y^s$	-	$L(x^s, y^s)$
Source-free DA	-	$x^t$	$L(x^t)$

## F DATASETS

Figures 5, 6, 7, 8 and 9 below visualize the different datasets we use for evaluation and analysis.

MNIST-M (Ganin et al., 2016) is constructed by combining digits from MNIST with random background colour patches from BSDS500 (Arbelaez et al., 2011). The source domain is standard MNIST and the target domain is the same digits coloured (see Figure 5). MNIST-C (Mu & Gilmer, 2019) contains 15 different corruptions of the MNIST digits. Again, the source domain is standard MNIST and the corruptions of the same digits make up the 15 possible target domains (see Figure 6).

As shown in Appendix K.1 many methods achieve good performance on these MNIST datasets. For this reason we create and release the more challenging EMNIST-DA dataset. EMNIST-DA contains 13 different shifts chosen to give a diverse range of initial accuracies when using a source model trained on standard EMNIST. In particular, a number of shifts result in very low initial performance but are conceptually simple to resolve (see Figure 7). Here, models are trained on the training set of EMNIST (source) before being adapted to a shifted test set of EMNIST-DA (target, unseen examples).

We also use the CIFAR-10-C and CIFAR-100-C corruption datasets (Hendrycks & Dietterich, 2019) to compare methods on object-recognition tasks. These datasets contain 19 different corruptions of the CIFAR-10 and CIFAR-100 test sets (see Figure 8). Here, a model is trained on the training set of CIFAR-10/CIFAR-100 (source, Krizhevsky 2009) before being adapted to a corrupted test set (target).

Finally, we show real-world measurement shift with CAMELYON<sub>17</sub> (Bandi et al., 2018), a medical dataset with histopathological images from 5 different hospitals which use different staining and imaging techniques (Figure 9). The goal is to determine whether or not an image contains tumour tissue. We train on examples from a single source hospital (hospital 3) before adapting to one of the 4 remaining target hospitals. We use the WILDS (Koh et al., 2021) implementation of CAMELYON<sub>17</sub>.



Figure 5: *Top*: samples from MNIST. *Bottom*: samples from MNIST-M.

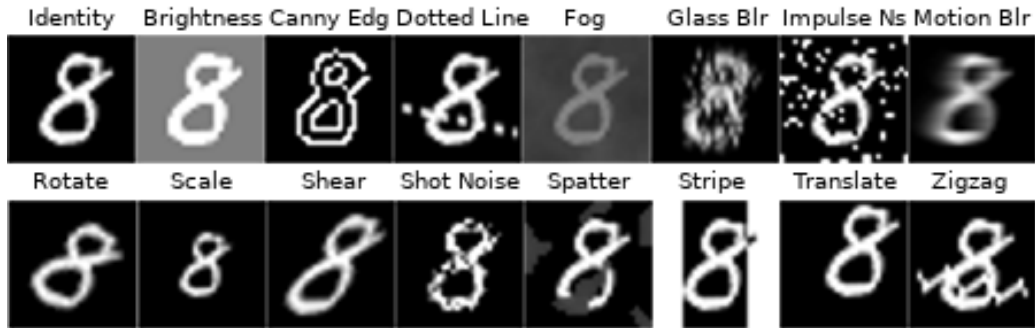


Figure 6: MNIST-C corruptions.

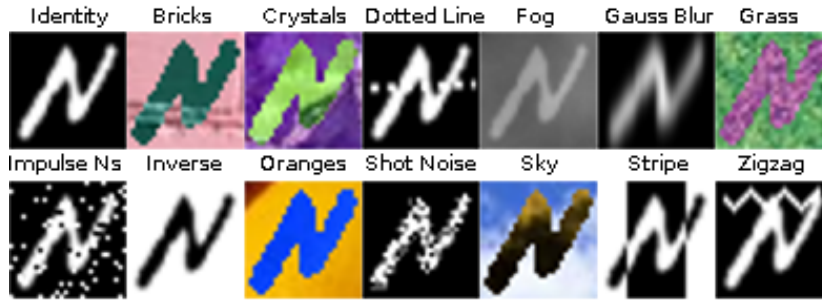


Figure 7: EMNIST-DA shifts.



Figure 8: CIFAR corruptions. The same corruptions are used for CIFAR-10-C and CIFAR-100-C.

Published as a conference paper at ICLR 2022

---

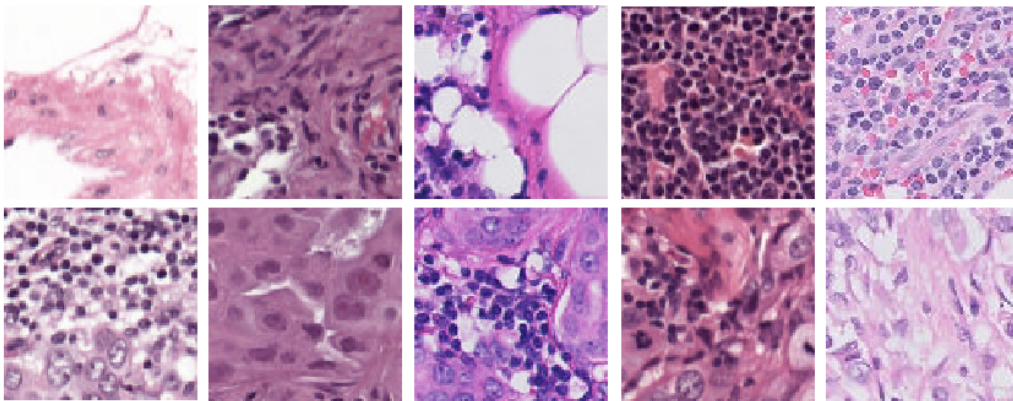


Figure 9: CAMELYON17. Columns show different hospitals. *Top row*: no tumour tissue. *Bottom row*: tumour tissue present.

Published as a conference paper at ICLR 2022

## G FURTHER IMPLEMENTATION DETAILS

**Architectures.** The architecture of the simple 5-layer CNN (a variant of LeNet, LeCun et al. 1998), which we use for digit and character datasets, is provided in Table 8. For the object-recognition and medical datasets, we use a standard ResNet-18 (He et al., 2016).

**Training details.** For all datasets and methods we train using SGD with momentum set to 0.9, use a batch size of 256, and report results over 5 random seeds. In line with previous UDA & SFDA works (although often not made explicit), we use a test-domain validation set for model selection (Gulrajani & Lopez-Paz, 2021). In particular, we select the best-performing learning rate from  $\{0.0001, 0.001, 0.01, 0.1, 1\}$ , and for BUFR, we train for 30 epochs per block and decay the learning rate as a function of the number of unfrozen blocks in order to further maintain structure. For all other methods, including FR, we train for 150 epochs with a constant learning rate. The temperature parameter  $\tau$  (see Appendix A, Eq. 4) is set to 0.01 in all experiments.

**Tracking feature and logit distributions.** To track the marginal feature and logit distributions, we implement a simple `StatsLayer` class in PyTorch that can be easily inserted into a network just like any other layer. This seamlessly integrates distribution-tracking into standard training processes. In the source domain, we simply: (i) add `StatsLayers` to our (pre)trained source model; (ii) pass the source data through the model; and (iii) save the model as normal in PyTorch (the tracked statistics, i.e. bin counts, are automatically saved as persistent buffers akin to BN-statistics). In the target domain, the source model can be loaded as normal and the inserted `StatsLayers` will contain the source-data statistics. Code is available at <https://github.com/cianeastwood/bufr>.

**The Full Gauss. baseline.** This baseline models the distribution of hidden features as a joint multivariate Gaussian, with dimensionality equal to the number of hidden units. After training a model on the source data, the source data is passed through once more and the empirical mean vector and covariance matrix are calculated and saved. To adapt to the target data the empirical mean and covariances are calculated for each minibatch and the distributions are aligned using the KL divergence  $D_{KL}(Q||P)$ , where  $Q$  is the Gaussian distribution estimated on the target data minibatch and  $P$  from the source data. This divergence has an analytic form (Duchi, 2007, Sec. 9) which we use as the loss function. We use this direction for the KL divergence as we only need to invert the covariance matrix once (for saved  $P$ ) rather than the covariance matrix for  $Q$  on every batch.

**Online setup.** In the online setting, where only a single epoch is permitted, we find that all methods are very sensitive to the learning rate (unsurprising, given that most methods will not have converged after a single epoch). For fair comparison, we thus search over learning rates in  $\{0.1, 0.01, 0.001, 0.0001\}$  for all methods, choosing the best-performing one. Additionally, when learning speed is of critical importance, we find it beneficial to slightly increase  $\tau$ . We thus set  $\tau = 0.05$  for all online experiments, compared to 0.01 for all “offline” experiments.

Table 8: Architecture of the CNN used on digit and character datasets. For conv. layers, the weights-shape is: *num. input channels*  $\times$  *num. output channels*  $\times$  *filter height*  $\times$  *filter width*.

Block	Weights-Shape	Stride	Padding	Activation	Dropout Prob.
Conv + BN	$3 \times 64 \times 5 \times 5$	2	2	ReLU	0.1
Conv + BN	$64 \times 128 \times 3 \times 3$	2	2	ReLU	0.3
Conv + BN	$128 \times 256 \times 3 \times 3$	2	2	ReLU	0.5
Linear + BN	$6400 \times 128$	N/A	N/A	ReLU	0.5
Linear	$128 \times \text{Number of Classes}$	N/A	N/A	Softmax	0



Published as a conference paper at ICLR 2022

## H RELIABILITY DIAGRAMS AND CONFIDENCE HISTOGRAMS

This section shows reliability diagrams (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005) and confidence histograms (Zadrozny & Elkan, 2001): (i) over all EMNIST-DA shifts (see Figure 10); (ii) a severe EMNIST-DA shift (see Figure 11); and (iii) a mild shift EMNIST-DA shift (see Figure 12). Reliability diagrams are given along with the corresponding Expected Calibration Error (ECE, Naeini et al. 2015) and Maximum Calibration Error (MCE, Naeini et al. 2015). ECE is calculated by binning predictions into 10 evenly-spaced bins based on confidence, and then taking a weighted average of the absolute difference between average accuracy and average confidence of the samples in each bin. MCE is the maximum absolute difference between average accuracy and average confidence over the bins. In Figures 10–12 below, we pair each reliability diagram with the corresponding confidence histogram, since reliability diagrams do not provide the underlying frequencies of each bin (as in Guo et al. 2017, Figure 1).

In general we see that most models are overconfident, but our models much less so. As seen by the difference in the size of the red ‘Gap’ bar in the rightmost bins of Figures 10b, 10c, and 10d, when our FR methods predict with high confidence they are much more likely to be correct than IM—a method which works by maximizing prediction confidence. Figure 11 shows that BUFR remains well-calibrated even when the initial shift is severe. Figure 12 shows that, even for a mild shift when all models achieve high accuracy, our methods are better-calibrated. Note that the label ‘Original’ in Figures 10a and 10e denotes the source model on the *source data*, while ‘Source-only’ in Figures 11a, 11e, 12a, and 12e denotes the source model on the *target data*.

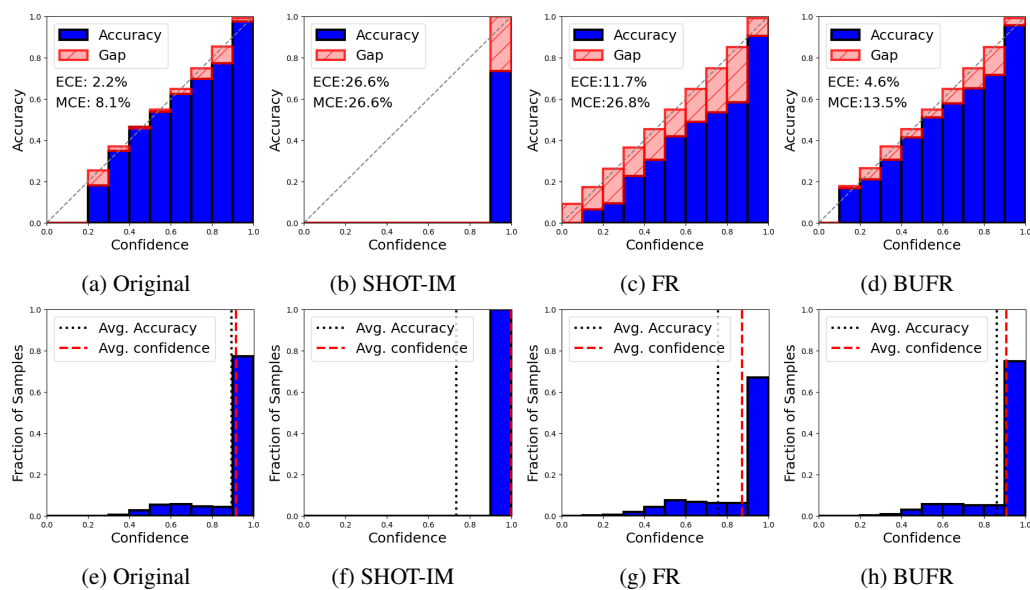


Figure 10: Reliability diagrams and confidence histograms over *all* EMNIST-DA corruptions. (a–d): Reliability diagrams showing the difference between average accuracy and average confidence for different methods. (e–h): Confidence histograms showing the frequency with which predictions are made with a given confidence. Each confidence histogram corresponds with the reliability diagram above it. (a & e): The source model is well-calibrated on the *source data*. (b & f): Entropy-minimization leads to extreme overconfidence. (c & g, d & h): Our methods, FR and BUFR, are much better-calibrated as they do not work by making predictions more confident.

Published as a conference paper at ICLR 2022

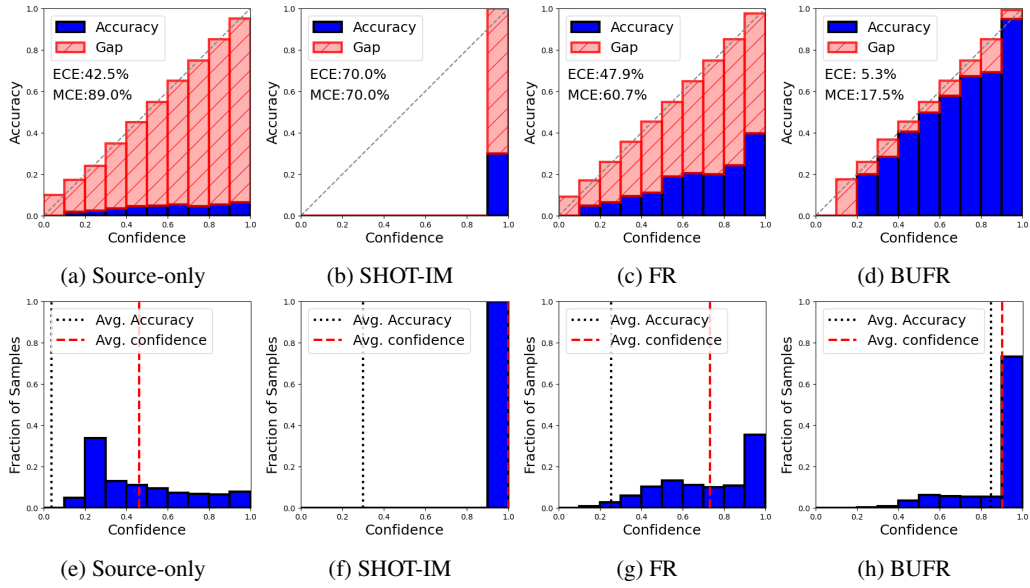


Figure 11: Reliability diagrams and confidence histograms for a severe EMNIST-DA shift (sky) where all methods except BUFR achieve poor accuracy. Each confidence histogram corresponds with the reliability diagram above it. (a & e): Source model on the *target data* achieves poor accuracy and often predicts with low confidence. (b & f): SHOT-IM also achieves poor accuracy but is highly confident. (d & h): Our BUFR method achieves better ECE and MCE than all other methods.

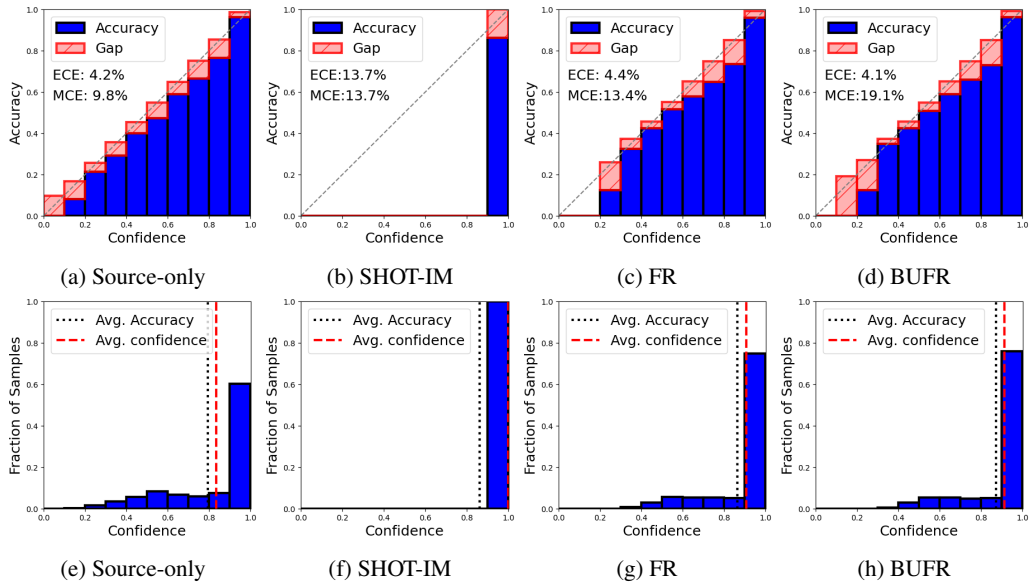


Figure 12: Reliability diagrams and confidence histograms for a mild EMNIST-DA shift (shot noise) where all methods achieve good accuracy. Each confidence histogram corresponds with the reliability diagram above it. When highly confident, our methods (d & h) are more often correct than IM (b & f).

Published as a conference paper at ICLR 2022

## I ACTIVATION DISTRIBUTIONS

**EMNIST-DA (skewed).** Figure 13 depicts histograms of the marginal feature and logit activation-distributions on the EMNIST-DA stripe shift. As shown, the marginal distributions on the source data (blue curve, those we wish to match) may be heavily-skewed. In contrast, the marginal distributions on the target data (*before adapting*, orange curve) tend to be more symmetric but have a similar mean.

**CIFAR-10 (bi-modal).** Figure 14 depicts histograms of the marginal feature and logit activation-distributions on the CIFAR-10-C impulse-noise shift. As shown, the marginal distributions on the source data (blue curve, those we wish to match) tend to be bi-modal. In contrast, the marginal distributions on the target data (*before adapting*, orange curve) tend to be uni-modal but have a similar mean. The two modes can be interpreted intuitively as “detected” and “not detected” or “present” and “not present” for a given feature-detector.

**Alignment after adapting.** Figure 15 shows histograms of the marginal feature activation-distributions on the EMNIST-DA stripe shift. This figure shows curves on the source data (blue curve, same as Figure 13a) and on the target data (*after adapting*, orange curve) for different methods. Evidently, our FR loss causes the marginal distributions to closely align (Figure 15c). In contrast, competing methods (Figures 15a, 15b) do not match the feature activation-distributions, even if they achieve high accuracy. Figure 16 shows the same trend for CIFAR-10-C.

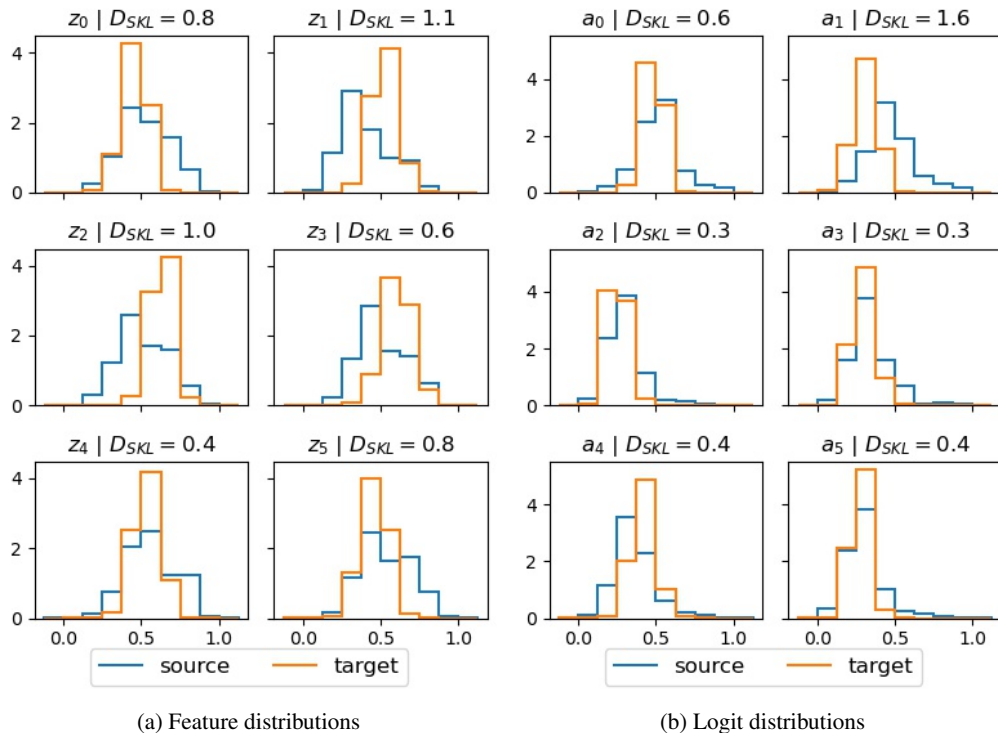


Figure 13: Histograms showing the first 6 marginal activation-distributions on the EMNIST-DA stripe shift. The blue curves are the saved marginal distributions under the source data (i.e. EMNIST). The orange curves are the marginal distributions under the target data *before adaptation* (i.e. the stripe shift). (a) Marginal feature activation-distributions. (b) Marginal logit activation-distributions.  $D_{SKL}$  denotes the symmetric KL divergence.

Published as a conference paper at ICLR 2022

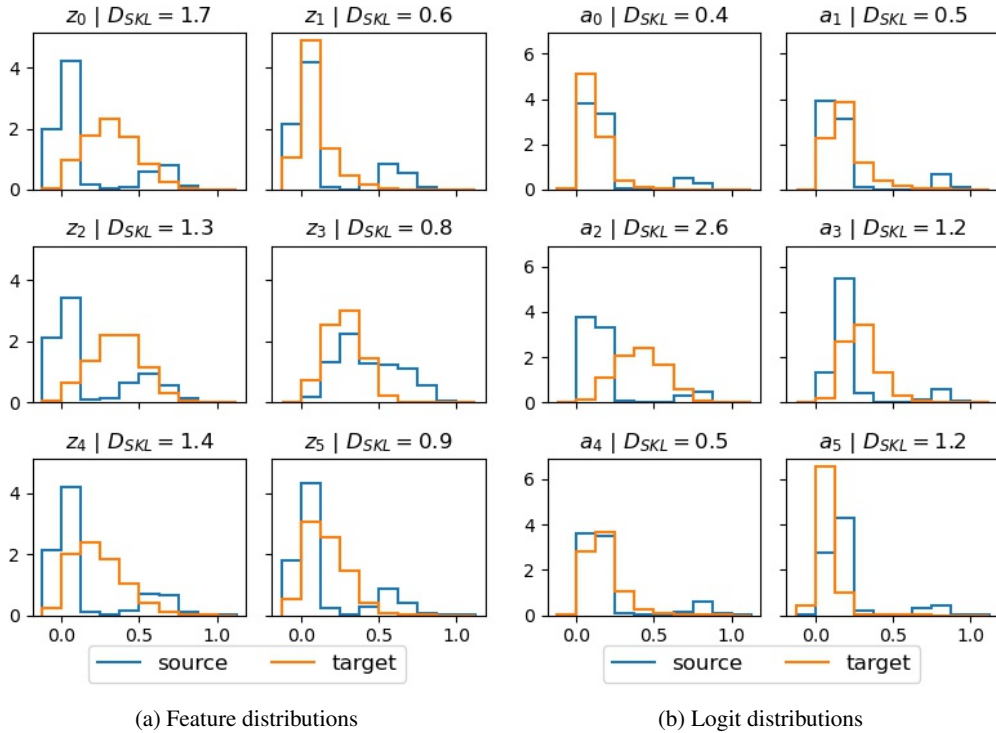


Figure 14: Histograms showing the first 6 marginal activation-distributions on the CIFAR-10-C impulse-noise shift. The blue curves are the saved marginal distributions under the source data (i.e. CIFAR-10). The orange curves are the marginal distributions under the target data *before adaptation* (i.e. the impulse-noise shift). (a) Marginal feature activation-distributions and (b) Marginal logit activation-distributions.  $D_{SKL}$  denotes the symmetric KL divergence.

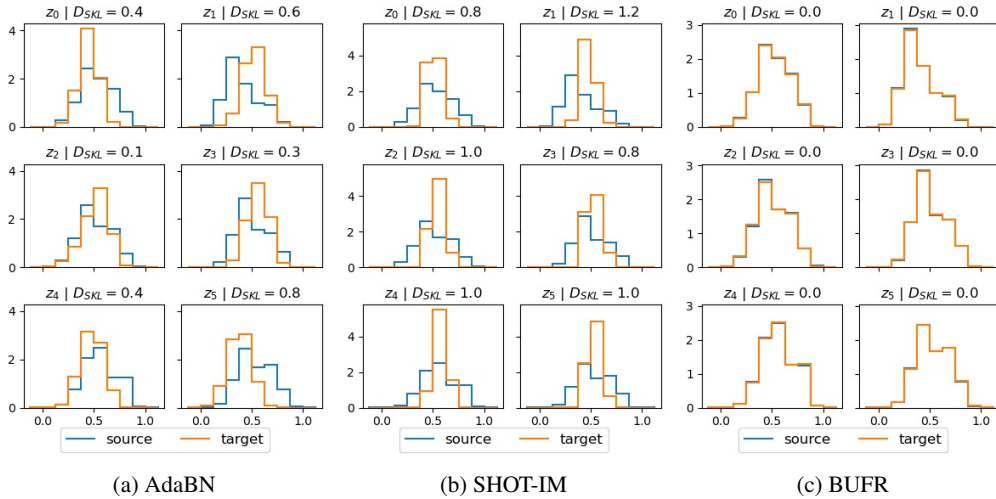


Figure 15: Histograms showing distribution-alignment on the EMNIST-DA stripe shift. The blue curves are the saved marginal distributions under the source data (i.e. EMNIST). The orange curves are the marginal distributions under the target data *after adaptation* (to the stripe shift). (a,b): AdaBN and SHOT-IM do not align the marginal distributions (despite achieving reasonable accuracy—see Table 17). (c) BUFR matches the activation-distributions very closely, making  $D_{SKL}$  very small.

Published as a conference paper at ICLR 2022

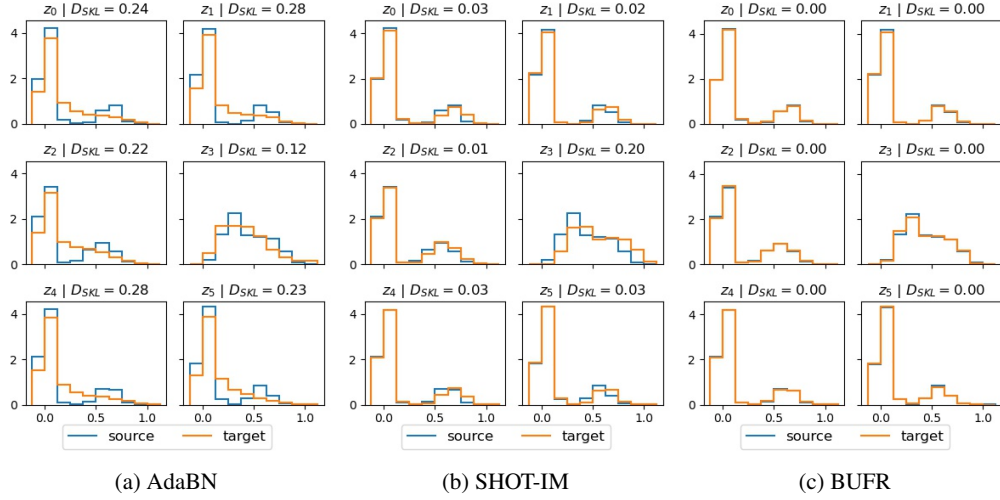


Figure 16: Histograms showing distribution-alignment on the CIFAR-10-C impulse-noise shift. The blue curves are the saved marginal distributions under the source data (i.e. CIFAR-10). The orange curves are the marginal distributions under the target data *after adaptation* (to the impulse-noise shift). (a) AdaBN does not align the marginal distributions. (b) SHOT-IM only partially-aligns the marginal distributions. (c) BUFR matches the activation-distributions very closely, making  $D_{SKL}$  very small.

## J FURTHER ANALYSIS

### J.1 EFFICACY OF BOTTOM-UP TRAINING

Table 9 reports EMNIST-DA accuracy vs. the number of (unlabelled) examples-per-class available in the target domain. BUFR retains strong performance even with only 5 examples-per-class.

Table 9: EMNIST-DA accuracy vs. examples-per-class.

Model	5	10	20	50	400
Marg. Gauss. (Ishii & Sugiyama, 2021)	49.3	49.9	50.4	50.7	50.6
Full Gauss.	55.4	59.7	61.0	63.3	68.3
PL (Lee et al., 2013)	45.8	46.3	46.0	46.7	49.7
BNM-IM (Ishii & Sugiyama, 2021)	50.5	51.5	53.0	54.7	61.4
SHOT-IM (Liang et al., 2020)	48.3	51.7	51.2	54.7	73.4
FR (ours)	50.8	50.5	60.1	63.1	75.6
BUFR (ours)	<b>78.0</b>	<b>82.3</b>	<b>83.8</b>	<b>84.9</b>	<b>86.2</b>

### J.2 LOSS ABLATION STUDY

Table 10 reports the performance of our FR loss on CIFAR-10-C and CIFAR-100-C without: (i) aligning the logit distributions; and (ii) using the symmetric KL divergence (we instead use the asymmetric reverse KL). While these components make little difference on the easier task of CIFAR-10-C, they significantly improve performance on the harder task of CIFAR-100-C.

Table 10: Ablation study of  $\mathcal{L}_{tgt}$  in Eq. 2.

Model	CFR-10-C	CFR-100-C
$\mathcal{L}_{tgt}$ w/o logits	86.7 $\pm$ 0.2	62.3 $\pm$ 1.3
$\mathcal{L}_{tgt}$ w/o $D_{KL}(P  Q)$	86.5 $\pm$ 0.3	61.5 $\pm$ 0.2
$\mathcal{L}_{tgt}$	<b>87.2 <math>\pm</math> 0.7</b>	<b>65.5 <math>\pm</math> 0.2</b>

Published as a conference paper at ICLR 2022

### J.3 WHO IS AFFECTED

We now analyse which layers are most affected by a measurement shift. Figure 17 shows the (symmetric) KL divergence between the unit-level activation distributions under the source (EMNIST) and target (EMNIST-DA crystals) data *before adapting* (17a) and *after adapting the first layer* (17b). Figure 17a shows that, before adapting, the unit-activation distributions in all layers of the network have changed significantly, as indicated by the large KL divergences. Figure 17b shows that, after updating just the first layer, “normality” is restored in all subsequent layers, with the unit-level activation distributions on the target data realigning with those saved on the source (shown via very low KL divergences). This indicates that measurement shifts primarily affect the first layer/block—since they can be mostly resolved by updating the first layer/block—and also further motivates bottom-up training for measurement shifts.

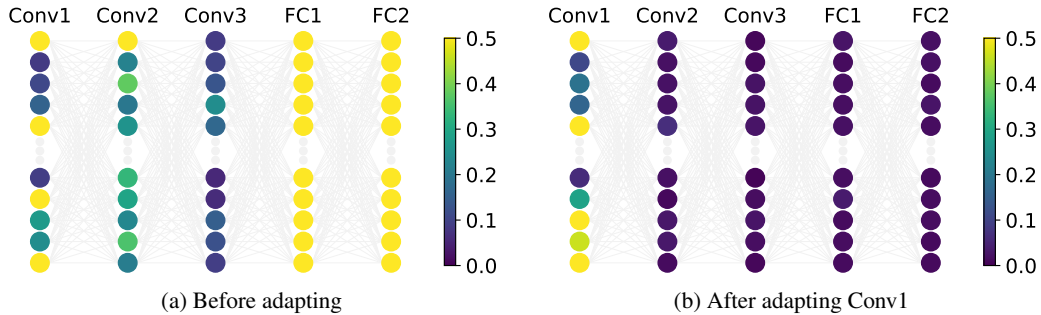


Figure 17: Symmetric KL divergence between the unit-level activation distributions under the source (EMNIST) and target (EMNIST-DA crystals) data: (a) before adapting; and (b) after adapting only the first layer (Conv1). For visual clarity, we show only 10 sample units per layer.

### J.4 WHO MOVES

We now analyse which layers are most updated by BUFR. Figure 18a shows that, on average, FR moves the weights of all layers of  $g_t$  a similar distance when adapting to the target data. Figure 18b shows that BUFR primarily updates the early layers, thus preserving learnt structure in later layers.

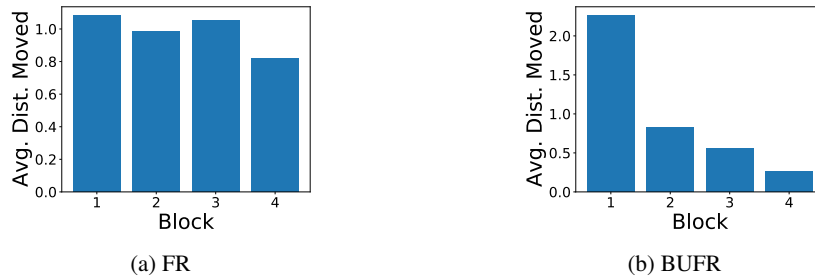


Figure 18: Average distance moved by a unit in each block of  $g_t$  on the EMNIST-DA stripe shift when training (a) all layers at once and (b) in a bottom-up manner. Both methods are trained with the same constant learning rate.

Published as a conference paper at ICLR 2022

## K FULL RESULTS

In this section we give the full results for all datasets and constituent domains.

### K.1 DIGIT AND CHARACTER SUMMARY RESULTS

The simplest datasets we use are variations of the MNIST dataset (LeCun et al., 1998). Here, a model is trained on MNIST (source domain) before being adapted to MNIST-M (Ganin et al., 2016) or one of the fifteen MNIST-C (Mu & Gilmer, 2019) corruptions (target domain). As mentioned in Section 5, the MNIST-based shifts can be well-resolved by a number of methods.

Tables 11 and 12 summarize the accuracy and ECEs across different models for the digit and character datasets. On MNIST-C, where source-only accuracy is very high, all methods achieve good results (accuracy  $\geq 95\%$ )—providing limited insight into their relative performances. On MNIST-M, our BUFR method outperforms all baselines, although SHOT is very similar in performance. As discussed in Section 5, our BUFR method outperforms all baseline methods on EMNIST-DA in terms of accuracy *and* ECE as it does not work by making predictions more confident.

Table 11: Digit and character accuracy (%) results. Shown are the mean and 1 standard deviation. EMNIST-DA: mean performance over all 13 EMNIST-DA shifts. EMNIST-DA-SVR & EMNIST-DA-MLD: sample “severe” and “mild” shifts from EMNIST-DA selected based on AdaBN performance.

Model	MNIST-C	MNIST-M	EMNIST-DA	EMNIST-DA-SVR	EMNIST-DA-MLD
No corruption	99.5 $\pm$ 0.1	99.5 $\pm$ 0.1	89.4 $\pm$ 0.1	89.4 $\pm$ 0.1	89.4 $\pm$ 0.1
Source-only	86.2 $\pm$ 1.8	42.7 $\pm$ 4.6	29.5 $\pm$ 0.5	3.8 $\pm$ 0.4	78.5 $\pm$ 0.7
AdaBN (Li et al., 2018)	94.2 $\pm$ 0.2	59.1 $\pm$ 1.9	46.2 $\pm$ 1.1	3.7 $\pm$ 0.7	84.9 $\pm$ 0.2
PL (Lee et al., 2013)	96.4 $\pm$ 0.4	43.1 $\pm$ 2.1	50.0 $\pm$ 0.6	2.7 $\pm$ 0.4	83.5 $\pm$ 0.1
SHOT-IM (Liang et al., 2020)	97.3 $\pm$ 0.2	66.9 $\pm$ 9.3	70.3 $\pm$ 3.7	24.0 $\pm$ 7.5	86.3 $\pm$ 0.1
SHOT (Liang et al., 2020)	<b>97.7 <math>\pm</math> 0.2</b>	94.4 $\pm$ 3.1	80.0 $\pm$ 4.4	55.1 $\pm$ 23.5	86.1 $\pm$ 0.1
FR (ours)	96.7 $\pm$ 0.1	86.5 $\pm$ 0.6	74.4 $\pm$ 0.8	15.3 $\pm$ 6.8	86.4 $\pm$ 0.1
BUFR (ours)	96.4 $\pm$ 0.6	<b>96.2 <math>\pm</math> 1.7</b>	<b>86.1 <math>\pm</math> 0.1</b>	<b>84.6 <math>\pm</math> 0.2</b>	<b>87.0 <math>\pm</math> 0.2</b>
Target-supervised	99.3 $\pm$ 0.0	98.5 $\pm$ 0.0	86.8 $\pm$ 0.6	85.7 $\pm$ 0.6	87.3 $\pm$ 0.7

Table 12: Digit and character ECE (%) results. Shown are the mean and 1 standard deviation. EMNIST-DA: mean performance over all 13 EMNIST-DA shifts. EMNIST-DA-SVR & EMNIST-DA-MLD: sample “severe” and “mild” shifts from EMNIST-DA selected based on AdaBN performance.

Model	MNIST-C	MNIST-M	EMNIST-DA	EMNIST-DA-SVR	EMNIST-DA-MLD
No corruption	0.3 $\pm$ 0.0	0.3 $\pm$ 0.0	2.3 $\pm$ 0.1	2.3 $\pm$ 0.1	2.3 $\pm$ 0.1
Source-only	7.2 $\pm$ 1.4	27.0 $\pm$ 7.1	30.8 $\pm$ 1.6	42.6 $\pm$ 3.5	4.8 $\pm$ 0.5
AdaBN (Li et al., 2018)	4.1 $\pm$ 0.1	24.4 $\pm$ 2.8	30.3 $\pm$ 1.1	52.4 $\pm$ 4.9	4.9 $\pm$ 0.3
PL (Lee et al., 2013)	3.1 $\pm$ 0.4	56.1 $\pm$ 2.2	49.9 $\pm$ 0.6	97.2 $\pm$ 0.4	16.4 $\pm$ 0.1
SHOT-IM (Liang et al., 2020)	2.3 $\pm$ 0.2	30.9 $\pm$ 9.0	29.6 $\pm$ 3.7	76.0 $\pm$ 7.5	13.7 $\pm$ 0.1
SHOT (Liang et al., 2020)	<b>2.0 <math>\pm</math> 0.2</b>	<b>2.8 <math>\pm</math> 2.9</b>	19.7 $\pm$ 4.4	42.7 $\pm$ 23.0	14.8 $\pm$ 0.1
FR (ours)	2.5 $\pm$ 0.2	9.8 $\pm$ 0.8	12.9 $\pm$ 0.9	58.0 $\pm$ 6.8	4.6 $\pm$ 0.3
BUFR (ours)	3.0 $\pm$ 0.6	2.9 $\pm$ 1.5	<b>4.7 <math>\pm</math> 0.2</b>	<b>5.6 <math>\pm</math> 0.3</b>	<b>4.2 <math>\pm</math> 0.2</b>
Target-supervised	0.5 $\pm$ 0.0	1.1 $\pm$ 0.1	7.3 $\pm$ 0.7	7.0 $\pm$ 0.5	8.4 $\pm$ 1.1

Published as a conference paper at ICLR 2022

## K.2 ONLINE RESULTS

Table 13 reports the online results for CIFAR-10-C and CIFAR-100-C. FR outperforms existing SFDA methods on CIFAR-10-C in terms of both accuracy and ECE. On CIFAR-100-C, our method is competitive with TENT (Wang et al., 2021)—a method designed specifically for this online setting. As in Wang et al. (2021), these results represent the average over batches *during training* (i.e. a single pass through the target data), rather than the average at the end of training, in order to evaluate *online* performance. We omit BUFR from this table as it is not easily applicable to the online setting—it is difficult to set the number of steps per block without information on the total number of steps/batches (generally not available in an online setting). Full per-shift results for this online setting are given in Tables 23 and 24 for CIFAR-10-C, and Tables 25 and 26 for CIFAR-100-C.

Table 13: Online results. Shown are the mean and 1 standard deviation.

Model	CIFAR-10-C		CIFAR-100-C	
	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$
AdaBN (Li et al., 2018)	80.3 $\pm$ 0.0	12.1 $\pm$ 0.0	56.6 $\pm$ 0.3	<b>10.0 <math>\pm</math> 0.1</b>
SHOT-IM (Liang et al., 2020)	83.2 $\pm$ 0.2	10.9 $\pm$ 0.1	62.3 $\pm$ 0.3	13.8 $\pm$ 0.1
TENT (Wang et al., 2021)	81.8 $\pm$ 0.2	11.5 $\pm$ 0.1	<b>63.1 <math>\pm</math> 0.3</b>	14.3 $\pm$ 0.1
FR (ours)	<b>85.9 <math>\pm</math> 0.3</b>	<b>9.5 <math>\pm</math> 0.2</b>	62.7 $\pm$ 0.3	13.6 $\pm$ 0.1

K.3 CAMELYON<sub>17</sub> RESULTS

Table 14 reports the accuracy and ECE results for CAMELYON<sub>17</sub>. With up to 50 target examples-per-class: (i) our methods reduce the error rate by approximately 20% compared to the next best method; (ii) only our methods meaningfully improve upon the simple AdaBN baseline which uses the target-data BN-statistics (i.e. neither PL or SHOT-IM actually work). With up to 500 target examples-per-class, our methods reduce the error rate by approximately 20% compared to the next best method. With over 15,000 examples-per-class, our methods are competitive with existing ones.

Table 14: CAMELYON<sub>17</sub> results for different numbers of (unlabelled) examples-per-class in the target domain.

Model	5		50		500		>15k	
	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$	ACC $\uparrow$	ECE $\downarrow$
Source-only	55.8 $\pm$ 1.6	40.8 $\pm$ 2.1	55.8 $\pm$ 1.6	40.8 $\pm$ 2.1	55.8 $\pm$ 1.6	40.8 $\pm$ 2.1	55.8 $\pm$ 1.6	40.8 $\pm$ 2.1
AdaBN	82.6 $\pm$ 2.2	14.7 $\pm$ 2.1	83.7 $\pm$ 1.0	13.7 $\pm$ 0.8	83.9 $\pm$ 0.8	13.5 $\pm$ 0.7	84.0 $\pm$ 0.5	13.5 $\pm$ 0.5
PL	82.5 $\pm$ 2.0	14.2 $\pm$ 1.1	83.6 $\pm$ 1.2	13.8 $\pm$ 1.0	85.0 $\pm$ 0.8	13.0 $\pm$ 0.8	<b>90.6 <math>\pm</math> 0.9</b>	<b>8.8 <math>\pm</math> 0.9</b>
SHOT-IM	82.6 $\pm$ 2.2	13.8 $\pm$ 1.8	83.7 $\pm$ 1.2	13.8 $\pm$ 1.1	86.4 $\pm$ 0.7	11.9 $\pm$ 0.7	89.9 $\pm$ 0.2	9.7 $\pm$ 0.2
FR (ours)	<b>84.6 <math>\pm</math> 0.6</b>	12.9 $\pm$ 0.5	86.0 $\pm$ 1.1	12.1 $\pm$ 1.1	89.0 $\pm$ 0.6	<b>9.7 <math>\pm</math> 0.6</b>	89.5 $\pm$ 0.4	9.8 $\pm$ 0.5
BUFR (ours)	84.5 $\pm$ 0.8	<b>12.8 <math>\pm</math> 0.8</b>	<b>87.0 <math>\pm</math> 1.2</b>	<b>11.1 <math>\pm</math> 1.1</b>	<b>89.1 <math>\pm</math> 0.8</b>	<b>9.7 <math>\pm</math> 0.8</b>	89.7 $\pm$ 0.5	9.6 $\pm$ 0.6



Published as a conference paper at ICLR 2022

## K.4 MNIST-C FULL RESULTS

Tables 15 and 16 show the accuracy and ECE results for each individual corruption of the MNIST-C dataset. We provide the average performance with and without the translate corruption as the assumptions behind the methods that rely on a fixed classifier  $h$  no longer hold. Without the translate corruption (Avg. \translate) we see that all methods achieve high accuracy ( $\geq 95\%$ ).

Table 15: MNIST-C accuracy (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	PL	SHOT-IM	SHOT	FR	BUFR
Brightness	84.8 ± 11.4	99.4 ± 0.0	99.5 ± 0.0	99.4 ± 0.1	99.5 ± 0.1	98.7 ± 0.1	99.2 ± 0.1
Canny Edges	72.2 ± 0.8	91.0 ± 0.7	96.2 ± 1.0	98.1 ± 0.1	98.6 ± 0.1	97.8 ± 0.1	98.5 ± 0.0
Dotted Line	98.6 ± 0.2	98.8 ± 0.2	99.3 ± 0.1	99.4 ± 0.1	99.4 ± 0.1	98.6 ± 0.1	99.1 ± 0.0
Fog	30.1 ± 12.5	93.9 ± 1.9	99.3 ± 0.0	99.4 ± 0.1	99.5 ± 0.0	98.6 ± 0.1	99.2 ± 0.0
Glass Blur	88.9 ± 2.3	95.3 ± 0.3	97.8 ± 0.1	98.2 ± 0.1	98.3 ± 0.0	97.2 ± 0.1	97.9 ± 0.0
Impulse Noise	95.2 ± 0.6	97.9 ± 0.1	98.4 ± 0.1	98.7 ± 0.1	98.9 ± 0.1	97.8 ± 0.0	98.6 ± 0.0
Motion Blur	85.6 ± 3.9	97.3 ± 0.4	98.8 ± 0.1	99.1 ± 0.1	99.2 ± 0.1	98.1 ± 0.1	98.8 ± 0.1
Rotate	96.7 ± 0.1	96.7 ± 0.0	97.7 ± 0.1	98.4 ± 0.1	98.8 ± 0.0	97.5 ± 0.1	97.9 ± 0.1
Scale	97.2 ± 0.1	97.2 ± 0.1	98.7 ± 0.1	99.1 ± 0.0	99.2 ± 0.0	98.0 ± 0.0	98.7 ± 0.2
Shear	98.9 ± 0.1	98.9 ± 0.0	99.0 ± 0.0	99.1 ± 0.0	99.2 ± 0.0	98.3 ± 0.1	98.8 ± 0.1
Shot Noise	98.6 ± 0.0	99.0 ± 0.0	99.2 ± 0.0	99.2 ± 0.1	99.2 ± 0.0	98.3 ± 0.2	99.0 ± 0.1
Spatter	98.7 ± 0.1	98.8 ± 0.1	99.0 ± 0.1	99.0 ± 0.0	99.1 ± 0.1	98.4 ± 0.1	98.8 ± 0.0
Stripe	91.1 ± 1.2	90.9 ± 1.5	97.9 ± 1.0	99.2 ± 0.0	99.4 ± 0.1	98.3 ± 0.1	99.1 ± 0.1
Translate	64.6 ± 0.5	64.4 ± 0.6	69.5 ± 0.8	75.1 ± 4.1	78.7 ± 3.1	76.7 ± 2.1	64.5 ± 8.9
Zigzag	91.8 ± 0.6	93.0 ± 0.2	98.2 ± 0.2	98.9 ± 0.1	99.2 ± 0.1	98.2 ± 0.1	98.8 ± 0.1
Avg.	86.2 ± 1.8	94.2 ± 0.2	96.6 ± 0.1	97.3 ± 0.2	97.7 ± 0.2	96.7 ± 0.1	96.4 ± 0.6
Avg.\translate	87.7 ± 1.9	96.3 ± 0.2	98.5 ± 0.1	98.9 ± 0.0	99.1 ± 0.0	98.1 ± 0.1	98.7 ± 0.0

Table 16: MNIST-C ECE (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	PL	SHOT-IM	SHOT	FR	BUFR
Brightness	2.4 ± 0.4	0.3 ± 0.1	0.3 ± 0.1	0.4 ± 0.1	0.9 ± 0.1	0.9 ± 0.1	0.5 ± 0.1
Canny Edges	22.2 ± 0.7	6.1 ± 0.7	4.0 ± 2.5	1.5 ± 0.1	0.6 ± 0.1	1.6 ± 0.1	1.0 ± 0.1
Dotted Line	0.7 ± 0.1	0.7 ± 0.1	0.5 ± 0.1	0.4 ± 0.1	0.9 ± 0.0	1.0 ± 0.1	0.6 ± 0.1
Fog	26.4 ± 18.0	3.5 ± 1.4	0.5 ± 0.0	0.4 ± 0.0	0.9 ± 0.1	1.0 ± 0.1	0.5 ± 0.1
Glass Blur	5.9 ± 1.6	3.1 ± 0.3	1.8 ± 0.1	1.4 ± 0.1	0.4 ± 0.2	2.1 ± 0.1	1.5 ± 0.1
Impulse Noise	1.2 ± 0.2	1.2 ± 0.1	1.2 ± 0.1	0.9 ± 0.1	0.7 ± 0.1	1.5 ± 0.1	1.0 ± 0.1
Motion Blur	9.1 ± 3.4	1.4 ± 0.2	1.0 ± 0.2	0.7 ± 0.1	0.8 ± 0.0	1.4 ± 0.1	0.8 ± 0.1
Rotate	2.0 ± 0.1	2.2 ± 0.1	1.9 ± 0.1	1.2 ± 0.1	0.6 ± 0.1	1.9 ± 0.1	1.5 ± 0.1
Scale	1.0 ± 0.1	1.7 ± 0.1	1.0 ± 0.1	0.7 ± 0.1	0.8 ± 0.1	1.5 ± 0.0	0.8 ± 0.1
Shear	0.7 ± 0.1	0.7 ± 0.0	0.8 ± 0.1	0.7 ± 0.1	0.8 ± 0.1	1.2 ± 0.1	0.9 ± 0.1
Shot Noise	0.7 ± 0.1	0.6 ± 0.1	0.5 ± 0.1	0.5 ± 0.1	0.8 ± 0.1	1.2 ± 0.1	0.7 ± 0.1
Spatter	0.6 ± 0.1	0.7 ± 0.1	0.7 ± 0.1	0.7 ± 0.1	0.9 ± 0.1	1.2 ± 0.1	0.8 ± 0.0
Stripe	4.3 ± 1.0	6.5 ± 1.4	2.8 ± 3.1	0.5 ± 0.1	0.9 ± 0.1	1.2 ± 0.2	0.6 ± 0.0
Translate	25.2 ± 0.3	28.6 ± 0.6	29.0 ± 0.7	24.2 ± 4.1	19.2 ± 3.0	18.8 ± 2.7	33.7 ± 8.9
Zigzag	5.4 ± 0.5	4.9 ± 0.3	1.3 ± 0.2	0.8 ± 0.0	0.8 ± 0.0	1.4 ± 0.1	0.8 ± 0.1
Avg.	7.2 ± 1.4	4.1 ± 0.1	3.1 ± 0.4	2.3 ± 0.2	2.0 ± 0.2	2.5 ± 0.2	3.0 ± 0.6
Avg.\translate	5.9 ± 1.5	2.4 ± 0.2	1.3 ± 0.4	0.8 ± 0.0	0.8 ± 0.0	1.4 ± 0.0	0.8 ± 0.0

Published as a conference paper at ICLR 2022

## K.5 EMNIST-DA FULL RESULTS

Tables 17 and 18 show the accuracy and ECE results for each individual shift of EMNIST-DA. We provide the average performance with and without the ‘background shifts’ (bgs), where the background and digit change colour, as these are often the more severe shifts.

By inspecting Table 17, we see that the sky shift resulted in the lowest AdaBN accuracy, while the shot-noise shift resulted in the highest AdaBN accuracy. Thus, we deem these to be the most and least severe EMNIST-DA shifts, i.e. the “severe” and “mild” shifts. We find AdaBN to be a better indicator of shift severity than source-only as some shifts with poor source-only performance can be well-resolved by simply updating the BN-statistics (no parameter updates), e.g. the fog shift.

Table 17: EMNIST-DA accuracy (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	Marg. Gauss.	Full Gauss.	PL	BNM-IM	SHOT-IM	SHOT	FR	BUFR
Bricks	4.2 ± 0.5	5.9 ± 1.0	9.1 ± 1.2	22.6 ± 2.1	6.8 ± 1.2	14.2 ± 1.4	20.5 ± 4.8	76.0 ± 0.2	32.4 ± 4.8	83.8 ± 0.3
Crystals	19.7 ± 3.1	42.1 ± 1.9	50.0 ± 0.7	60.0 ± 1.0	47.4 ± 1.6	61.2 ± 2.5	71.5 ± 3.8	80.1 ± 0.2	76.8 ± 0.4	82.6 ± 0.3
Dotted Line	76.2 ± 0.7	80.8 ± 0.4	82.3 ± 0.4	82.6 ± 0.5	80.7 ± 0.5	86.5 ± 0.4	87.1 ± 0.1	87.5 ± 0.1	87.6 ± 0.1	88.3 ± 0.0
Fog	4.5 ± 0.9	69.0 ± 2.6	77.1 ± 0.6	85.1 ± 0.6	77.4 ± 3.3	86.3 ± 0.3	86.2 ± 0.1	87.0 ± 0.1	87.0 ± 0.1	88.3 ± 0.1
Gaussian Blur	45.1 ± 3.2	65.2 ± 1.6	77.1 ± 0.8	82.3 ± 0.2	78.8 ± 0.8	83.7 ± 0.3	83.7 ± 0.2	83.9 ± 0.2	83.0 ± 0.4	86.0 ± 0.1
Grass	2.3 ± 0.1	6.1 ± 0.4	5.9 ± 1.1	52.7 ± 3.5	6.7 ± 1.8	14.6 ± 6.1	42.4 ± 40.9	61.8 ± 36.5	79.2 ± 0.3	84.5 ± 0.2
Impulse Noise	36.8 ± 1.6	76.7 ± 0.8	81.4 ± 0.3	82.6 ± 0.1	79.9 ± 0.6	84.2 ± 0.3	84.4 ± 0.2	84.4 ± 0.2	84.8 ± 0.2	86.0 ± 0.1
Inverse	5.6 ± 0.5	8.1 ± 2.1	14.1 ± 6.2	64.4 ± 4.3	11.3 ± 2.0	60.4 ± 23.4	83.2 ± 0.4	85.1 ± 0.2	83.1 ± 0.7	88.3 ± 0.1
Oranges	26.5 ± 2.7	40.7 ± 2.3	49.3 ± 1.0	77.1 ± 0.6	43.0 ± 3.1	79.9 ± 0.6	80.5 ± 0.3	82.4 ± 0.2	82.3 ± 0.3	84.8 ± 0.3
Shot Noise	78.5 ± 0.7	84.9 ± 0.2	85.8 ± 0.3	85.7 ± 0.2	85.0 ± 0.3	86.5 ± 0.1	86.3 ± 0.1	86.1 ± 0.1	86.4 ± 0.1	87.0 ± 0.2
Sky	3.8 ± 0.4	3.7 ± 0.7	4.8 ± 0.4	29.8 ± 9.8	3.3 ± 0.6	8.3 ± 1.3	24.0 ± 7.5	55.1 ± 23.5	15.3 ± 6.8	84.6 ± 0.2
Stripe	15.4 ± 1.1	46.9 ± 4.9	63.0 ± 5.6	82.3 ± 0.8	63.8 ± 3.7	82.8 ± 0.5	83.9 ± 0.3	85.1 ± 0.2	84.5 ± 0.4	87.1 ± 0.1
Zigzag	65.0 ± 0.2	71.3 ± 0.2	73.8 ± 0.1	76.1 ± 0.3	72.3 ± 0.2	79.7 ± 0.5	81.0 ± 0.5	85.8 ± 0.2	85.7 ± 0.2	87.5 ± 0.2
Avg.	29.5 ± 0.5	46.2 ± 1.1	51.8 ± 1.1	67.9 ± 0.7	50.5 ± 0.6	63.7 ± 2.2	70.3 ± 3.7	80.0 ± 4.4	74.4 ± 0.8	86.1 ± 0.1
Avg.bgs	40.9 ± 0.4	62.8 ± 1.1	69.3 ± 1.4	80.1 ± 0.5	68.6 ± 0.5	81.2 ± 3.1	84.5 ± 0.1	85.6 ± 0.1	85.2 ± 0.1	87.3 ± 0.0

Table 18: EMNIST-DA ECE (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	Marg. Gauss.	Full Gauss.	PL	BNM-IM	SHOT-IM	SHOT	FR	BUFR
Bricks	54.6 ± 5.0	64.4 ± 1.1	62.0 ± 0.9	52.4 ± 2.0	93.2 ± 1.2	84.9 ± 1.4	79.4 ± 4.8	22.5 ± 0.3	44.1 ± 4.2	6.2 ± 0.4
Crystals	27.0 ± 3.0	29.0 ± 1.2	24.0 ± 0.3	22.0 ± 0.6	52.7 ± 1.7	38.1 ± 2.5	28.5 ± 3.8	19.3 ± 0.1	10.5 ± 0.3	6.9 ± 0.3
Dotted Line	11.4 ± 0.6	9.2 ± 0.5	7.9 ± 0.4	7.5 ± 0.4	19.6 ± 0.7	13.0 ± 0.4	12.9 ± 0.1	12.9 ± 0.1	3.8 ± 0.2	3.4 ± 0.2
Fog	19.2 ± 6.5	13.8 ± 1.5	8.8 ± 0.8	4.8 ± 0.4	24.1 ± 4.0	13.3 ± 0.4	13.8 ± 0.1	13.5 ± 0.1	3.9 ± 0.2	3.3 ± 0.1
Gaussian Blur	15.0 ± 3.9	15.3 ± 0.9	8.7 ± 0.5	6.8 ± 0.3	21.2 ± 0.8	15.8 ± 0.4	16.4 ± 0.2	16.0 ± 0.3	6.4 ± 0.7	4.9 ± 0.1
Grass	21.6 ± 5.4	61.3 ± 0.8	61.0 ± 1.0	27.2 ± 2.7	93.6 ± 1.5	84.6 ± 6.2	57.5 ± 40.9	37.5 ± 36.1	8.7 ± 0.6	5.7 ± 0.2
Impulse Noise	32.0 ± 1.8	9.9 ± 0.6	7.1 ± 0.3	6.6 ± 0.1	20.1 ± 0.6	15.3 ± 0.3	15.6 ± 0.2	15.8 ± 0.2	5.3 ± 0.1	4.7 ± 0.1
Inverse	65.1 ± 5.8	60.8 ± 2.2	54.9 ± 5.7	18.1 ± 3.0	89.3 ± 2.1	39.0 ± 23.3	16.9 ± 0.5	14.7 ± 0.1	5.6 ± 0.5	3.3 ± 0.1
Oranges	23.8 ± 2.7	25.3 ± 2.0	22.5 ± 2.3	10.1 ± 0.6	57.6 ± 2.7	19.6 ± 0.6	19.6 ± 0.4	17.4 ± 0.2	6.9 ± 0.5	5.5 ± 0.3
Shot Noise	4.8 ± 0.5	4.9 ± 0.3	4.5 ± 0.3	4.9 ± 0.2	16.4 ± 0.1	13.0 ± 0.1	13.7 ± 0.1	14.8 ± 0.1	4.6 ± 0.3	4.2 ± 0.2
Sky	42.6 ± 3.5	52.4 ± 4.9	51.6 ± 6.4	45.8 ± 8.4	97.2 ± 0.4	90.2 ± 1.1	76.0 ± 7.5	42.7 ± 23.0	58.0 ± 6.8	5.6 ± 0.3
Stripe	63.8 ± 3.0	31.6 ± 4.4	20.2 ± 4.8	6.8 ± 0.4	36.2 ± 3.8	16.8 ± 0.5	16.1 ± 0.3	15.0 ± 0.3	5.4 ± 0.2	4.1 ± 0.1
Zigzag	19.9 ± 0.3	16.7 ± 0.2	14.6 ± 0.1	12.7 ± 0.3	27.6 ± 0.2	19.7 ± 0.5	19.0 ± 0.5	14.4 ± 0.1	4.9 ± 0.1	3.8 ± 0.2
Avg.	30.8 ± 1.6	30.3 ± 1.1	26.7 ± 1.1	17.4 ± 0.7	49.9 ± 0.6	35.6 ± 2.2	29.6 ± 3.7	19.7 ± 4.4	12.9 ± 0.9	4.7 ± 0.2
Avg.bgs	28.9 ± 1.3	20.3 ± 0.8	15.8 ± 1.1	8.5 ± 0.4	31.8 ± 0.5	18.2 ± 3.1	15.6 ± 0.1	14.6 ± 0.1	5.0 ± 0.2	4.0 ± 0.1

Published as a conference paper at ICLR 2022

## K.6 CIFAR-10-C FULL RESULTS

Tables 19 and 20 show the accuracy and ECE results for each individual corruption of CIFAR-10-C. It is worth noting that BUFR achieves the biggest wins on the more severe shifts, i.e. those on which AdaBN (Li et al., 2017) performs poorly.

Table 19: CIFAR-10-C accuracy (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	PL	SHOT-IM	TENT	FR	BUFR
Brightness	91.4 ± 0.4	91.5 ± 0.3	91.9 ± 0.2	92.7 ± 0.3	93.2 ± 0.3	93 ± 0.4	93.3 ± 0.3
Contrast	32.3 ± 1.3	87.1 ± 0.3	86.6 ± 3.2	90.8 ± 0.9	91.3 ± 1.7	90.9 ± 0.9	92.9 ± 0.7
Defocus blr	53.1 ± 6.4	88.8 ± 0.4	89.3 ± 0.5	90.5 ± 0.4	90.9 ± 0.5	90.9 ± 0.3	91.5 ± 0.5
Elastic	77.6 ± 0.6	78.2 ± 0.4	79.2 ± 0.8	81.4 ± 0.5	82.7 ± 0.5	82.7 ± 0.4	84.2 ± 0.3
Fog	72.9 ± 2.6	85.9 ± 0.9	86.5 ± 0.8	88.7 ± 0.4	89.5 ± 0.4	89.5 ± 0.5	91.5 ± 0.5
Frost	64.4 ± 2.4	80.7 ± 0.7	82.4 ± 1.2	85.4 ± 0.6	86.8 ± 0.7	87 ± 0.6	89.1 ± 0.9
Gauss. blr	35.9 ± 8	88.2 ± 0.6	89 ± 0.7	90.5 ± 0.5	91 ± 0.6	91.2 ± 0.6	92.3 ± 0.4
Gauss. nse	27.7 ± 5.1	69.2 ± 1	74.6 ± 0.6	79.2 ± 0.9	81.3 ± 0.5	81.9 ± 0.1	85.9 ± 0.4
Glass blr	51.3 ± 1.8	66.7 ± 0.4	69 ± 0.3	73.7 ± 1	74.7 ± 0.8	76.8 ± 0.8	80.3 ± 0.5
Impulse nse	25.9 ± 3.8	62.1 ± 1	67.2 ± 0.5	73.2 ± 0.8	75.3 ± 0.8	76.6 ± 0.4	89.3 ± 1.4
Jpeg compr.	74.9 ± 1	74.1 ± 1	77.3 ± 0.6	81 ± 0.3	82.9 ± 0.5	83.4 ± 0.5	85.8 ± 0.6
Motion blr	66.1 ± 1.7	87.2 ± 0.2	87.8 ± 0.2	89.1 ± 0.2	90 ± 0.3	89.8 ± 0.2	90.8 ± 0.2
Pixelate	48.2 ± 2.2	80.4 ± 0.5	82 ± 0.4	85.5 ± 0.7	87.6 ± 0.9	87.5 ± 0.8	89.9 ± 0.6
Saturate	89.9 ± 0.4	92 ± 0.1	92.5 ± 0.3	93.1 ± 0.1	93.3 ± 0.1	93.4 ± 0.4	93.5 ± 0.3
Shot nse	34.4 ± 4.9	71.2 ± 1.2	77.1 ± 0.9	81.6 ± 0.7	83.5 ± 0.6	85.6 ± 1.9	87 ± 0.2
Snow	76.6 ± 1.1	82.4 ± 0.6	83.8 ± 1.1	86.4 ± 0.6	87.8 ± 0.7	88.4 ± 1.7	89.7 ± 0.5
Spatter	75 ± 0.8	83.3 ± 0.5	85.5 ± 0.3	88 ± 0.2	88.5 ± 0.3	91 ± 2.4	92.6 ± 0.5
Speckle nse	40.7 ± 3.7	70.4 ± 0.8	76.1 ± 1.2	81.3 ± 1	83.2 ± 0.9	85.8 ± 1.7	87.4 ± 0.4
Zoom blr	60.5 ± 5.1	88.1 ± 0.3	89 ± 0.4	90.6 ± 0.2	91.3 ± 0.3	91.2 ± 0.7	91.6 ± 0.2
Avg.	57.8 ± 0.7	80.4 ± 0.1	82.5 ± 0.3	85.4 ± 0.2	86.6 ± 0.3	87.2 ± 0.7	89.4 ± 0.2

Table 20: CIFAR-10-C ECE (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	PL	SHOT-IM	TENT	FR	BUFR
Brightness	4.7 ± 0.2	4 ± 0.1	8.1 ± 0.2	7.2 ± 0.4	6.4 ± 0.3	5.9 ± 0.3	6.2 ± 0.2
Contrast	43.5 ± 2.8	5.7 ± 0.4	13.2 ± 3.1	9.8 ± 0.9	8.4 ± 1.6	6.6 ± 0.4	6.6 ± 0.7
Defocus blr	28.2 ± 4	6.1 ± 0.4	10.7 ± 0.5	9.4 ± 0.4	8.6 ± 0.5	7.8 ± 0.3	7.9 ± 0.4
Elastic	12.4 ± 0.7	12.6 ± 0.4	20.8 ± 0.8	18.6 ± 0.5	16.5 ± 0.5	15.1 ± 0.5	15.2 ± 0.3
Fog	17.4 ± 2.1	7.5 ± 0.7	13.5 ± 0.8	11.3 ± 0.3	10.1 ± 0.4	8.9 ± 0.3	8 ± 0.5
Frost	22.7 ± 1.7	10.4 ± 0.6	17.5 ± 1.2	14.6 ± 0.6	12.7 ± 0.6	10.7 ± 0.8	10.3 ± 0.9
Gauss. blr	40.7 ± 6.2	6.1 ± 0.4	11 ± 0.7	9.5 ± 0.4	8.5 ± 0.5	7.5 ± 0.4	7.3 ± 0.4
Gauss. nse	57.6 ± 6.7	18.5 ± 0.7	25.3 ± 0.6	20.9 ± 0.9	18 ± 0.4	15.9 ± 0.3	13.3 ± 0.4
Glass blr	31.2 ± 1.3	20.8 ± 0.4	30.9 ± 0.3	26.3 ± 1	24.2 ± 0.8	20.9 ± 0.7	18.9 ± 0.5
Impulse nse	51.2 ± 4	23.3 ± 0.8	32.7 ± 0.5	26.8 ± 0.9	23.7 ± 0.8	20.6 ± 0.5	10.2 ± 1.3
Jpeg compr.	14.6 ± 0.8	15.5 ± 0.7	22.6 ± 0.6	18.9 ± 0.4	16.4 ± 0.5	14.5 ± 0.4	13.6 ± 0.7
Motion blr	21.1 ± 1.3	6.8 ± 0.3	12.1 ± 0.2	10.9 ± 0.2	9.5 ± 0.3	8.7 ± 0.3	8.6 ± 0.2
Pixelate	36.9 ± 2.5	11.1 ± 0.4	17.9 ± 0.4	14.5 ± 0.7	11.9 ± 0.9	10.7 ± 0.7	9.5 ± 0.6
Saturate	5.5 ± 0.3	4.2 ± 0.1	7.4 ± 0.3	6.9 ± 0.1	6.4 ± 0.1	5.9 ± 0.2	6 ± 0.3
Shot nse	50.2 ± 5.9	17 ± 0.9	22.8 ± 0.9	18.4 ± 0.7	15.9 ± 0.6	14 ± 0.3	12.3 ± 0.2
Snow	14.3 ± 0.5	9.8 ± 0.4	16.1 ± 1.1	13.6 ± 0.6	11.6 ± 0.7	10.5 ± 0.7	9.7 ± 0.4
Spatter	16.9 ± 0.8	9.3 ± 0.3	14.5 ± 0.3	12 ± 0.2	11 ± 0.2	9.3 ± 0.2	7 ± 0.6
Speckle nse	43.2 ± 4.5	17.9 ± 0.5	23.8 ± 1.1	18.7 ± 1	16.1 ± 0.9	13.9 ± 0.7	11.9 ± 0.4
Zoom blr	24.4 ± 3.5	6.2 ± 0.1	11 ± 0.4	9.4 ± 0.2	8.3 ± 0.3	7.6 ± 0.5	7.9 ± 0.2
Avg.	28.2 ± 0.4	11.2 ± 0.1	17.5 ± 0.3	14.6 ± 0.2	12.8 ± 0.3	11.3 ± 0.3	10 ± 0.2

Published as a conference paper at ICLR 2022

## K.7 CIFAR-100-C FULL RESULTS

Tables 21 and 22 show the accuracy and ECE results for each individual corruption of CIFAR-100-C. It is worth noting that BUFR achieves the biggest wins on the more severe shifts, i.e. those on which AdaBN (Li et al., 2017) performs poorly.

Table 21: CIFAR-100-C accuracy (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	PL	SHOT-IM	TENT	FR	BUFR
Brightness	63.2 ± 1.1	66.1 ± 0.5	69.6 ± 0.7	72.6 ± 0.6	72.2 ± 0.5	71.8 ± 0.5	73.6 ± 0.2
Contrast	13.9 ± 0.6	61.4 ± 0.4	59.2 ± 3.5	70.1 ± 0.4	64 ± 3.1	68 ± 0.5	72.2 ± 0.5
Defocus blr	35.9 ± 0.7	65.6 ± 0.1	69.3 ± 0.1	71.8 ± 0.3	71 ± 0.5	71.2 ± 0.1	72.2 ± 0.2
Elastic	58.5 ± 0.7	60.4 ± 0.2	63.9 ± 0.4	66.9 ± 0.2	65.5 ± 0.2	65.9 ± 0.4	67.1 ± 0.5
Fog	36.9 ± 0.5	55.4 ± 0.6	60.4 ± 0.6	66.5 ± 0.6	67.1 ± 0.6	64.9 ± 0.4	70.1 ± 0.5
Frost	41.1 ± 0.9	55.3 ± 0.6	60.1 ± 0.8	65.2 ± 0.4	65.3 ± 0.9	63 ± 0.5	67.5 ± 0.7
Gauss. blr	28.2 ± 1	64.3 ± 0.3	68.9 ± 0.1	71.7 ± 0.2	71 ± 0.3	70.9 ± 0.3	72.9 ± 0.6
Gauss. nse	11.9 ± 1.2	43.8 ± 0.6	53.1 ± 0.7	60.3 ± 0.4	59.5 ± 0.6	57.7 ± 0.4	63 ± 0.3
Glass blr	45.1 ± 0.9	53.3 ± 0.6	57.3 ± 0.7	62.4 ± 0.3	61.4 ± 0.5	60.5 ± 0.3	63.2 ± 0.4
Impulse nse	7.2 ± 0.8	40.8 ± 0.4	50.6 ± 0.5	58.4 ± 0.6	56.3 ± 0.7	55.2 ± 0.9	66.9 ± 0.6
Jpeg compr.	48.6 ± 0.9	49.8 ± 0.7	55.8 ± 0.3	61.2 ± 0.5	60.8 ± 0.1	59.3 ± 0.5	62.6 ± 0.4
Motion blr	45.1 ± 0.5	63.4 ± 0.2	66.3 ± 0.6	69.7 ± 0.2	69 ± 0.5	68.6 ± 0.4	70.8 ± 0.2
Pixelate	22.3 ± 0.4	59.4 ± 0.6	64.9 ± 0.6	69.7 ± 0.4	69.8 ± 0.4	68.1 ± 0.3	71.4 ± 0.5
Saturate	55.8 ± 0.4	65.7 ± 0.4	70.2 ± 0.8	72.6 ± 0.2	71.4 ± 0.7	72.2 ± 0.5	72.4 ± 0.6
Shot nse	14.1 ± 1.2	44.6 ± 0.9	56.1 ± 0.8	61.9 ± 0.6	60.3 ± 0.4	59.8 ± 0.3	62.1 ± 2.8
Snow	49.4 ± 0.8	53.5 ± 0.4	59.8 ± 0.9	65 ± 0.6	65.6 ± 0.4	63.8 ± 0.6	65.9 ± 2.2
Spatter	54.8 ± 1.1	64.9 ± 0.6	72.1 ± 0.3	73.8 ± 0.4	72.9 ± 0.5	73.8 ± 0.5	74.3 ± 0.2
Speckle nse	15.6 ± 1.3	42.3 ± 1	54.2 ± 1.5	62.1 ± 0.6	59.8 ± 0.3	59.6 ± 0.8	62.1 ± 2.7
Zoom blr	45.1 ± 0.7	65.9 ± 0.3	69.1 ± 0.5	71.9 ± 0.3	71.1 ± 0.8	71 ± 0.6	71.2 ± 0.4
Avg.	36.4 ± 0.5	56.6 ± 0.3	62.1 ± 0.2	67 ± 0.2	66 ± 0.4	65.5 ± 0.2	68.5 ± 0.2

Table 22: CIFAR-100-C ECE (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	PL	SHOT-IM	TENT	FR	BUFR
Brightness	6.3 ± 0.3	9.4 ± 0.3	30.2 ± 0.7	27.4 ± 0.4	20.7 ± 0.4	12.4 ± 0.3	12 ± 0.6
Contrast	37.8 ± 2.2	11.4 ± 0.3	40.5 ± 3.4	29.6 ± 0.8	29.5 ± 3.5	14 ± 0.2	12.8 ± 0.5
Defocus blr	16 ± 0.8	9.7 ± 0.3	30.6 ± 0.2	28.2 ± 0.4	21.6 ± 0.3	13.4 ± 0.3	12.7 ± 0.2
Elastic	8 ± 0.1	10.8 ± 0.2	35.9 ± 0.4	33 ± 0.3	25.8 ± 0.1	15.2 ± 0.2	15.3 ± 0.3
Fog	21 ± 0.6	12.2 ± 0.3	39.5 ± 0.6	33.3 ± 0.7	24.8 ± 0.5	15.9 ± 0.3	14 ± 0.6
Frost	14.1 ± 1.1	13.3 ± 0.4	39.7 ± 0.8	34.8 ± 0.4	26.1 ± 0.7	16.3 ± 0.3	15.3 ± 0.2
Gauss. blr	20.5 ± 1.4	10 ± 0.4	31 ± 0.1	28.4 ± 0.2	21.7 ± 0.2	13.5 ± 0.2	12.5 ± 0.3
Gauss. nse	39.3 ± 5.5	16.7 ± 0.2	46.8 ± 0.6	39.8 ± 0.5	30.8 ± 0.7	19.4 ± 0.5	17.5 ± 0.5
Glass blr	15.7 ± 1.1	13.4 ± 0.1	42.5 ± 0.7	37.6 ± 0.3	29.1 ± 0.4	17.9 ± 0.4	17.6 ± 0.6
Impulse nse	35.1 ± 2.6	17.4 ± 0.2	49.3 ± 0.6	41.5 ± 0.7	33.7 ± 0.8	20.5 ± 0.3	15.2 ± 0.2
Jpeg compr.	8.6 ± 0.2	15 ± 0.4	44.1 ± 0.4	38.8 ± 0.5	29.6 ± 0.2	19.1 ± 0.2	18.2 ± 0.5
Motion blr	12.2 ± 0.2	10.4 ± 0.3	33.6 ± 0.6	30.3 ± 0.2	23.2 ± 0.4	14.3 ± 0.3	13.6 ± 0.3
Pixelate	27.5 ± 1	11.6 ± 0.4	35 ± 0.6	30.3 ± 0.4	22.5 ± 0.3	14.2 ± 0.4	13.6 ± 0.4
Saturate	8.8 ± 0.2	9.5 ± 0.3	29.6 ± 0.8	27.4 ± 0.3	21.2 ± 0.6	12.7 ± 0.2	12.3 ± 0.7
Shot nse	37.2 ± 5.9	16 ± 0.2	43.7 ± 0.8	38.1 ± 0.5	30.2 ± 0.8	18.6 ± 0.4	17 ± 0.6
Snow	8.5 ± 0.3	14.4 ± 0.2	40.1 ± 0.9	34.9 ± 0.7	25.7 ± 0.5	17 ± 0.5	14.9 ± 0.1
Spatter	6.7 ± 0.3	9.3 ± 0.1	27.8 ± 0.3	26.2 ± 0.4	20 ± 0.6	12 ± 0.3	11 ± 0.4
Speckle nse	34.5 ± 5.4	17.2 ± 0.3	45.7 ± 1.6	37.9 ± 0.6	30.6 ± 0.2	18.7 ± 0.6	16.8 ± 0.8
Zoom blr	10.5 ± 0.3	9.1 ± 0.2	30.8 ± 0.5	28.2 ± 0.4	21.5 ± 0.7	13.1 ± 0.5	13.2 ± 0.7
Avg.	19.4 ± 0.9	12.5 ± 0.1	37.7 ± 0.2	32.9 ± 0.2	25.7 ± 0.4	15.7 ± 0.1	14.5 ± 0.3

Published as a conference paper at ICLR 2022

## K.8 CIFAR-10-C FULL ONLINE RESULTS

Tables 23 and 24 show the accuracy and ECE results for each individual corruption of CIFAR-10-C when adapting in an *online* fashion (see Appendix K.2). It is worth noting that FR achieves the biggest wins on the more severe shifts, i.e. those on which AdaBN (Li et al., 2017) performs poorly.

Table 23: CIFAR-10-C *online* accuracy (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	SHOT-IM	TENT	FR
Brightness	91.4 ± 0.4	91.6 ± 0.2	92.2 ± 0.4	91.8 ± 0.3	92.8 ± 0.3
Contrast	32.3 ± 1.3	87.1 ± 0.4	87.8 ± 0.5	87.8 ± 0.6	89.8 ± 0.6
Defocus blr	53.1 ± 6.4	88.7 ± 0.5	89.7 ± 0.5	89.1 ± 0.5	90.6 ± 0.5
Elastic	77.6 ± 0.6	78 ± 0.3	80.3 ± 0.6	79.2 ± 0.5	82 ± 0.4
Fog	72.9 ± 2.6	85.9 ± 1.1	87.2 ± 0.5	86.5 ± 0.8	89 ± 0.8
Frost	64.4 ± 2.4	80.7 ± 0.8	83 ± 0.8	81.8 ± 0.8	85.9 ± 0.7
Gauss. blr	35.9 ± 8	88.3 ± 0.7	89.5 ± 0.6	88.8 ± 0.5	90.8 ± 0.6
Gauss. nse	27.7 ± 5.1	68.8 ± 0.9	75.4 ± 0.8	72.3 ± 0.7	80.6 ± 0.6
Glass blr	51.3 ± 1.8	66.7 ± 0.5	70.6 ± 1	68.3 ± 0.6	74.7 ± 0.9
Impulse nse	25.9 ± 3.8	62 ± 1.2	68.8 ± 0.8	65.5 ± 0.7	74.5 ± 0.4
Jpeg compr.	74.9 ± 1	73.9 ± 1.2	78.4 ± 0.9	76.2 ± 0.9	82.2 ± 0.5
Motion blr	66.1 ± 1.7	87 ± 0.1	88.2 ± 0.3	87.6 ± 0.3	89.5 ± 0.2
Pixelate	48.2 ± 2.2	80.5 ± 0.4	83.2 ± 0.7	81.7 ± 0.5	86.7 ± 0.7
Saturate	89.9 ± 0.4	91.9 ± 0.1	92.4 ± 0.1	92.3 ± 0.2	92.8 ± 0.2
Shot nse	34.4 ± 4.9	70.9 ± 1.2	77.7 ± 1.6	74.6 ± 1.3	82.2 ± 0.6
Snow	76.6 ± 1.1	82.6 ± 0.7	84.5 ± 0.9	83.4 ± 0.9	86.8 ± 0.6
Spatter	75 ± 0.8	83.2 ± 0.5	86 ± 0.2	84.6 ± 0.2	88.6 ± 0.2
Speckle nse	40.7 ± 3.7	70.2 ± 0.7	77.2 ± 0.6	74.2 ± 0.6	82.4 ± 0.2
Zoom blr	60.5 ± 5.1	88 ± 0.4	89.4 ± 0.2	88.6 ± 0.3	90.7 ± 0.2
Avg.	57.8 ± 0.7	80.3 ± 0	83.2 ± 0.2	81.8 ± 0.2	85.9 ± 0.3

Table 24: CIFAR-10-C *online* ECE (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	SHOT-IM	TENT	FR
Brightness	4.7 ± 0.2	5.4 ± 0.2	5.1 ± 0.3	5.1 ± 0.2	4.9 ± 0.3
Contrast	43.5 ± 2.8	6.8 ± 0.4	8.7 ± 0.5	7.6 ± 0.5	6.1 ± 0.4
Defocus blr	28.2 ± 4	7.1 ± 0.4	6.7 ± 0.3	7 ± 0.3	6.4 ± 0.3
Elastic	12.4 ± 0.7	13.5 ± 0.3	12.7 ± 0.4	12.9 ± 0.5	12.3 ± 0.4
Fog	17.4 ± 2.1	8.4 ± 0.6	8.3 ± 0.3	8.3 ± 0.4	7.3 ± 0.5
Frost	22.7 ± 1.7	11.2 ± 0.7	10.9 ± 0.5	11 ± 0.5	9 ± 0.6
Gauss. blr	40.7 ± 6.2	7.3 ± 0.4	6.8 ± 0.4	7 ± 0.3	6.3 ± 0.4
Gauss. nse	57.6 ± 6.7	19.2 ± 0.7	16 ± 0.6	17.6 ± 0.4	13.2 ± 0.6
Glass blr	31.2 ± 1.3	21.4 ± 0.6	19.6 ± 0.7	20.7 ± 0.5	17.9 ± 0.7
Impulse nse	51.2 ± 4	23.8 ± 0.9	20.6 ± 0.8	22.2 ± 0.4	17.9 ± 0.4
Jpeg compr.	14.6 ± 0.8	16.3 ± 0.9	14 ± 0.5	15.1 ± 0.6	12.2 ± 0.4
Motion blr	21.1 ± 1.3	7.8 ± 0.1	7.6 ± 0.3	7.7 ± 0.2	7 ± 0.2
Pixelate	36.9 ± 2.5	12 ± 0.4	10.8 ± 0.6	11.5 ± 0.5	8.9 ± 0.5
Saturate	5.5 ± 0.3	5.1 ± 0.1	5 ± 0.1	5.1 ± 0.1	4.9 ± 0.2
Shot nse	50.2 ± 5.9	17.9 ± 0.9	14.3 ± 1.1	16 ± 0.8	12 ± 0.5
Snow	14.3 ± 0.5	10.7 ± 0.3	9.9 ± 0.6	10.4 ± 0.6	8.9 ± 0.5
Spatter	16.9 ± 0.8	10.2 ± 0.4	9.1 ± 0.2	9.6 ± 0.2	7.6 ± 0.2
Speckle nse	43.2 ± 4.5	18.8 ± 0.5	14.8 ± 0.5	16.3 ± 0.5	11.9 ± 0.2
Zoom blr	24.4 ± 3.5	7.3 ± 0.3	6.8 ± 0.2	7.1 ± 0.2	6.3 ± 0.1
Avg.	28.2 ± 0.4	12.1 ± 0	10.9 ± 0.1	11.5 ± 0.1	9.5 ± 0.2

Published as a conference paper at ICLR 2022

## K.9 CIFAR-100-C FULL ONLINE RESULTS

Tables 25 and 26 show the accuracy and ECE results for each individual corruption of CIFAR-100-C when adapting in an *online* fashion (see Appendix K.2). It is worth noting that FR achieves the biggest wins on the more severe shifts, i.e. those on which AdaBN (Li et al., 2017) performs poorly.

Table 25: CIFAR-100-C *online* accuracy (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	SHOT-IM	TENT	FR
Brightness	63.2 ± 1.1	66.1 ± 0.4	69.3 ± 0.9	69.9 ± 0.7	69.4 ± 0.4
Contrast	13.9 ± 0.6	61.4 ± 0.5	64.8 ± 0.5	66.6 ± 1.1	64.5 ± 0.3
Defocus blr	35.9 ± 0.7	65.6 ± 0.1	69 ± 0.1	69.4 ± 0.3	68.6 ± 0.2
Elastic	58.5 ± 0.7	60.4 ± 0.2	63.3 ± 0.5	63.7 ± 0.1	63.4 ± 0.3
Fog	36.9 ± 0.5	55.4 ± 0.6	61 ± 0.5	62.5 ± 0.7	61.7 ± 0.5
Frost	41.1 ± 0.9	55.3 ± 0.6	60.5 ± 1	61.8 ± 0.6	60.8 ± 0.8
Gauss. blr	28.2 ± 1	64.3 ± 0.3	68.6 ± 0.2	69 ± 0.6	68.4 ± 0.5
Gauss. nse	11.9 ± 1.2	43.8 ± 0.6	53.5 ± 0.2	55.1 ± 0.5	54.7 ± 0.3
Glass blr	45.1 ± 0.9	53.3 ± 0.6	57.8 ± 0.4	58.2 ± 0.5	57.9 ± 0.5
Impulse nse	7.2 ± 0.8	40.8 ± 0.5	50.2 ± 0.4	50.9 ± 0.7	51.7 ± 0.8
Jpeg compr.	48.6 ± 0.9	49.8 ± 0.7	56 ± 0.2	57.2 ± 0.2	56.6 ± 0.6
Motion blr	45.1 ± 0.5	63.4 ± 0.2	66.4 ± 0.4	66.7 ± 0.6	66 ± 0.3
Pixelate	22.3 ± 0.4	59.4 ± 0.6	65.1 ± 0.7	67.1 ± 0.4	65.6 ± 0.6
Saturate	55.8 ± 0.4	65.7 ± 0.4	69.5 ± 0.6	69.5 ± 0.6	69.3 ± 0.4
Shot nse	14.1 ± 1.2	44.6 ± 0.9	54.9 ± 0.1	55.5 ± 0.4	56.4 ± 0.3
Snow	49.4 ± 0.8	53.5 ± 0.4	59.7 ± 1.1	61.6 ± 0.8	60.5 ± 0.5
Spatter	54.8 ± 1.1	64.9 ± 0.6	71.3 ± 0.5	70.6 ± 0.6	71.6 ± 0.7
Speckle nse	15.6 ± 1.3	42.3 ± 1	54.2 ± 0.3	54.9 ± 0.3	55.8 ± 0.6
Zoom blr	45.1 ± 0.7	65.9 ± 0.3	68.9 ± 0.6	69 ± 0.4	68.8 ± 0.3
Avg.	36.4 ± 0.5	56.6 ± 0.3	62.3 ± 0.3	63.1 ± 0.3	62.7 ± 0.3

Table 26: CIFAR-100-C *online* ECE (%) results. Shown are the mean and 1 standard deviation.

	Src-only	AdaBN	SHOT-IM	TENT	FR
Brightness	6.3 ± 0.3	11.4 ± 0.1	11.6 ± 0.4	11.9 ± 0.2	10.9 ± 0.2
Contrast	37.8 ± 2.2	12.5 ± 0.2	14.6 ± 0.3	14.1 ± 0.6	12.5 ± 0.2
Defocus blr	16 ± 0.8	11.4 ± 0.2	11.3 ± 0.2	11.8 ± 0.2	11.4 ± 0.3
Elastic	8 ± 0.1	12.7 ± 0.2	12.9 ± 0.2	13.4 ± 0.3	13 ± 0.2
Fog	21 ± 0.6	13.8 ± 0.2	13.9 ± 0.2	14.4 ± 0.2	13.6 ± 0.3
Frost	14.1 ± 1.1	14.5 ± 0.2	14.5 ± 0.6	14.8 ± 0.3	13.9 ± 0.4
Gauss. blr	20.5 ± 1.4	11.9 ± 0.3	11.6 ± 0.4	11.9 ± 0.2	11.9 ± 0.4
Gauss. nse	39.3 ± 5.5	17.7 ± 0.3	16.7 ± 0.3	17.2 ± 0.7	16.5 ± 0.3
Glass blr	15.7 ± 1.1	15 ± 0.1	15.2 ± 0.5	16.1 ± 0.3	15.1 ± 0.1
Impulse nse	35.1 ± 2.6	18.4 ± 0.2	18.1 ± 0.2	19.4 ± 0.5	17.9 ± 0.3
Jpeg compr.	8.6 ± 0.2	16.2 ± 0.3	15.9 ± 0.1	16.4 ± 0.4	16.2 ± 0.2
Motion blr	12.2 ± 0.2	12.2 ± 0.2	12.2 ± 0.3	12.9 ± 0.2	12.5 ± 0.2
Pixelate	27.5 ± 1	13 ± 0.3	12.5 ± 0.3	12.4 ± 0.1	12.3 ± 0.2
Saturate	8.8 ± 0.2	11.4 ± 0.1	11.3 ± 0.3	11.7 ± 0.4	11.3 ± 0.4
Shot nse	37.2 ± 5.9	17.2 ± 0.3	16.4 ± 0.2	17.5 ± 0.7	16.2 ± 0.3
Snow	8.5 ± 0.3	15.6 ± 0.2	15 ± 0.4	14.7 ± 0.3	14.8 ± 0.1
Spatter	6.7 ± 0.3	11.4 ± 0.2	10.7 ± 0.2	11.3 ± 0.3	10.7 ± 0.3
Speckle nse	34.5 ± 5.4	18.2 ± 0.4	16.5 ± 0.4	17.5 ± 0.3	16.1 ± 0.4
Zoom blr	10.5 ± 0.3	11.2 ± 0.2	11.2 ± 0.3	11.6 ± 0.3	11.4 ± 0.1
Avg.	19.4 ± 0.9	14 ± 0.1	13.8 ± 0.1	14.3 ± 0.1	13.6 ± 0.1

Published as a conference paper at ICLR 2022

## L NOTATIONS

Table 27 summarizes the notations used in the paper.

Table 27: Notations.

	Symbol	Description
Distributions	$p_{\mathbf{z}}$	Source feature distribution
	$q_{\mathbf{z}}$	Target feature distribution
	$p_{z_d}$	Source $d$ -th marginal feature distribution
	$q_{z_d}$	Target $d$ -th marginal feature distribution
	$\pi_{\mathbf{z}}^s$	Source approx. marginal feature distributions
	$\pi_{\mathbf{z}}^t$	Target approx. marginal feature distributions
	$\pi_{z_d}^s$	Source $d$ -th approx. marginal feature distribution
	$\pi_{z_d}^t$	Target $d$ -th approx. marginal feature distribution
	$\pi_{\mathbf{a}}^s$	Source approx. marginal logit distributions
	$\pi_{\mathbf{a}}^t$	Target approx. marginal logit distributions
	$\pi_{a_k}^s$	Source $k$ -th approx. marginal logit distribution
	$\pi_{a_k}^t$	Target $k$ -th approx. marginal logit distribution
	Sets	$\mathcal{D}_s$
$\mathcal{D}_t$		Unlabelled target dataset
$\mathcal{X}_s$		Input-set of the source domain
$\mathcal{X}_t$		Input-set of the target domain
$\mathcal{Y}_s$		Label-set of the target domain
$\mathcal{Y}_t$		Label-set of the target domain
Network	$f_s$	Source model, $f_s = h(g_s(\cdot))$
	$f_t$	Target model, $f_t = h(g_t(\cdot))$
	$g_s$	Source feature-extractor
	$g_t$	Target feature-extractor
	$h$	Classifier (or regressor)
Other	$\mathbf{u}$	Soft-binning function
	$z_d^{min}$	Minimum value of feature $d$ (on the source data)
	$z_d^{max}$	Maximum value of feature $d$ (on the source data)
	$\tau$	Temperature parameter for soft binning

## **A.2 Domain Generalisation: A Probabilistic Framework** **(§ 4.2)**



# Appendices

## Table of Contents

---

<b>A Causality</b>	<b>20</b>
A.1 Definitions and example . . . . .	20
A.2 EQRM recovers the causal predictor . . . . .	20
<b>B On the equivalence of different DG formulations</b>	<b>26</b>
B.1 Connecting formulations for QRM via a push-forward measure . . . . .	26
B.2 Connecting (2.2) to the essential supremum problem (3.1) . . . . .	27
<b>C Notes on KDE bandwidth selection</b>	<b>29</b>
<b>D Generalization bounds</b>	<b>29</b>
D.1 Main generalization bound and proof . . . . .	29
D.2 Kernel density estimator . . . . .	32
<b>E Further implementation details</b>	<b>35</b>
E.1 Algorithm . . . . .	35
E.2 ColoredMNIST . . . . .	35
E.3 DomainBed . . . . .	36
E.4 WILDS . . . . .	36
<b>F Connections between QRM and DRO</b>	<b>37</b>
F.1 Notation for this appendix . . . . .	37
F.2 (Strong) Duality of the superquantile . . . . .	37
<b>G Additional analyses and experiments</b>	<b>38</b>
G.1 Linear regression . . . . .	38
G.2 DomainBed . . . . .	40
G.3 WILDS . . . . .	45
<b>H Limitations of our work</b>	<b>46</b>

---

## A Causality

### A.1 Definitions and example

As in previous causal works on DG [9, 41, 53–55], our causality results assume all domains share the same underlying *structural causal model* (SCM) [56], with different domains corresponding to different interventions. For example, the different camera-trap deployments depicted in Fig. 1a may induce changes in (or interventions on) equipment, lighting, and animal-species prevalence rates.

**Definition A.1.** An SCM<sup>5</sup>  $\mathcal{M} = (\mathcal{S}, \mathbb{P}_N)$  consists of a collection of  $d$  structural assignments

$$\mathcal{S} = \{X_j \leftarrow g_j(\text{Pa}(X_j), N_j)\}_{j=1}^d, \quad (\text{A.1})$$

where  $\text{Pa}(X_j) \subseteq \{X_1, \dots, X_d\} \setminus \{X_j\}$  are the *parents* or *direct causes* of  $X_j$ , and  $\mathbb{P}_N = \prod_{j=1}^d \mathbb{P}_{N_j}$ , a joint distribution over the (jointly) independent noise variables  $N_1, \dots, N_d$ . An SCM  $\mathcal{M}$  induces a (“causal”) graph  $\mathcal{G}$  which is obtained by creating a node for each  $X_j$  and then drawing a directed edge from each parent in  $\text{Pa}(X_j)$  to  $X_j$ . We assume this graph to be acyclic.

We can draw samples from the *observational distribution*  $\mathbb{P}_{\mathcal{M}}(X)$  by first sampling a noise vector  $n \sim \mathbb{P}_N$ , and then using the structural assignments to generate a data point  $x \sim \mathbb{P}_{\mathcal{M}}(X)$ , recursively computing the value of every node  $X_j$  whose parents’ values are known. We can also manipulate or *intervene* upon the structural assignments of  $\mathcal{M}$  to obtain a related SCM  $\mathcal{M}^e$ .

**Definition A.2.** An *intervention*  $e$  is a modification to one or more of the structural assignments of  $\mathcal{M}$ , resulting in a new SCM  $\mathcal{M}^e = (\mathcal{S}^e, \mathbb{P}_N^e)$  and (potentially) new graph  $\mathcal{G}^e$ , with structural assignments

$$\mathcal{S}^e = \{X_j^e \leftarrow g_j^e(\text{Pa}^e(X_j^e), N_j^e)\}_{j=1}^d. \quad (\text{A.2})$$

We can draw samples from the *intervention distribution*  $\mathbb{P}_{\mathcal{M}^e}(X^e)$  in a similar manner to before, now using the modified structural assignments. We can connect these ideas to DG by noting that each intervention  $e$  creates a new domain or *environment*  $e$  with interventional distribution  $\mathbb{P}(X^e, Y^e)$ .

**Example A.3.** Consider the following linear SCM, with  $N_j \sim \mathcal{N}(0, \sigma_j^2)$ :

$$X_1 \leftarrow N_1, \quad Y \leftarrow X_1 + N_Y, \quad X_2 \leftarrow Y + N_2.$$

Here, interventions could replace the structural assignment of  $X_1$  with  $X_1^e \leftarrow 10$  and change the noise variance of  $X_2$ , resulting in a set of training environments  $\mathcal{E}_{\text{tr}} = \{\text{fix } X_1 \text{ to } 10, \text{ replace } \sigma_2 \text{ with } 10\}$ .

### A.2 EQRM recovers the causal predictor

**Overview.** We now prove that EQRM recovers the causal predictor in two stages. First, we prove the formal versions of Prop. 4.3, i.e. that EQRM learns a minimal invariant-risk predictor as  $\alpha \rightarrow 1$  when using the following estimators of  $\mathbb{T}_f$ : (i) a Gaussian estimator (Prop. A.4 of Appendix A.2.1); and (ii) kernel-density estimators with certain bandwidth-selection methods (Prop. A.5 of Appendix A.2.2). Second, we prove Thm. 4.4, i.e. that learning a minimal invariant-risk predictor is sufficient to recover the causal predictor under weaker assumptions than those of Peters et al. [54, Thm 2] and Krueger et al. [41, Thm 1] (Appendix A.2.3). Throughout this section, we consider the “population” setting within each domain (i.e.,  $n \rightarrow \infty$ ); in general, with only finitely-many observations from each domain, only approximate versions of these results are possible.

**Notation.** Given  $m$  training risks  $\{\mathcal{R}^{e_1}(f), \dots, \mathcal{R}^{e_m}(f)\}$  corresponding to the risks of a fixed predictor  $f$  on  $m$  training domains, let

$$\hat{\mu}_f = \frac{1}{m} \sum_{i=1}^m \mathcal{R}^{e_i}(f)$$

denote the sample mean and

$$\hat{\sigma}_f^2 = \frac{1}{m-1} \sum_{i=1}^m (\mathcal{R}^{e_i}(f) - \hat{\mu}_f)^2$$

the sample variance of the risks of  $f$ .

<sup>5</sup>A Non-parametric Structural Equation Model with Independent Errors (NP-SEM-IE) to be precise.

### A.2.1 Gaussian estimator

When using a Gaussian estimator for  $\widehat{\mathbb{T}}_f$ , we can rewrite the EQRM objective of (4.1) in terms of the standard-Normal inverse CDF  $\Phi^{-1}$  as

$$\hat{f}_\alpha := \arg \min_{f \in \mathcal{F}} \hat{\mu}_f + \Phi^{-1}(\alpha) \cdot \hat{\sigma}_f. \quad (\text{A.3})$$

Informally, we see that  $\alpha \rightarrow 1 \implies \Phi^{-1}(\alpha) \rightarrow \infty \implies \hat{\sigma}_f \rightarrow 0$ . More formally, we now show that, as  $\alpha \rightarrow 1$ , minimizing (A.3) leads to a predictor with minimal invariant-risk:

**Proposition A.4** (Gaussian QRM learns a minimal invariant-risk predictor as  $\alpha \rightarrow 1$ ). *Assume*

1.  $\mathcal{F}$  contains an invariant-risk predictor  $f_0 \in \mathcal{F}$  with finite mean risk (i.e.,  $\hat{\sigma}_{f_0} = 0$  and  $\hat{\mu}_{f_0} < \infty$ ), and
2. there are no arbitrarily negative mean risks (i.e.,  $\mu_* := \inf_{f \in \mathcal{F}} \mu_f > -\infty$ ).

Then, for the Gaussian QRM predictor  $\hat{f}_\alpha$  given in Eq. (A.3),

$$\lim_{\alpha \rightarrow 1} \hat{\sigma}_{\hat{f}_\alpha} = 0 \quad \text{and} \quad \limsup_{\alpha \rightarrow 1} \hat{\mu}_{\hat{f}_\alpha} \leq \hat{\mu}_{f_0}.$$

Prop. A.4 essentially states that, if an invariant-risk predictor exists, then Gaussian EQRM equalizes risks across the  $m$  domains, to a value at most the risk of the invariant-risk predictor. As we discuss in Appendix A.2.3, an invariant-risk predictor  $f_0$  (Assumption 1. of Prop. A.4 above) exists under the assumption that the mechanism generating the labels  $Y$  does not change between domains and is contained in the hypothesis class  $\mathcal{F}$ , together with a homoscedasticity assumption (see Appendix G.1.2). Meanwhile, Assumption 2. of Prop. A.4 above is quite mild and holds automatically for most loss functions used in supervised learning (e.g., squared loss, cross-entropy, hinge loss, etc.). We now prove Prop. A.4.

*Proof.* By definitions of  $\hat{f}_\alpha$  and  $f_0$ ,

$$\hat{\mu}_{\hat{f}_\alpha} + \Phi^{-1}(\alpha) \cdot \hat{\sigma}_{\hat{f}_\alpha} \leq \hat{\mu}_{f_0} + \Phi^{-1}(\alpha) \cdot \hat{\sigma}_{f_0} = \hat{\mu}_{f_0}. \quad (\text{A.4})$$

Since for  $\alpha \geq 0.5$  we have that  $\Phi^{-1}(\alpha) \hat{\sigma}_{\hat{f}_\alpha} \geq 0$ , it follows that  $\hat{\mu}_{\hat{f}_\alpha} \leq \hat{\mu}_{f_0}$ . Moreover, rearranging and using the definition of  $\mu_*$ , we obtain

$$\hat{\sigma}_{\hat{f}_\alpha} \leq \frac{\hat{\mu}_{f_0} - \hat{\mu}_{\hat{f}_\alpha}}{\Phi^{-1}(\alpha)} \leq \frac{\hat{\mu}_{f_0} - \mu_*}{\Phi^{-1}(\alpha)} \rightarrow 0 \quad \text{as} \quad \alpha \rightarrow 1.$$

□

**Connection to VREx.** For the special case of using a Gaussian estimator for  $\widehat{\mathbb{T}}_f$ , we can equate the EQRM objective of (A.3) with the  $\mathcal{R}_{\text{VREx}}$  objective of [41, Eq. 8]. To do so, we rewrite  $\mathcal{R}_{\text{VREx}}$  in terms of the sample mean and variance:

$$\arg \min_{f \in \mathcal{F}} \mathcal{R}_{\text{VREx}}(f) = \arg \min_{f \in \mathcal{F}} m \cdot \hat{\mu}_f + \beta \cdot \hat{\sigma}_f^2. \quad (\text{A.5})$$

Note that as  $\beta \rightarrow \infty$ ,  $\mathcal{R}_{\text{VREx}}$  learns a minimal invariant-risk predictor under the same assumptions, and by the same argument, as Prop. A.4. Dividing this objective by the positive constant  $m > 0$ , we can rewrite it in a form that allows a direct comparison of our  $\alpha$  parameter and this  $\beta$  parameter:

$$\arg \min_{f \in \mathcal{F}} \hat{\mu}_f + \left( \frac{\beta \cdot \hat{\sigma}_f}{m} \right) \cdot \hat{\sigma}_f. \quad (\text{A.6})$$

Comparing (A.6) and (A.3), we note the relation  $\beta = m \cdot \Phi^{-1}(\alpha) / \hat{\sigma}_f$  for a fixed  $f$ . For different  $f$ s, a particular setting of our parameter  $\alpha$  corresponds to different settings of Krueger et al.'s  $\beta$  parameter, depending on the sample standard deviation over training risks  $\hat{\sigma}_f$ .

### A.2.2 Kernel density estimator

We now consider the case of using a kernel density estimate, in particular,

$$\hat{F}_{\text{KDE},f}(x) = \frac{1}{m} \sum_{i=1}^m \Phi \left( \frac{x - R^{e_i}(f)}{h_f} \right) \quad (\text{A.7})$$

to estimate the cumulative risk distribution.

**Proposition A.5** (Kernel EQRM learns a minimal risk-invariant predictor as  $\alpha \rightarrow 1$ ). *Let*

$$\hat{f}_\alpha := \arg \min_{f \in \mathcal{F}} \hat{F}_{\text{KDE},f}^{-1}(\alpha),$$

be the kernel EQRM predictor, where  $\hat{F}_{\text{KDE},f}^{-1}$  denotes the quantile function computed from the kernel density estimate over (empirical) risks of  $f$  with a standard Gaussian kernel. Suppose we use a data-dependent bandwidth  $h_f$  such that  $h_f \rightarrow 0$  implies  $\hat{\sigma}_f \rightarrow 0$  (e.g., the ‘‘Gaussian-optimal’’ rule  $h_f = (4/3m)^{0.2} \cdot \hat{\sigma}_f$  [65]). As in Proposition A.4, suppose also that

1.  $\mathcal{F}$  contains an invariant-risk predictor  $f_0 \in \mathcal{F}$  with finite training risks (i.e.,  $\hat{\sigma}_{f_0} = 0$  and each  $R^{e_i}(f_0) < \infty$ ), and
2. there are no arbitrarily negative training risks (i.e.,  $R_* := \inf_{f \in \mathcal{F}, i \in [m]} R^{e_i}(f) > -\infty$ ).

For any  $f \in \mathcal{F}$ , let  $R_f^* := \min_{i \in [m]} R^{e_i}(f)$  denote the smallest of the (empirical) risks of  $f$  across domains. Then,

$$\lim_{\alpha \rightarrow 1} \hat{\sigma}_{\hat{f}_\alpha} = 0 \quad \text{and} \quad \limsup_{\alpha \rightarrow 1} R_{\hat{f}_\alpha}^* \leq R_{f_0}^*.$$

As in Prop. A.4, Assumption 1 depends on invariance of the label-generating mechanism across domains (as discussed further in Appendix A.2.3 below), while Assumption 2 automatically holds for most loss functions used in supervised learning. We now prove Prop. A.5.

*Proof.* By our assumption on the choice of bandwidth, it suffices to show that, as  $\alpha \rightarrow 1$ ,  $h_{\hat{f}_\alpha} \rightarrow 0$ .

Let  $\Phi$  denote the standard Gaussian CDF. Since  $\Phi$  is non-decreasing, for all  $x \in \mathbb{R}$ ,

$$\hat{F}_{\text{KDE},\hat{f}_\alpha}(x) = \frac{1}{m} \sum_{i=1}^m \Phi \left( \frac{x - R^{e_i}(\hat{f}_\alpha)}{h_{\hat{f}_\alpha}} \right) \leq \Phi \left( \frac{x - R_{\hat{f}_\alpha}^*}{h_{\hat{f}_\alpha}} \right).$$

In particular, for  $x = \hat{F}_{\text{KDE},\hat{f}_\alpha}^{-1}(\alpha)$ , we have

$$\alpha = \hat{F}_{\text{KDE},\hat{f}_\alpha}(\hat{F}_{\text{KDE},\hat{f}_\alpha}^{-1}(\alpha)) \leq \Phi \left( \frac{\hat{F}_{\text{KDE},\hat{f}_\alpha}^{-1}(\alpha) - R_{\hat{f}_\alpha}^*}{h_{\hat{f}_\alpha}} \right).$$

Inverting  $\Phi$  and rearranging gives

$$R_{\hat{f}_\alpha}^* + h_{\hat{f}_\alpha} \cdot \Phi^{-1}(\alpha) \leq \hat{F}_{\text{KDE},\hat{f}_\alpha}^{-1}(\alpha).$$

Hence, by definitions of  $\hat{f}_\alpha$  and  $f_0$ ,

$$R_{\hat{f}_\alpha}^* + h_{\hat{f}_\alpha} \cdot \Phi^{-1}(\alpha) \leq \hat{F}_{\text{KDE},\hat{f}_\alpha}^{-1}(\alpha) \leq \hat{F}_{\text{KDE},f_0}^{-1}(\alpha) = R_{f_0}^*. \quad (\text{A.8})$$

Since, for  $\alpha \geq 0.5$  we have that  $h_{\hat{f}_\alpha} \cdot \Phi^{-1}(\alpha) \geq 0$ , it follows that  $R_{\hat{f}_\alpha}^* \leq R_{f_0}^*$ . Moreover, rearranging Inequality (A.8) and using the definition of  $R_*$ , we obtain

$$h_{\hat{f}_\alpha} \leq \frac{R_{f_0}^* - R_{\hat{f}_\alpha}^*}{\Phi^{-1}(\alpha)} \leq \frac{R_{f_0}^* - R_*}{\Phi^{-1}(\alpha)} \rightarrow 0$$

as  $\alpha \rightarrow 1$ . □

### A.2.3 Causal recovery

We now discuss and prove our main result, Thm. 4.4, regarding the conditions under which the causal predictor is the only minimal invariant-risk predictor. Together with Props. A.4 and A.5, this provides the conditions under which EQRM successfully performs “causal recovery”, i.e., correctly recovers the true causal coefficients in a linear causal model of the data. As discussed in Appendix G.1.2, EQRM recovers the causal predictor by seeking *invariant risks* across domains, which differs from seeking *invariant functions* or coefficients (as in IRM [9]). As we discuss below, Thm. 4.4 generalizes related results in the literature regarding causal recovery based on *invariant risks* [41, 54].

**Assumption (v).** In contrast to both Peters et al. [54] and Krueger et al. [41], we do not require specific types of interventions on the covariates. In particular, our main assumption on the distributions of the covariates across domains, namely that the system of  $d$ -variate quadratic equations in (4.3) has a unique solution, is more general than these comparable results. For example, whereas both Peters et al. [54] and Krueger et al. [41] require one or more separate interventions for *every* covariate  $X_j$ , Example 4 below shows that we only require interventions on the subset of covariates that are effects of  $Y$ , while weaker conditions suffice for other covariates. Although this generality comes at the cost of abstraction, we now provide some concrete examples with different types of interventions to aid understanding. Note that, to simplify calculations and provide a more intuitive form, (4.3) of Thm. 4.4 assumes, without loss of generality, that all covariates are standardized to have mean 0 and variance 1, except where interventions change these. We can, however, rewrite (4.3) of Thm. 4.4 in a slightly more general form which does not require this assumption of standardized covariates:

$$\begin{aligned} 0 &\geq x^\top \mathbb{E}_{X \sim e_1} [XX^\top] x + 2x^\top \mathbb{E}_{N, X \sim e_1} [NX] \\ &= \dots \\ &= x^\top \mathbb{E}_{X \sim e_m} [XX^\top] x + 2x^\top \mathbb{E}_{N, X \sim e_m} [NX]. \end{aligned} \quad (\text{A.9})$$

We now present a number of concrete examples or special cases in which Assumption (v) of Thm. 4.4 would be satisfied, using this slightly more general form. In each example, we assume that variables are generated according to an SCM with an acyclic causal graph, as described in Appendix A.1.

1. *No effects of  $Y$ .* In the case that there are no effects of  $Y$  (i.e., no  $X_j$  is a causal descendant of  $Y$ , and hence each  $X_j$  is uncorrelated with  $N$ ), it suffices for there to exist at least one environment  $e_i$  in which the covariance  $\text{Cov}_{X \sim e_i} [X]$  has full rank. These are standard conditions for identifiability in linear regression. More generally, it suffices for  $\sum_{i=1}^m \text{Cov}_{X \sim e_i} [X]$  to have full rank; this is the same condition one would require if simply performing linear regression on the pooled data from all  $m$  environments. Intuitively, this full-rank condition guarantees that the observed covariate values are sufficiently uncorrelated to distinguish the effect of each covariate on the response  $Y$ . However, it does not necessitate interventions on the covariates, which are necessary to identify the *direction of causation* in a linear model; hence, this full-rank condition fails to imply causal recovery in the presence of effects of  $Y$ . See Appendix G.1.2 for a concrete example of this failure.
2. *Hard interventions.* For each covariate  $X_j$ , compared to some baseline environment  $e_0$ , there is some environment  $e_{X_j}$  arising from a hard single-node intervention  $do(X_j = z)$ , with  $z \neq 0$ . If  $X_j$  is any leaf node in the causal DAG, then in  $e_{X_j}$ ,  $X_j$  is uncorrelated with  $N$  and with each  $X_k$  ( $k \neq j$ ), so the inequality in (A.9) gives

$$0 \geq x^\top \mathbb{E}_{X \sim e_{X_j}} [XX^\top] x = x_j^2 z^2 + x_{-j}^\top \mathbb{E}_{X \sim e_0} [XX^\top] x_{-j}.$$

Since the matrix  $\mathbb{E}_{X \sim e} [XX^\top]$  is positive semidefinite (and  $z \neq 0$  implies  $z^2 > 0$ ), it follows that  $x_j = 0$ . The terms in (A.9) containing  $x_j$  thus vanish, and iterating this argument for parents of leaf nodes in the causal DAG, and so on, gives  $x = 0$ . This condition is equivalent to that in Theorem 2(a) of Peters et al. [54] and is a strict improvement over Corollary 2 of Yin et al. [66] and Theorem 1 of Krueger et al. [41], which respectively require two and three distinct hard interventions on each variable.

3. *Shift interventions.* For each covariate  $X_j$ , compared to some baseline environment  $e_0$ , there is some environment  $e_{X_j}$  consisting of the shift intervention  $X_j \leftarrow g_j(\text{Pa}(X_j), N_j) + z$ , for some  $z \neq 0$ . Recalling that we assumed each covariate was centered (i.e.,  $\mathbb{E}_{X \sim e_0} [X_k] = 0$ ) in  $e_0$ , if  $X_j$  is any leaf node in the causal DAG, then every other covariate remains centered in  $e_{X_j}$  (i.e.,

$\mathbb{E}_{X \sim e_{X_j}}[X_k] = 0$  for each  $k \neq j$ ). Hence, the excess risk is

$$x^\top \mathbb{E}_{X \sim e_{X_j}}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_{X_j}}[NX] = x_j^2 z^2 + x^\top \mathbb{E}_{X \sim e_0}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_0}[NX].$$

Since, by (A.9),

$$x^\top \mathbb{E}_{X \sim e_0}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_0}[NX] = x^\top \mathbb{E}_{X \sim e_{X_j}}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_{X_j}}[NX],$$

it follows that  $x_j^2 z^2 = 0$ , and so, since  $z \neq 0$ ,  $x_j = 0$ . As above, the terms in (A.9) containing  $x_j$  thus vanish, and iterating this argument for parents of leaf nodes in the causal DAG, and so on, gives  $x = 0$ . This condition is equivalent to the additive setting of Theorem 2(b) of Peters et al. [54].

4. *Noise interventions.* Suppose that each covariate is related to its causal parents through an additive noise model; i.e.,

$$X_j = g_j(\text{Pa}(X_j)) + N_j,$$

where  $\mathbb{E}[N_j] = 0$  and  $0 < \mathbb{E}[N_j^2] < \infty$ . Theorem 2(b) of Peters et al. [54] considers “noise” interventions, of the form

$$X_j \leftarrow g_j(\text{Pa}(X_j)) + \sigma N_j,$$

where  $\sigma^2 \neq 1$ . Suppose that, for each covariate  $X_j$ , compared to some baseline environment  $e_0$ , there exists an environment  $e_{X_j}$  consisting of the above noise intervention. If  $X_j$  is any leaf node in the causal DAG, then, since we assumed  $\mathbb{E}_{X \sim e_0}[X_j^2] = 1$ ,

$$\begin{aligned} & x^\top \mathbb{E}_{X \sim e_{X_j}}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_{X_j}}[NX] \\ &= (\sigma^2 - 1)x_j^2 \mathbb{E}[N_j^2] + x^\top \mathbb{E}_{X \sim e_0}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_0}[NX]. \end{aligned}$$

Hence, the system (A.9) implies  $0 = (\sigma^2 - 1)x_j^2 \mathbb{E}[N_j^2]$ . Since  $\sigma^2 \neq 1$  and  $\mathbb{E}[N_j^2] > 0$ , it follows that  $x_j = 0$ .

5. *Scale interventions.* For each covariate  $X_j$ , compared to some baseline environment  $e_0$ , there exist two environments  $e_{X_j, i}$  ( $i \in \{1, 2\}$ ) consisting of scale interventions  $X_j \leftarrow \sigma_i g_j(\text{Pa}(X_j), N_j)$ , for some  $\sigma_i \neq \pm 1$ , with  $\sigma_1 \neq \sigma_2$ . If  $X_j$  is any leaf node in the causal DAG, then, since we assumed  $\mathbb{E}_{X \sim e_0}[X_j^2] = 1$ ,

$$\begin{aligned} & x^\top \mathbb{E}_{X \sim e_{X_j}}[XX^\top]x + 2x^\top \mathbb{E}_{N, X \sim e_{X_j}}[NX] \\ &= (\sigma_i^2 - 1)x_j^2 + 2(\sigma_i - 1)x_j \mathbb{E}_{X \sim e_0}[X_j X_{-j}^\top]x_{-j}^\top + x^\top \mathbb{E}_{X \sim e_0}[XX^\top]x \\ &+ 2(\sigma_i - 1)x_j \mathbb{E}_{N, X \sim e_0}[X_j N] + 2x^\top \mathbb{E}_{N, X \sim e_0}[NX]. \end{aligned}$$

Hence, the system (A.9) implies

$$0 = (\sigma_i^2 - 1)x_j^2 + 2(\sigma_i - 1)x_j \left( \mathbb{E}_{X \sim e_0}[X_j X_{-j}^\top]x_{-j}^\top + \mathbb{E}_{N, X \sim e_0}[X_j N] \right).$$

Since  $\sigma_i^2 \neq 1$ , if  $x_j \neq 0$ , then solving for  $x_j$  gives

$$x_j = -2 \frac{\mathbb{E}_{X \sim e_0}[X_j X_{-j}^\top]x_{-j}^\top + \mathbb{E}_{N, X \sim e_0}[X_j N]}{\sigma_i + 1}.$$

Since  $\sigma_1 \neq \sigma_2$ , this is possible only if  $x_j = 0$ . This provides an example where a single intervention per covariate would be insufficient to guarantee causal recovery, but two distinct interventions per covariate suffice.

6. *Sufficiently uncorrelated causes and intervened-upon effects.* Suppose that, within the true causal DAG,  $\text{De}(Y) \subseteq [d]$  indexes the *descendants*, or *effects* of  $Y$  (e.g., in Figure 5,  $\text{De}(Y) = \{5, 6, 7\}$ ). Suppose that for every  $j \in \text{De}(Y)$ , compared to a single baseline environment  $e_0$ , there is

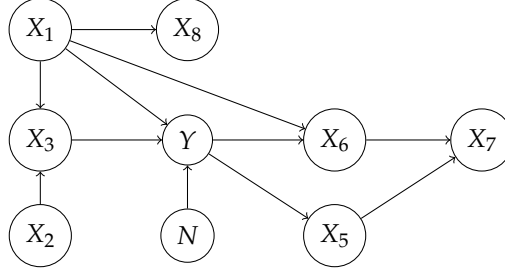


Figure 5: Example causal DAG with various types of covariates.  $X_1$  and  $X_3$  are the parents of  $Y$ , and so the true causal coefficient  $\beta$  has only two non-zero coordinates  $\beta_1$  and  $\beta_3$ .  $X_1$ ,  $X_2$ , and  $X_3$  are ancestors of  $Y$ .  $X_5$ ,  $X_6$ , and  $X_7$  are effects, or descendants, of  $Y$  and are the only covariates for which  $\mathbb{E}[X_j N]$  can be nonzero; hence,  $X_5$ ,  $X_6$ , and  $X_7$  are the only covariates on which interventions are generally necessary.

a environment  $e_{X_j}$  consisting of either a  $do(X_j = z)$  intervention or a shift intervention  $X_j \leftarrow g_j(\text{Pa}(X_j), N_j) + z$ , with  $z \neq 0$  and that the matrix

$$\sum_{i=1}^m \text{Cov}_{X \sim e_i} \left[ X_{[d] \setminus \text{De}(Y)} \right] \quad (\text{A.10})$$

has full rank. Then, as argued in the previous two cases, for each  $j \in \text{De}(Y)$ ,  $x_j = 0$ . Moreover, for any  $j \in [d] \setminus \text{De}(Y)$ ,  $\mathbb{E}[X_j N] = 0$ , and so the system of equations (A.9) reduces to

$$\begin{aligned} 0 &\geq x_{[d] \setminus \text{De}(Y)}^\top \mathbb{E}_{X \sim e_1} \left[ X_{[d] \setminus \text{De}(Y)} X_{[d] \setminus \text{De}(Y)}^\top \right] x_{[d] \setminus \text{De}(Y)} \\ &= \dots \\ &= x_{[d] \setminus \text{De}(Y)}^\top \mathbb{E}_{X \sim e_m} \left[ X_{[d] \setminus \text{De}(Y)} X_{[d] \setminus \text{De}(Y)}^\top \right] x_{[d] \setminus \text{De}(Y)}. \end{aligned}$$

Since each  $\mathbb{E}_{X \sim e_m} \left[ X_{[d] \setminus \text{De}(Y)} X_{[d] \setminus \text{De}(Y)}^\top \right]$  is positive semidefinite, the solution  $x = 0$  to this reduced system of equations is unique if (and only if) the matrix (A.10) has full rank. This example demonstrates that interventions are only needed for effect covariates, while a weaker full-rank condition suffices for the remaining ones. In many practical settings, it may be possible to determine *a priori* that a particular covariate  $X_j$  is not a descendant of  $Y$ ; in this case, the practitioner need not intervene on  $X_j$ , as long as sufficiently diverse observational data on  $X_j$  is available. To the best of our knowledge, this does not follow from any existing results in the literature, such as Theorem 2 of Peters et al. [54] or Corollary 2 of [66].

We conclude this section with the proof of Thm. 4.4:

*Proof.* Under the linear SEM setting with squared-error loss, for any estimator  $\hat{\beta}$ ,

$$\begin{aligned} \mathcal{R}^e(\hat{\beta}) &= \mathbb{E}_{N, X \sim e} \left[ ((\beta - \hat{\beta})^\top X + N)^2 \right] \\ &= \mathbb{E}_{X \sim e} \left[ ((\beta - \hat{\beta})^\top X)^2 \right] + 2\mathbb{E}_{N, X \sim e} [(\beta - \hat{\beta})^\top NX] + \mathbb{E}_N [N^2]. \end{aligned}$$

Since the second moment of the noise term  $\mathbb{E}_N [N^2]$  is equal to the risk  $\mathbb{E}_{(X, Y) \sim e} [(Y - \beta^\top X)^2]$  of the causal predictor  $\beta$ , by the definition of  $Y = \beta^\top X + N$ , we have that  $\mathbb{E}_N [N^2]$  is invariant across environments. Thus, minimizing the squared error risk  $\mathcal{R}^e(\hat{\beta})$  is equivalent to minimizing the excess risk

$$\begin{aligned} &\mathbb{E}_{X \sim e} \left[ ((\beta - \hat{\beta})^\top X)^2 \right] + 2\mathbb{E}_{N, X \sim e} [(\beta - \hat{\beta})^\top NX] \\ &= (\beta - \hat{\beta})^\top \mathbb{E}_{X \sim e} [XX^\top] (\beta - \hat{\beta}) + 2(\beta - \hat{\beta})^\top \mathbb{E}_{N, X \sim e} [NX] \end{aligned}$$

over estimators  $\hat{\beta}$ . Since the true coefficient  $\beta$  is an invariant-risk predictor with 0 excess risk, if  $\hat{\beta}$  is a minimal invariant-risk predictor, it has at most 0 invariant excess risk; i.e.,

$$\begin{aligned} 0 &\geq (\beta - \hat{\beta})^\top \mathbb{E}_{X \sim e_1} [XX^\top] (\beta - \hat{\beta}) + 2(\beta - \hat{\beta})^\top \mathbb{E}_{N, X \sim e_1} [NX] \\ &= \dots \\ &= (\beta - \hat{\beta})^\top \mathbb{E}_{X \sim e_m} [XX^\top] (\beta - \hat{\beta}) + 2(\beta - \hat{\beta})^\top \mathbb{E}_{N, X \sim e_m} [NX]. \end{aligned} \quad (\text{A.11})$$

By Assumption (v), the unique solution to this is  $\beta - \hat{\beta} = 0$ ; i.e.,  $\hat{\beta} = \beta$ .  $\square$

## B On the equivalence of different DG formulations

In Section 3, we claimed that under mild conditions, the minimax domain generalization problem in (2.2) is equivalent to the essential supremum problem in (3.1). In this subsection, we formally describe the conditions under which these two problems are equivalent. We also highlight several examples in which the assumptions needed to prove this equivalence hold.

Specifically, this appendix is organized as follows. First, in § B.1 we offer a more formal analysis of the equivalence between the probable domain general problems in (3.2) and (QRM). Next, in § B.2, we connect the domain generalization problem in (2.2) to the essential supremum problem in (3.1).

### B.1 Connecting formulations for QRM via a push-forward measure

To begin, we consider the abstract measure space  $(\mathcal{E}_{\text{all}}, \mathcal{A}, \mathbb{Q})$ , where  $\mathcal{A}$  is a  $\sigma$ -algebra defined on the subsets of  $\mathcal{E}_{\text{all}}$ . Recall that in our setting, the domains  $e \in \mathcal{E}_{\text{all}}$  are assumed to be drawn from the distribution  $\mathbb{Q}$ . Given this setting, in § 3 we introduced the probable domain generalization problem in (3.2), which we rewrite below for convenience:

$$\min_{f \in \mathcal{F}, t \in \mathbb{R}} t \quad \text{subject to} \quad \Pr_{e \sim \mathbb{Q}} \{ \mathcal{R}^e(f) \leq t \} \geq \alpha. \quad (\text{B.1})$$

Our objective is to formally show that this problem is equivalent to (QRM). To do so, for each  $f \in \mathcal{F}$ , let consider a second measurable space  $(\mathbb{R}_+, \mathcal{B})$ , where  $\mathbb{R}_+$  denotes the set of non-negative real numbers and  $\mathcal{B}$  denotes the Borel  $\sigma$ -algebra over this space. For each  $f \in \mathcal{F}$ , we can now define the  $(\mathbb{R}_+, \mathcal{B})$ -valued random variable<sup>6</sup>  $G_f : \mathcal{E}_{\text{all}} \rightarrow \mathbb{R}_+$  via

$$G_f : e \mapsto \mathcal{R}^e(f) = \mathbb{E}_{\mathbb{P}(X^e, Y^e)} [\ell(f(X^e), Y^e)]. \quad (\text{B.2})$$

Concretely,  $G_f$  maps an domain  $e$  to the corresponding risk  $\mathcal{R}^e(f)$  of  $f$  in that domain. In this way,  $G_f$  effectively summarizes  $e$  by its effect on our predictor's risk, thus projecting from the often-unknown and potentially high-dimensional space of possible distribution shifts or interventions to the one-dimensional space of observed, real-valued risks. However, note that  $G_f$  is not necessarily injective, meaning that two domains  $e_1$  and  $e_2$  may be mapped to the same risk value under  $G_f$ .

The utility of defining  $G_f$  is that it allows us to formally connect (3.2) with (QRM) via a push-forward measure through  $G_f$ . That is, given any  $f \in \mathcal{F}$ , we can define the measure<sup>7</sup>

$$\mathbb{T}_f =^d G_f \# \mathbb{Q} \quad (\text{B.3})$$

where  $\#$  denotes the *push-forward* operation and  $=^d$  denotes equality in distribution. Observe that the relationship in (B.3) allows us to explicitly connect  $\mathbb{Q}$ —the often unknown distribution over (potentially high-dimensional and/or non-Euclidean) domain shifts in Fig. 1b—to  $\mathbb{T}_f$ —the distribution over real-valued risks in Fig. 1c, from which we can directly observe samples. In this way, we find that for each  $f \in \mathcal{F}$ ,

$$\Pr_{e \sim \mathbb{Q}} \{ \mathcal{R}^e(f) \leq t \} = \Pr_{R \sim \mathbb{T}_f} \{ R \leq t \}. \quad (\text{B.4})$$

This relationship lays bare the connection between (3.2) and (QRM), in that the domain or environment distribution  $\mathbb{Q}$  can be replaced by a distribution over risks  $\mathbb{T}_f$ .

<sup>6</sup>For brevity, we will assume that  $G_f$  is always measurable with respect to the underlying  $\sigma$ -algebra  $\mathcal{A}$ .

<sup>7</sup>Here  $\mathbb{T}_f$  is defined over the induced measurable space  $(\mathbb{R}_+, \mathcal{B})$ .



## B.2 Connecting (2.2) to the essential supremum problem (3.1)

We now study the relationship between (2.2) and (3.1). In particular, in § B.2.1 and § B.2.2, we consider the distinct settings wherein  $\mathcal{E}_{\text{all}}$  comprises continuous and discrete spaces respectively.

### B.2.1 Continuous domain sets $\mathcal{E}_{\text{all}}$

When  $\mathcal{E}_{\text{all}}$  is a continuous space, it can be shown that (2.2) and (3.1) are *equivalent* whenever: (a) the map  $G_f$  defined in Section B.1 is continuous; and (b) the measure  $\mathbb{Q}$  satisfies very mild regularity conditions.

**The case when  $\mathbb{Q}$  is the Lebesgue measure.** Our first result concerns the setting in which  $\mathcal{E}_{\text{all}}$  is a subset of Euclidean space and where  $\mathbb{Q}$  is chosen to be the Lebesgue measure on  $\mathcal{E}_{\text{all}}$ . We summarize this result in the following proposition.

**Proposition B.1.** *Let us assume that the map  $G_f$  is continuous for each  $f \in \mathcal{F}$ . Further, let  $\mathbb{Q}$  denote the Lebesgue measure over  $\mathcal{E}_{\text{all}}$ ; that is, we assume that domains are drawn uniformly at random from  $\mathcal{E}_{\text{all}}$ . Then (2.2) and (3.1) are equivalent.*

*Proof.* To prove this claim, it suffices to show that under the assumptions in the statement of the proposition, it holds for any  $f \in \mathcal{F}$  that

$$\sup_{e \in \mathcal{E}_{\text{all}}} R^e(f) = \text{ess sup}_{e \sim \mathbb{Q}} R^e(f). \quad (\text{B.5})$$

To do so, let us fix an arbitrary  $f \in \mathcal{F}$  and write

$$A := \sup_{e \in \mathcal{E}_{\text{all}}} R^e(f) \quad \text{and} \quad B := \text{ess sup}_{e \sim \mathbb{Q}} R^e(f). \quad (\text{B.6})$$

At a high-level, our approach is to show that  $B \leq A$ , and then that  $A \leq B$ , which together will imply the result in (B.5). To prove the first inequality, observe that by the definition of the supremum, it holds that  $R^e(f) \leq A \forall e \in \mathcal{E}_{\text{all}}$ . Therefore,  $\mathbb{Q}\{e \in \mathcal{E}_{\text{all}} : R^e(f) > A\} = 0$ , which directly implies that  $B \leq A$ . Now for the second inequality, let  $\epsilon > 0$  be arbitrarily chosen. Consider that due to the continuity of  $G_f$ , there exists an  $e_0 \in \mathcal{E}_{\text{all}}$  such that

$$R^{e_0}(f) + \epsilon > A. \quad (\text{B.7})$$

Now again due to the continuity of  $G_f$ , we can choose a ball  $\mathcal{B}_\epsilon \subset \mathcal{E}_{\text{all}}$  centered at  $e_0$  such that  $|R^e(f) - R^{e_0}(f)| \leq \epsilon \forall e \in \mathcal{B}_\epsilon$ . Given such a ball, observe that  $\forall e \in \mathcal{B}_\epsilon$ , it holds that

$$R^e(f) \geq R^{e_0}(f) - \epsilon > A - 2\epsilon \quad (\text{B.8})$$

where the first inequality follows from the reverse triangle inequality and the second inequality follows from (B.7). Because  $\mathbb{Q}\{e \in \mathcal{B}_\epsilon : R^e(f) > A - 2\epsilon\} > 0$ , it directly follows that  $A - 2\epsilon \leq B$ . As  $\epsilon > 0$  was chosen arbitrarily, this inequality holds for any  $\epsilon > 0$ , and thus we can conclude that  $A \leq B$ , completing the proof.  $\square$

**Generalizing Prop. B.1 to other measure  $\mathbb{Q}$ .** We note that this proof can be generalized to measures  $\mathbb{Q}$  other than the Lebesgue measure. Indeed, the result holds for any measure  $\mathbb{Q}$  taking support on  $\mathcal{E}_{\text{all}}$  for which it holds that  $\mathbb{Q}$  places non-zero probability mass on any closed subset of  $\mathcal{E}_{\text{all}}$ . This would be the case, for instance, if  $\mathbb{Q}$  was a truncated Gaussian distribution with support on  $\mathcal{E}_{\text{all}}$ . Furthermore, if we let  $\mathbb{L}$  denote the Lebesgue measure on  $\mathcal{E}_{\text{all}}$ , then another more general instance of this property occurs whenever  $\mathbb{L}$  is absolutely continuous with respect to  $\mathbb{Q}$ , i.e., whenever  $\mathbb{L} \ll \mathbb{Q}$ .

**Corollary B.2.** *Let us assume that  $\mathbb{Q}$  places nonzero mass on every open ball with radius strictly larger than zero. Then under the continuity assumptions of Prop. B.1, it holds that (2.2) and (3.1) are equivalent.*

*Proof.* The proof of this fact follows along the same lines as that of Prop. B.1. In particular, the same argument shows that  $B \leq A$ . Similarly, to show that  $A \leq B$ , we can use the same argument, noting that  $\mathbb{Q}\{e \in \mathcal{B}_\epsilon : R^e(f) > A - 2\epsilon\}$  continues to hold, due to our assumption that  $\mathbb{Q}$  places nonzero mass on  $\mathcal{B}_\epsilon$ .  $\square$

**Examples.** We close this subsection by considering several real-world examples in which the conditions of Prop. B.1 hold. In particular, we focus on examples in the spirit of “Model-Based Domain Generalization” [22]. In this setting, it is assumed that the variation from domain to domain is parameterized by a *domain transformation model*  $x^e \mapsto D(x^e, e') =: x^{e'}$ , which maps the covariates  $x^e$  from a given domain  $e \in \mathcal{E}_{\text{all}}$  to another domain  $e' \in \mathcal{E}_{\text{all}}$ . As discussed in [22], domain transformation models cover settings in which inter-domain variation is due to *domain shift* [122, §1.8]. Indeed, under this model (formally captured by Assumptions 4.1 and 4.2 in [22]), the domain generalization problem in (2.2) can be equivalently rewritten as

$$\min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}_{\text{all}}} \mathbb{E}_{(X,Y)}[\ell(f(D(X,e)), Y)]. \quad (\text{B.9})$$

For details, see Prop. 4.3 in [22]. In this problem,  $(X, Y)$  denote an underlying pair of random variables such that

$$\mathbb{P}(X^e) =^d D \# (\mathbb{P}(X), \delta(e)) \quad \text{and} \quad \mathbb{P}(Y^e) =^d \mathbb{P}(Y) \quad (\text{B.10})$$

for each  $e \in \mathcal{E}_{\text{all}}$  where  $\delta(e)$  is a Dirac measure placed at  $e \in \mathcal{E}_{\text{all}}$ . Now, turning our attention back to Prop. B.1, we can show the following result for (B.9).

**Remark B.3.** Let us assume that the map  $e \mapsto D(\cdot, e)$  is continuous with respect to a metric  $d_{\mathcal{E}_{\text{all}}}(e, e')$  on  $\mathcal{E}_{\text{all}}$  and that  $x \mapsto \ell(x, \cdot)$  is continuous with respect to the absolute value. Further, assume that each predictor  $f \in \mathcal{F}$  is continuous in the standard Euclidean metric on  $\mathbb{R}^d$ . Then (2.2) and (3.1) are equivalent.

*Proof.* By Prop. B.1, it suffices to show that the map

$$G_f : e \mapsto \mathbb{E}_{(X,Y)}[\ell(f(D(X,e)), Y)] \quad (\text{B.11})$$

is a continuous function. To do so, recall that the composition of continuous functions is continuous, and therefore we have, by the assumptions listed in the above remark, that the map  $e \mapsto \ell(f(D(x,e)), y)$  is continuous for each  $(x, y) \sim (X, Y)$ . To this end, let us define the function  $h_f(x, y, e) := \ell(f(D(x,e)), y)$  and let  $\epsilon > 0$ . By the continuity of  $h_f$  in  $e$ , there exists a  $\delta = \delta(\epsilon) > 0$  such that  $|h_f(x, y, e) - h_f(x, y, e')| < \epsilon$  whenever  $d_{\mathcal{E}_{\text{all}}}(e, e') < \delta$ . Now observe that

$$\left| \mathbb{E}_{(X,Y)}[h_f(X, Y, e)] - \mathbb{E}_{(X,Y)}[h_f(X, Y, e')] \right| \quad (\text{B.12})$$

$$= \left| \int_{\mathcal{E}_{\text{all}}} h_f(X, Y, e) d\mathbb{P}(X, Y) - \int_{\mathcal{E}_{\text{all}}} h_f(X, Y, e') d\mathbb{P}(X, Y) \right| \quad (\text{B.13})$$

$$= \left| \int_{\mathcal{E}_{\text{all}}} (h_f(X, Y, e) - h_f(X, Y, e')) d\mathbb{P}(X, Y) \right| \quad (\text{B.14})$$

$$\leq \int_{\mathcal{E}_{\text{all}}} |h_f(X, Y, e) - h_f(X, Y, e')| d\mathbb{P}(X, Y). \quad (\text{B.15})$$

Therefore, whenever  $d_{\mathcal{E}_{\text{all}}}(e, e') < \delta$  it holds that

$$\left| \mathbb{E}_{(X,Y)}[h_f(X, Y, e)] - \mathbb{E}_{(X,Y)}[h_f(X, Y, e')] \right| \leq \int_{\mathcal{E}_{\text{all}}} \epsilon d\mathbb{P}(X, Y) = \epsilon \quad (\text{B.16})$$

by the monotonicity of expectation. This completes the proof that  $G_f$  is continuous.  $\square$

In this way, provided that the risks in each domain vary in a continuous way through  $e$ , (2.2) and (3.1) are equivalent. As a concrete example, consider an image classification setting in which the variation from domain to domain corresponds to different rotations of the images. This is the case, for instance, in the *RotatedMNIST* dataset [38, 127], wherein the training domains correspond to different rotations of the MNIST digits. Here, a domain transformation model  $D$  can be defined by

$$D(x, e) = R(e)x \quad \text{where} \quad e \in \mathcal{E}_{\text{all}} \subseteq [0, 2\pi), \quad (\text{B.17})$$

and where  $R(e)$  is a rotation matrix. In this case, it is clear that  $D$  is a continuous function of  $e$  (in fact, the map is *linear*), and therefore the result in (B.3) holds.

### B.2.2 Discrete domain sets $\mathcal{E}_{\text{all}}$

When  $\mathcal{E}_{\text{all}}$  is a discrete set, the conditions we require for (2.2) and (3.1) to be equivalent are even milder. In particular, the only restriction we place on the problems is that  $\mathbb{Q}$  must place non-zero mass on each element of  $\mathcal{E}_{\text{all}}$ ; that is,  $\mathbb{Q}(e) > 0 \forall e \in \mathcal{E}_{\text{all}}$ . We state this more formally below.

**Proposition B.4.** *Let us assume that  $\mathcal{E}_{\text{all}}$  is discrete, and that  $\mathbb{Q}$  is such that  $\forall e \in \mathcal{E}_{\text{all}}$ , it holds that  $\mathbb{Q}(e) > 0$ . Then it holds that (2.2) and (3.1) are equivalent.*

## C Notes on KDE bandwidth selection

In our setting, we are interested in bandwidth-selection methods which: (i) work well for 1D distributions and small sample sizes  $m$ ; and (ii) guarantee recovery of the causal predictor as  $\alpha \rightarrow 1$  by satisfying  $h_f \rightarrow 0 \implies \hat{\sigma}_f \rightarrow 0$ , where  $h_f$  is the data-dependent bandwidth and  $\hat{\sigma}_f$  is the sample standard deviation (see Appendices A.2.2 and A.2.3). We thus investigated three popular bandwidth-selection methods: (1) the Gaussian-optimal rule [65],  $h_f = (4/3m)^{0.2} \cdot \hat{\sigma}_f$ ; (2) Silverman’s rule-of-thumb [65],  $h_f = m^{-0.2} \cdot \min(\hat{\sigma}_f, \frac{\text{IQR}}{1.34})$ , with IQR the interquartile range; and (3) the median-heuristic [128–130], which sets the bandwidth to be the median pairwise-distance between data points. Note that many sensible methods exist, as do more complete studies on bandwidth selection—see e.g. [65].

For (i), we found Silverman’s rule-of-thumb [65] to perform very well, the Gaussian-optimal rule [65] to perform well, and the median-heuristic [128–130] to perform poorly. For (ii), only the Gaussian-optimal rule satisfies  $h_f \rightarrow 0 \implies \hat{\sigma}_f \rightarrow 0$ . Thus, in practice, we use either the Gaussian-optimal rule (particularly when causal predictor’s are sought as  $\alpha \rightarrow 1$ ), or Silverman’s rule-of-thumb.

## D Generalization bounds

This appendix states and proves our main generalization bound, Theorem D.1. Theorem D.1 applies for many possible estimates  $\hat{\mathbb{T}}_f$ , and we further show how to apply Theorem D.1 to the specific case of using a kernel density estimate.

### D.1 Main generalization bound and proof

Suppose that, from each of  $N$  IID environments  $e_1, \dots, e_N \sim \mathbb{P}(e)$ , we observe  $n$  IID labeled samples  $(X_{i,1}, Y_{i,1}), \dots, (X_{i,n}, Y_{i,n}) \sim \mathbb{P}(X^e, Y^e)$ . Fix a hypothesis class  $\mathcal{F}$  and confidence level  $\alpha \in [0, 1]$ . For any hypothesis  $f: \mathcal{X} \rightarrow \mathcal{Y}$ , define the *empirical risk on environment  $e_i$*  by

$$\hat{\mathcal{R}}^{e_i}(f) := \frac{1}{n} \sum_{j=1}^n \ell(Y_{i,j}, f(X_{i,j})), \quad \text{for each } i \in [N].$$

Throughout this section, we will abbreviate the distribution  $F_{\mathbb{T}_f}(t) = \Pr_e[\mathcal{R}^e(f) \leq t]$  of  $f$ ’s risk by  $F_f(t)$  and its estimate  $F_{\hat{\mathbb{T}}_f}$ , computed from the observed empirical risks  $\hat{\mathcal{R}}^{e_1}(f), \dots, \hat{\mathcal{R}}^{e_N}(f)$ , by  $\hat{F}_f$ .

We propose to select a hypothesis by minimizing this over our hypothesis class:

$$\hat{f} := \arg \min_{f \in \mathcal{F}} F_{\hat{\mathbb{T}}_f}^{-1}(\alpha). \quad (\text{D.1})$$

In this section, we prove a uniform generalization bound, which in particular, provides conditions under which the estimator (D.1) generalizes uniformly over  $\mathcal{F}$ . Because the novel aspect of the present paper is the notion of generalizing *across* environments, we will take for granted that the hypothesis class  $\mathcal{F}$  generalizes uniformly *within* each environments (i.e., that each  $\sup_{f \in \mathcal{F}} \mathcal{R}^{e_i}(f) - \hat{\mathcal{R}}^{e_i}(f)$  can be bounded with high probability); myriad generalization bounds from learning theory can be used to show this.

**Theorem D.1.** *Let  $\mathcal{G} := \{\hat{F}(\mathcal{R}^{e_1}(f), \mathcal{R}^{e_2}(f), \dots, \mathcal{R}^{e_N}(f)) : f \in \mathcal{F}, e_1, \dots, e_N \in \mathcal{E}_{\text{all}}\}$  denote the class of possible estimated risk distributions over  $N$  environments, and, for any  $\epsilon > 0$ , let  $\mathcal{N}_\epsilon(\mathcal{G})$*

denote the  $\epsilon$ -covering number of  $\mathcal{G}$  under  $\mathcal{L}_\infty(\mathbb{R})$ . Suppose the class  $\mathcal{F}$  generalizes uniformly within environments; i.e., for any  $\delta > 0$ , there exists  $t_{n,\delta,\mathcal{F}}$  such that

$$\operatorname{ess\,sup}_e \Pr_{\{(X_j, Y_j)\}_{j=1}^n \sim \mathbb{P}(X^e, Y^e)} \left[ \sup_{f \in \mathcal{F}} R^e(f) - \widehat{\mathcal{R}}^e(f) > t_{n,\delta,\mathcal{F}} \right] \leq \delta.$$

Let

$$\operatorname{Bias}(\mathcal{F}, \widehat{F}) := \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f(t) - \mathbb{E}_{e_1, \dots, e_N} [\widehat{F}_f(t)]$$

denote the worst-case bias of the estimator  $\widehat{F}$  over the class  $f$ . Noting that  $\widehat{F}_f$  is a function of the empirical risk CDF

$$\widehat{Q}_f(t) := \frac{1}{N} \sum_{i=1}^N \mathbf{1}\{\mathcal{R}^{e_i}(f) \leq t\},$$

suppose that the function  $\widehat{Q}_f \mapsto \widehat{F}_f$  is  $L$ -Lipschitz under  $\mathcal{L}_\infty(\mathbb{R})$ . Then, for any  $\epsilon, \delta > 0$ ,

$$\Pr_{\{(X_j, Y_j)\}_{j=1}^n \sim \mathbb{P}(X^{e_i}, Y^{e_i})} \left[ \sup_{f \in \mathcal{F}} F_f^{-1}(\alpha - B(\mathcal{F}, \widehat{F}) - \epsilon) - \widehat{F}_f^{-1}(\alpha) > t_{n, \frac{\epsilon}{N}, \mathcal{F}} \right] \leq \delta + 8\mathcal{N}_{\epsilon/16}(\mathcal{G})e^{-\frac{N\epsilon^2}{64L}}. \quad (\text{D.2})$$

The key technical observation of Theorem D.1 is that we can pull the supremum over  $\mathcal{F}$  outside the probability by incurring a  $\mathcal{N}_{\epsilon/16}(\mathcal{G})$  factor increase in the probability of failure. To ensure  $\mathcal{N}_{\epsilon/16}(\mathcal{G}) < \infty$ , we need to limit the space of possible empirical risk profiles  $\mathcal{G}$  (e.g., by kernel smoothing), incurring an additional bias term  $B(\mathcal{F}, \widehat{F})$ . As we demonstrate later, for common distribution estimators, such as kernel density estimators, one can bound the covering number  $\mathcal{N}_{\epsilon/16}(\mathcal{G})$  in Inequality (D.2) by standard methods, and the Lipschitz constant  $L$  is typically 1. Under mild (e.g., smoothness) assumptions on the family of possible true risk profiles, one can additionally bound the Bias Term, again by standard arguments.

Before proving Theorem D.1, we state two standard lemmas used in the proof:

**Lemma D.2** (Symmetrization; Lemma 2 of [131]). *Let  $X$  and  $X'$  be independent realizations of a random variable with respect to which  $\mathcal{F}$  is a family of integrable functions. Then, for any  $\epsilon > 0$ ,*

$$\Pr \left[ \sup_{f \in \mathcal{F}} f(X) - \mathbb{E} f(X) > \epsilon \right] \leq 2 \Pr \left[ \sup_{f \in \mathcal{F}} f(X) - f(X') > \frac{\epsilon}{2} \right].$$

**Lemma D.3** (Dvoretzky–Kiefer–Wolfowitz (DKW) Inequality; Corollary 1 of [132]). *Let  $X_1, \dots, X_n$  be IID  $\mathbb{R}$ -valued random variables with CDF  $P$ . Then, for any  $\epsilon > 0$ ,*

$$\Pr \left[ \sup_{t \in \mathbb{R}} \left| F_f(t) - \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{X_i \leq t\} \right| > \epsilon \right] \leq 2e^{-2n\epsilon^2}.$$

We now prove our main result, Theorem D.1.

*Proof of Theorem D.1.* For convenience, let  $F_f(t) := \mathbb{P}_{e \sim \mathbb{P}(e)}[R^e(f) \leq t]$ . In preparation for Symmetrization, for any  $f \in \mathcal{F}$ , let  $\widehat{F}'_f$  denote  $\widehat{F}_f$  computed on an independent “ghost” sample  $e'_1, \dots, e'_N \sim \mathbb{P}(e)$ . Let  $P_{\epsilon/16} \subseteq \mathcal{G}$  denote an  $(\epsilon/16)$ -cover of  $\mathcal{G}$  with  $|P_{\epsilon/16}| = \mathcal{N}_{\epsilon/16}$ . For any  $F \in \mathcal{G}$ , let  $DF \in \arg \min_{G \in P_{\epsilon/16}} \|G - F\|_\infty$  denote any projection of  $F$  onto  $P_{\epsilon/16}$ . Let  $\widehat{Q}_f$  denote

the empirical CDF, as defined in Theorem D.1. Then,

$$\Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} \mathbb{E}_{e_1, \dots, e_N} [\widehat{F}_f(t)] - \widehat{F}_f(t) > \epsilon \right] \quad (\text{D.3})$$

$$\leq 2 \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} \widehat{F}'_f(t) - \widehat{F}_f(t) > \epsilon/2 \right] \quad (\text{D.4})$$

$$\leq 2 \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \sup_{f \in \mathcal{F}} \|\widehat{F}'_f - \widehat{F}_f\|_\infty > \epsilon/2 \right] \quad (\text{D.5})$$

$$\leq 2 \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \sup_{f \in \mathcal{F}} \epsilon/8 + \|D\widehat{F}'_f - D\widehat{F}_f\|_\infty > \epsilon/2 \right] \quad (\text{D.6})$$

$$\leq 2\mathcal{N}_{\epsilon/16} \sup_{f \in \mathcal{F}} \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \epsilon/8 + \|D\widehat{F}'_f - D\widehat{F}_f\|_\infty > \epsilon/2 \right] \quad (\text{D.7})$$

$$\leq 2\mathcal{N}_{\epsilon/16} \sup_{f \in \mathcal{F}} \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \epsilon/4 + \|\widehat{F}'_f - \widehat{F}_f\|_\infty > \epsilon/2 \right] \quad (\text{D.8})$$

$$= 2\mathcal{N}_{\epsilon/16} \sup_{f \in \mathcal{F}} \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \|\widehat{F}'_f - \widehat{F}_f\|_\infty > \epsilon/4 \right] \quad (\text{D.9})$$

$$\leq 2\mathcal{N}_{\epsilon/16} \sup_{f \in \mathcal{F}} \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \|\widehat{Q}'_f - \widehat{Q}_f\|_\infty > \frac{\epsilon}{4L} \right] \quad (\text{D.10})$$

$$\leq 4\mathcal{N}_{\epsilon/16} \sup_{f \in \mathcal{F}} \Pr_{\substack{e_1, \dots, e_N \\ e'_1, \dots, e'_N}} \left[ \|\mathbb{E}[\widehat{Q}_f] - \widehat{Q}_f\|_\infty > \frac{\epsilon}{8L} \right] \quad (\text{D.11})$$

$$= 4\mathcal{N}_{\epsilon/16} \sup_{f \in \mathcal{F}} \Pr_{e_1, \dots, e_N} \left[ \sup_{t \in \mathbb{R}} \left| F_f(t) - \frac{1}{N} \sum_{i=1}^N 1\{\mathcal{R}^e(f) \leq t\} \right| > \frac{\epsilon}{8L} \right] \quad (\text{D.12})$$

$$\leq 8\mathcal{N}_{\epsilon/16} \exp\left(-\frac{N\epsilon^2}{64L}\right). \quad (\text{D.13})$$

Here, line (D.4) follows from the Symmetrization Lemma (Lemma D.2), lines (D.6) and (D.8) follow from the definition of  $D$ , line (D.7) is a union bound over  $\widehat{\mathcal{P}}_{\epsilon/16}$ , line (D.10) follows from the Lipschitz assumption, line (D.11) follows from the triangle inequality, line (D.12) follows from the fact that the empirical CDF is an unbiased estimate of the true CDF, and line (D.13) follows from the DKW Inequality (Lemma D.3).

Since  $\sup_x f(x) - \sup_x g(x) \leq \sup_x f(x) - g(x)$ ,

$$\begin{aligned} & \Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f(t) - \widehat{F}_f(t) > \epsilon + \text{Bias}(\mathcal{F}, \widehat{F}) \right] \\ &= \Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f(t) - \widehat{F}_f(t) > \epsilon + \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f(t) - \mathbb{E}_{e_1, \dots, e_N} [\widehat{F}_f(t)] \right] \\ &\leq \Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} \mathbb{E}_{e_1, \dots, e_N} [\widehat{F}_f(t)] - \widehat{F}_f(t) > \epsilon \right] \\ &\leq 8\mathcal{N}_{\epsilon/16} \exp\left(-\frac{N\epsilon^2}{64L}\right), \end{aligned} \quad (\text{D.14})$$

by (D.13). Meanwhile, applying the presumed uniform bound on within-environment generalization error together with a union bound over the  $N$  environments, gives us a high-probability bound on the

maximum generalization error of  $f$  within any of the  $N$  environments:

$$\Pr_{\substack{\{e_i\}_{i=1}^N \sim \mathbb{P}(e) \\ \{(X_{i,j}, Y_{i,j})\}_{j=1}^n \sim \mathbb{P}(X^{e_i}, Y^{e_i})}} \left[ \max_{i \in [N]} \sup_{f \in \mathcal{F}} \mathcal{R}^{e_i}(f) - \widehat{\mathcal{R}}^{e_i}(f) \leq t_{n, \frac{\delta}{2N}, \mathcal{F}} \right] \leq \delta/2,$$

It follows that, with probability at least  $1 - \delta/2$ , for all  $f \in \mathcal{F}$  and  $t \in \mathbb{R}$ ,

$$\widehat{F}_f \left( t + t_{n, \frac{\delta}{2N}, \mathcal{F}} \right) \leq \widehat{F}_{\widehat{\mathcal{R}}^{e_1}(f), \dots, \widehat{\mathcal{R}}^{e_1}(f)}(t),$$

where  $\widehat{F}_{\widehat{\mathcal{R}}^{e_1}(f), \dots, \widehat{\mathcal{R}}^{e_1}(f)}(t)$  is the actually empirical estimate  $\widehat{F}_f(t)$  of computed using the  $N$  empirical risks  $\widehat{\mathcal{R}}^{e_1}(f), \dots, \widehat{\mathcal{R}}^{e_N}(f)$ . Plugging this into the left-hand side of Inequality (D.14),

$$\Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f \left( t + t_{n, \frac{\delta}{2N}, \mathcal{F}} \right) - \widehat{F}_{\widehat{\mathcal{R}}^{e_1}(f), \dots, \widehat{\mathcal{R}}^{e_1}(f)}(t) > \epsilon + \text{Bias}(\mathcal{F}, \widehat{F}) \right] \leq 8\mathcal{N}_{\epsilon/16} \exp \left( -\frac{N\epsilon}{64L} \right).$$

Setting  $t = \widehat{F}_{\widehat{\mathcal{R}}^{e_1}(f), \dots, \widehat{\mathcal{R}}^{e_1}(f)}^{-1}(\alpha)$  and applying the non-decreasing function  $F_f^{-1}$  gives the desired result:

$$\Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f^{-1} \left( \alpha - \epsilon - \text{Bias}(\mathcal{F}, \widehat{F}) \right) - \widehat{F}_{\widehat{\mathcal{R}}^{e_1}(f), \dots, \widehat{\mathcal{R}}^{e_1}(f)}^{-1}(\alpha) \geq t_{n, \frac{\delta}{2N}, \mathcal{F}} \right] \leq 8\mathcal{N}_{\epsilon/16} \exp \left( -\frac{N\epsilon}{64L} \right).$$

□

## D.2 Kernel density estimator

In this section, we apply our generalization bound Theorem (D.1) to the kernel density estimator (KDE)

$$\widehat{F}_h(t) = \int_{-\infty}^t \frac{1}{nh} \sum_{i=1}^n K \left( \frac{\tau - X_i}{h} \right) d\tau$$

of the cumulative risk distribution under the assumptions that:

1. the loss  $\ell$  takes values in a bounded interval  $[a, b] \subseteq \mathbb{R}$ , and
2. for all  $f \in \mathcal{F}$ , the true risk profile  $F_f$  is  $\beta$ -Hölder continuous with constant  $L$ , for any  $\beta > 0$ .

We also make standard integrability and symmetry assumptions on the kernel  $K : \mathbb{R} \rightarrow \mathbb{R}$  (see Section 1.2.2 [133] for discussion of these assumptions):

$$\int_{\mathbb{R}} |K(u)| du < \infty, \quad \int_{\mathbb{R}} K(u) du = 1, \quad \int_{\mathbb{R}} |u|^\beta |K(u)| du < \infty,$$

and, for each positive integer  $j < \beta$ ,

$$\int_{\mathbb{R}} u^j K(u) du = 0. \tag{D.15}$$

We will use Theorem D.1 to show that, for an appropriately chosen bandwidth  $h$ ,

$$\sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f(t) - \widehat{F}_f(t) \in O_p \left( \left( \frac{\log N}{N} \right)^{\frac{\beta}{2\beta+1}} \right).$$

We start by bounding the bias term  $B(\mathcal{F}, \widehat{F})$ . Since

$$\begin{aligned} \mathbb{E}_{X_1, \dots, X_n} \left[ \left| \int_{-\infty}^t \frac{1}{nh} \sum_{i=1}^n K \left( \frac{\tau - X_i}{h} \right) d\tau \right| \right] &\leq \frac{1}{h} \mathbb{E}_X \left[ \left| \int_{-\infty}^{\infty} K \left( \frac{\tau - X_i}{h} \right) d\tau \right| \right] \\ &\leq \|K\|_1 < \infty, \end{aligned}$$

applying Fubini's theorem, linearity of expectation, the change of variables  $x \mapsto \tau + xh$ , Fubini's theorem again, and the fact that  $\int_{\mathbb{R}} K(u) dx = 1$ ,

$$\begin{aligned}
F_f(t) - \mathbb{E}_{X_1, \dots, X_n} [\widehat{F}_h(t)] &= F_f(t) - \mathbb{E}_{e_1, \dots, e_N} \left[ \int_{-\infty}^t \frac{1}{nh} \sum_{i=1}^n K \left( \frac{\tau - X_i}{h} \right) \right] \\
&= F_f(t) - \int_{-\infty}^t \mathbb{E}_{X_1, \dots, X_n} \left[ \frac{1}{nh} \sum_{i=1}^n K \left( \frac{\tau - X_i}{h} \right) \right] \\
&= F_f(t) - \int_{-\infty}^t \int_{\mathbb{R}} \frac{1}{h} K \left( \frac{\tau - x}{h} \right) p(x) dx d\tau \\
&= F_f(t) - \int_{-\infty}^t \int_{\mathbb{R}} K(x) p(\tau + xh) dx d\tau \\
&= F_f(t) - \int_{\mathbb{R}} K(x) \int_{-\infty}^t p(\tau + xh) d\tau dx \\
&= \int_{\mathbb{R}} K(x) (F_f(t) - F(t + xh)) dx.
\end{aligned}$$

By Taylor's theorem for some  $\pi \in [0, 1]$ ,

$$F(t + xh) = \sum_{j=0}^{[\beta]-1} \frac{(xh)^j}{j!} \frac{d^j}{dt^j} F_f(t) + \frac{(xh)^{[\beta]}}{[\beta]!} \frac{d^{[\beta]}}{dt^{[\beta]}} F(t + \pi xh).$$

Hence, by the assumption (D.15),

$$\begin{aligned}
F_f(t) - \mathbb{E}_{X_1, \dots, X_n} [\widehat{F}_h(t)] &= \int_{\mathbb{R}} K(x) \left( F_f(t) - \sum_{j=0}^{[\beta]-1} \frac{(xh)^j}{j!} \frac{d^j}{dt^j} F_f(t) + \frac{(xh)^{[\beta]}}{[\beta]!} \frac{d^{[\beta]}}{dt^{[\beta]}} F(t + \pi xh) \right) dx \\
&= \int_{\mathbb{R}} K(x) \left( \frac{(xh)^{[\beta]}}{[\beta]!} \frac{d^{[\beta]}}{dt^{[\beta]}} F(t + \pi xh) \right) dx \\
&= \int_{\mathbb{R}} K(x) \frac{(xh)^{[\beta]}}{[\beta]!} \left( \frac{d^{[\beta]}}{dt^{[\beta]}} F(t + \pi xh) - \frac{d^{[\beta]}}{dt^{[\beta]}} F_f(t) \right) dx.
\end{aligned}$$

Thus, by the Hölder continuity assumption,

$$\begin{aligned}
\left| F_f(t) - \mathbb{E}_{X_1, \dots, X_n} [\widehat{F}_h(t)] \right| &\leq \int_{\mathbb{R}} K(x) \frac{(xh)^{[\beta]}}{[\beta]!} \left| \frac{d^{[\beta]}}{dt^{[\beta]}} F(t + \pi xh) - \frac{d^{[\beta]}}{dt^{[\beta]}} F_f(t) \right| dx \\
&\leq \int_{\mathbb{R}} K(x) \frac{(xh)^{[\beta]}}{[\beta]!} L(\pi xh)^{\beta - [\beta]} dx \leq Ch^\beta, \tag{D.16}
\end{aligned}$$

where  $C := \frac{L}{[\beta]!} \int_{\mathbb{R}} |x|^\beta |K(x)| dx$  is a constant.

Next, since, by the Fundamental Theorem of Calculus,

$$\frac{d^{[\beta+1]}}{dt^{[\beta+1]}} \widehat{F}_f(t) = \frac{d^{[\beta+1]}}{dt^{[\beta+1]}} \int_{-\infty}^t \frac{1}{nh} \sum_{i=1}^N K \left( \frac{\tau - X_i}{h} \right) d\tau = \frac{1}{nh} \sum_{i=1}^N \frac{d^{[\beta+1]}}{dt^{[\beta+1]}} K \left( \frac{t - X_i}{h} \right),$$

$\|F_f\|_{C^{\beta+1}} \leq \|K_h\|_{C^\beta} = h^{-(\beta+1)} \|K\|_{C^\beta}$ . Hence, by standard bounds on the covering number of Hölder continuous functions [134], there exists a constant  $c > 0$  depending only on  $\beta$  such that

$$\mathcal{N}_{\epsilon/16}(\mathcal{N}) \leq \exp \left( c(b-a) \left( \frac{\|K\|_{C^\beta}}{h^{\beta+1}\epsilon} \right)^{\frac{1}{\beta+1}} \right) = \exp \left( c \frac{(b-a)}{h} \left( \frac{\|K\|_{C^\beta}}{\epsilon} \right)^{\frac{1}{\beta+1}} \right). \tag{D.17}$$

Finally, since  $\widehat{F}_h = \widehat{Q} * K_h$  (where  $*$  denotes convolution), by linearity of the convolution and Young's convolution inequality [135, p.34],

$$\left\| \widehat{F}_h - \widehat{F}'_h \right\|_{\infty} \leq \left\| \widehat{Q} - \widehat{Q}' \right\|_{\infty} \|K_h\|_1.$$

Since, by a change of variables,  $\|K_h\|_1 = \|K\|_1 = 1$ , the KDE is a 1-Lipschitz function of the empirical CDF, under  $\mathcal{L}_\infty(\mathbb{R})$ .

Thus, plugging Inequality (D.16), Inequality (D.17), and  $L = 1$  into Theorem D.1 and taking  $n \rightarrow \infty$  gives, for any  $\epsilon > 0$ ,

$$\Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}} F_f^{-1} \left( \alpha - Ch^\beta - \epsilon \right) - \widehat{F}_f^{-1}(\alpha) > 0 \right] \leq 8 \exp \left( c \frac{b-a}{h} \left( \frac{\|K\|_{C^\beta}}{\epsilon} \right)^{\frac{1}{\beta+1}} \right) e^{-\frac{N\epsilon^2}{64}}.$$

Plugging in  $\epsilon = \sqrt{\frac{\log \frac{1}{\delta} + c \frac{b-a}{h}}{N}}$  gives

$$\Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}} F_f^{-1} \left( \alpha - Ch^\beta - \sqrt{\frac{\log \frac{1}{\delta} + c \frac{b-a}{h}}{N}} \right) - \widehat{F}_f^{-1}(\alpha) > 0 \right] \leq \delta.$$

This bound is optimized by  $h \asymp \left( (b-a) \frac{\log N}{N} \right)^{\frac{1}{2\beta+1}}$ , giving an overall bound of

$$\begin{aligned} & \Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}, t \in \mathbb{R}} F_f(t) - \widehat{F}_f(t) > ch^{\frac{\beta}{2\beta+1}} \right] \leq \delta \\ & \Pr_{e_1, \dots, e_N} \left[ \sup_{f \in \mathcal{F}} F_f^{-1} \left( \alpha - ch^{\frac{\beta}{2\beta+1}} + \sqrt{\frac{\log \frac{1}{\delta}}{N}} \right) - \widehat{F}_f^{-1}(\alpha) > 0 \right] \leq \delta. \end{aligned}$$

for some  $c > 0$ . In particular, as  $N, n \rightarrow \infty$ , the EQRM estimate  $\widehat{f}$  satisfies

$$F_{\widehat{f}}^{-1}(\alpha) \rightarrow \inf_{f \in \mathcal{F}} F_f^{-1}(\alpha).$$



## E Further implementation details

### E.1 Algorithm

Below we detail the EQRM algorithm. Note that: (i) any distribution estimator may be used in place of DIST so long as the functions DIST.ESTIMATE\_PARAMS and DIST.ICDF are differentiable; (ii) other bandwidth-selection methods may be used on line 14, with the Gaussian-optimal rule serving as the default; and (iii) the bisection method BISECT on line 20 requires an additional parameter, the maximum number of steps, which we always set to 32.

---

#### Algorithm 1: Empirical Quantile Risk Minimization (EQRM).

---

**Input:** Predictor  $f_\theta$ , loss function  $\ell$ , desired probability of generalization  $\alpha$ , learning rate  $\eta$ , distribution estimator DIST,  $M$  datasets with  $D^m = \{(x_i^m, y_i^m)\}_{i=1}^{n_m}$ .

- 1 Initialize  $f_\theta$ ;
- 2 **while not converged do**
  - 3  $L^m \leftarrow \frac{1}{n_m} \sum_{i=1}^{n_m} \ell(f_\theta(x_i^m), y_i^m)$ , for  $m = 1, \dots, M$ ; \*/
  - 4  $\widehat{\mathbb{T}}_f \leftarrow \text{DIST.ESTIMATE\_PARAMS}(\mathbf{L})$ ; \*/
  - 5  $q \leftarrow \text{DIST.ICDF}(\alpha)$ ; \*/
  - 6  $\theta \leftarrow \theta - \eta \cdot \nabla_{\theta} q$ ; \*/

**Output:**  $f_\theta$

- 7 **Procedure** GAUSS.ESTIMATE\_PARAMS( $\mathbf{L}$ )
  - 8  $\hat{\mu} \leftarrow \frac{1}{M} \sum_{m=1}^M L^m$ ; \*/
  - 9  $\hat{\sigma}^2 \leftarrow \frac{1}{M-1} \sum_{m=1}^M (L^m - \hat{\mu})^2$ ; \*/
- 10 **Procedure** GAUSS.ICDF( $\alpha$ )
  - 11 **return**  $\hat{\mu} + \hat{\sigma} \cdot \Phi^{-1}(\alpha)$ ;
- 12 **Procedure** KDE.ESTIMATE\_PARAMS( $\mathbf{L}$ )
  - 13  $\hat{\sigma}^2 \leftarrow \frac{1}{M-1} \sum_{m=1}^M (L^m - \frac{1}{M} \sum_{j=1}^M L^j)^2$ ; \*/
  - 14  $h \leftarrow (\frac{4}{3M})^{0.2} \cdot \hat{\sigma}$  \*/
- 15 **Procedure** KDE.ICDF( $\alpha$ )
  - 16  $F_m(x') \leftarrow L^m + h \cdot \Phi(x')$ ; \*/
  - 17  $F(x') \leftarrow \frac{1}{M} \sum_{m=1}^M F_m(x')$ ; \*/
  - 18  $\text{mn} \leftarrow \min_m F_m^{-1}(\alpha)$ ; \*/
  - 19  $\text{mx} \leftarrow \max_m F_m^{-1}(\alpha)$ ; \*/
  - 20 **return** BISECT( $F, \alpha, \text{mn}, \text{mx}$ ); \*/

---

### E.2 ColoredMNIST

For the CMNIST results of § 6.1, we used full batches (size 25000), 400 steps for ERM pretraining, 600 total steps for IRM, VREx, EQRM, and 1000 total steps for GroupDRO, SD, and IGA. We used the original MNIST training set to create training and validation sets for each domain, and the original MNIST test set for the test sets of each domain. We also decayed the learning rate using cosine annealing/scheduling. We swept over penalty weights in  $\{50, 100, 500, 1000, 5000\}$  for IRM,

VREx and IGA, penalty weights in  $\{0.001, 0.01, 0.1, 1\}$  for SD,  $\eta$ 's in  $\{0.001, 0.01, 0.1, 0.5, 1.0\}$  for GroupDRO, and  $\alpha$ 's in  $1 - \{e^{-100}, e^{-250}, e^{-500}, e^{-750}, e^{-1000}\}$  for EQRm. To allow these values of  $\alpha$ , which are *very* close to 1, we used an asymptotic expression for the Normal inverse CDF, namely  $\Phi^{-1}(\alpha) \approx \sqrt{-2 \ln(1 - \alpha)}$  as  $\alpha \rightarrow 1$  [136]. This allowed us to parameterize  $\alpha = 1 - e^{-1000}$  as  $\ln(1 - \alpha) = \ln(e^{-1000}) = -1000$ , avoiding issues with floating-point precision. As is the standard for CMNIST, we used a test-domain validation set to select the best settings (after the total number of steps), then reported the mean and standard deviation over 10 random seeds on a test-domain test set. As in previous works, the hyperparameter ranges of all methods were selected by peeking at test-domain performance. While not ideal, this is quite difficult to avoid with CMNIST and highlights the problem of model selection more generally in DG—as discussed by many previous works [9, 38, 41, 115]. Finally, we note several observations from our CMNIST, WILDS and DomainBed experiments which, despite not being thoroughly investigated with their own set of experiments (yet), may prove useful for future work: (i) ERM pretraining seems an effective strategy for DG methods, and can likely replace the more delicate penalty-annealing strategies (as also observed in [115]); (ii) lowering the learning rate after ERM pretraining seems to stabilize DG methods; and (iii) EQRm often requires a lower learning rate than other DG methods after ERM pretraining, with its loss and gradients tending to be significantly larger.

### E.3 DomainBed

For EQRm, we used the default algorithm setup: a kernel-density estimator of the risk distribution with the ‘‘Gaussian-optimal’’ rule [65] for bandwidth selection. We used the standard hyperparameter-sampling procedure of Domainbed, running over 3 trials for 20 randomly-sampled hyperparameters per trial. For EQRm, this involved:

Hparam	Default	Sampling
$\alpha$	0.75	$U(0.5, 0.99)$
Burn-in/anneal iters	2500	$10^k$ , with $k \sim U(2.5, 3.5)$
EQRm learning rate (post burn-in)	$10^{-6}$	$10^k$ , with $k \sim U(-7, -5)$

All other all hyperparameters remained as their DomainBed-defaults, while the baseline results were taken directly from the most up-to-date DomainBed tables<sup>8</sup>. See our code for further details.

### E.4 WILDS

We considered two WILDS datasets: iWildCam and OGB-MolPCBA (henceforth OGB). For both of these datasets, we used the architectures use in the original WILDS paper [12]; that is, for iWildCam we used a ResNet-50 architecture [137] pretrained on ImageNet [138], and for OGB, we used a Graph Isomorphism Network [139] combined with virtual nodes [140]. To perform model-selection, we followed the guidelines provided in the original WILDS paper [12]. In particular, for each of the baselines we consider, we performed grid searches over the hyperparameter ranges listed in [12] with respect to the given validation sets; see [12, Appendices E.1.2 and E.4.2] for a full list of these hyperparameter ranges.

**EQRm.** For both datasets, we ran EQRm with KDE using the Gaussian-optimal bandwidth-selection method. All EQRm models were initialized with the same ERM checkpoint, which is obtained by training ERM using the code provided by [12]. Following [12], for iWildCam, we trained ERM for 12 epochs, and for OGB, we trained ERM for 100 epochs. We again followed [12] by using a batch size of 32 for iWildCam and 8 groups per batch. For OGB, we performed grid searches over the batch size in the range  $B \in \{32, 64, 128, 256, 512, 1024, 2048\}$ , and we used  $0.25B$  groups per batch. We selected the learning rate for EQRm from  $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$ .

**Computational resources.** All experiments on the WILDS datasets were run across two four-GPU workstations, comprising a total of eight Quadro RTX 5000 GPUs.

<sup>8</sup>[https://github.com/facebookresearch/DomainBed/tree/main/domainbed/results/2020\\_10\\_06\\_7df6f06](https://github.com/facebookresearch/DomainBed/tree/main/domainbed/results/2020_10_06_7df6f06)

## F Connections between QRM and DRO

In this appendix we draw connections between quantile risk minimization (QRM) and distributionally robust optimization (DRO) by considering an alternative optimization problem which we call *superquantile risk minimization*<sup>9</sup>:

$$\min_{f \in \mathcal{F}} \text{SQ}_\alpha(R; \mathbb{T}_f) \quad \text{where} \quad \text{SQ}_\alpha(R; \mathbb{T}_f) := \mathbb{E}_{R \sim \mathbb{T}_f} \left[ R \mid R \geq F_{\mathbb{T}_f}^{-1}(\alpha) \right]. \quad (\text{F.1})$$

Here,  $\text{SQ}_\alpha$  represents the *superquantile*—also known as the *conditional value-at-risk* (CVaR) or *expected tail loss*—at level  $\alpha$ , which can be seen as the conditional expectation of a random variable  $R$  subject to  $R$  being larger than the  $\alpha$ -quantile  $F^{-1}(\alpha)$ . In our case, where  $R$  represents the statistical risk on a randomly-sampled environment,  $\text{SQ}_\alpha$  can be seen as the expected risk in the worst  $100 \cdot (1 - \alpha)\%$  of cases/domains. Below, we exploit the well-known duality properties of CVaR to formally connect (QRM) and GroupDRO [45]; see Prop. F.1 for details.

### F.1 Notation for this appendix

Throughout this appendix, for each  $f \in \mathcal{F}$ , we will let the risk random variable  $R$  be defined on the probability space  $(\mathbb{R}_+, \mathcal{B}, \mathbb{T}_f)$ , where  $\mathbb{R}_+$  denotes the nonnegative real numbers and  $\mathcal{B}$  denotes the Borel  $\sigma$ -algebra on  $\mathbb{R}_+$ . We will also consider the Lebesgue spaces  $L^p := L^p(\mathbb{R}_+, \mathcal{B}, \mathbb{T}_f)$  of functions  $h$  for which  $\mathbb{E}_{r \sim \mathbb{T}_f}[|h(r)|^p]$  is finite. For conciseness, we will use the notation

$$\langle g(r), h(r) \rangle := \int_{r \geq 0} g(r)h(r)dr \quad (\text{F.2})$$

to denote the standard inner product on  $\mathbb{R}_+$ . Furthermore, we will use the notation  $\mathbb{U} \ll \mathbb{V}$  to signify that  $\mathbb{U}$  is *absolutely continuous* with respect to  $\mathbb{V}$ , meaning that if  $\mathbb{U}(A) = 0$  for every set  $A$  for which  $\mathbb{V}(A) = 0$ . We also use the abbreviation ‘‘a.e.’’ to mean ‘‘almost everywhere.’’ Finally, the notation  $\Pi_{[a,b]}(c)$  denotes the projection of a number  $c$  into the real interval  $[a, b]$ .

### F.2 (Strong) Duality of the superquantile

We begin by proving that strong duality holds for the superquantile function  $\text{SQ}_\alpha$ . We note that this duality result is well-known in the literature (see, e.g., [90]), and has been exploited in the context of adaptive sampling [94] and offline reinforcement learning [141]. We state this result and proof for the sake of exposition.

**Proposition F.1** (Dual representation of  $\text{SQ}_\alpha$ ). *If  $R \in L^p$  for some  $p \in (1, \infty)$ , then*

$$\text{SQ}_\alpha(R; \mathbb{T}_f) = \max_{\mathbb{U} \in \mathcal{U}_f(\alpha)} \mathbb{E}_{\mathbb{U}}[R] \quad (\text{F.3})$$

where the uncertainty set  $\mathcal{U}_f(\alpha)$  is defined as

$$\mathcal{U}_f(\alpha) := \left\{ \mathbb{U} \in L^q : \mathbb{U} \ll \mathbb{T}_f, \mathbb{U} \in [0, 1/(1-\alpha)] \text{ a.e.}, \|\mathbb{U}\|_{L^1} = 1 \right\}. \quad (\text{F.4})$$

*Proof.* Note that the primal objective can be equivalently written as

$$\text{SQ}_\alpha(R; \mathbb{T}_f) = \min_{t \in \mathbb{R}} \left\{ t + \frac{1}{1-\alpha} \langle (R-t)_+, \mathbb{T}_f \rangle \right\} \quad (\text{F.5})$$

where  $(z)_+ = \max\{0, z\}$  [97], which in turn has the following epigraph form:

$$\min_{t \in \mathbb{R}, s \in L_+^p} \quad t + \frac{1}{1-\alpha} \langle s, \mathbb{T}_f \rangle \quad (\text{F.6})$$

$$\text{subject to} \quad R(r) - t \leq s(r) \text{ a.e. } r \in \mathbb{R}_+. \quad (\text{F.7})$$

<sup>9</sup>This definition assumes that  $\mathbb{T}_f$  is continuous; for a more general treatment, see [97].

When written in Lagrangian form, we can express this problem as

$$\min_{t \in \mathbb{R}} \max_{s \in L_+^p, \lambda \in L_+^q} \left\{ t(1 - \langle \mathbf{1}, \lambda \rangle) + \left\langle s, \frac{1}{1-\alpha} \mathbb{T}_f - \lambda \right\rangle + \langle R, \lambda \rangle \right\}. \quad (\text{F.8})$$

Note that this objective is *linear* in  $t$ ,  $s$ , and  $\lambda$ , and therefore due to the strong duality of linear programs, we can optimize over  $s$ ,  $t$ , and  $\lambda$  in any order [142]. Minimizing over  $t$  reveals that the problem is unbounded unless  $\int_{r \geq 0} \lambda(r) dr = 1$ , meaning that  $\lambda$  is a probability distribution since  $\lambda(r) \geq 0$  almost everywhere. Thus, the problem can be written as

$$\min_{s \in L_+^p} \max_{\lambda \in \mathcal{P}(\mathbb{R}_+)} \left\{ \left\langle s, \frac{1}{1-\alpha} \mathbb{T}_f - \lambda \right\rangle + \langle R, \lambda \rangle \right\} \quad (\text{F.9})$$

where  $\mathcal{P}^q(\mathbb{R}_+)$  denotes the subspace of  $L^q$  of probability distributions on  $\mathbb{R}_+$ .

Now consider the maximization over  $s$ . Note that if there is a set  $A \subset \mathcal{E}_{\text{all}}$  of nonzero Lebesgue measure on which  $\lambda(A) \geq (1/1-\alpha)\mathbb{T}_f(A)$ , then the problem is unbounded below because  $s(A)$  can be made arbitrarily large. Therefore, it must be the case that  $\lambda \leq (1/1-\alpha)\mathbb{T}_f$  almost everywhere. On the other hand, if  $\lambda(A) \leq (1/1-\alpha)\mathbb{T}_f(A)$ , then  $s(A) = 0$  minimizes the first term in the objective. Therefore,  $s$  can be eliminated provided that  $\lambda \leq (1/1-\alpha)\mathbb{T}_f$  almost everywhere. Thus, we can write the problem as

$$\max_{\lambda \in \mathcal{P}^q(\mathbb{R}_+)} \quad \langle R, \lambda \rangle = \mathbb{E}_\lambda[R] \quad (\text{F.10})$$

$$\text{subject to} \quad \lambda(r) \leq \frac{1}{1-\alpha} \mathbb{T}_f(r) \text{ a.e. } r \geq 0. \quad (\text{F.11})$$

Now observe that the constraint in the above problem is equivalent to  $\lambda \ll \mathbb{Q}$ . Thus, by defining  $\mathbb{U} = d\lambda/d\mathbb{T}_f$  to be the Radon-Nikodym derivative of  $\lambda$  with respect to  $\mathbb{Q}$ , we can write the problem in the form of (F.3), completing the proof.  $\square$

Succinctly, this proposition shows that provided that  $R$  is sufficiently smooth (i.e., an element of  $L^p$ ), it holds that minimizing the superquantile function is equivalent to solving

$$\min_{f \in \mathcal{F}} \max_{\mathbb{U} \in \mathcal{U}_f(\alpha)} \mathbb{E}_{\mathbb{U}}[R] \quad (\text{F.12})$$

which is a distributionally robust optimization (DRO) problem with uncertainty set  $\mathcal{U}_f(\alpha)$  as defined in (F.4). In plain terms, for any  $\alpha \in (0, 1)$ , this uncertainty set contains probability distributions on  $\mathbb{R}_+$  which can place no larger than  $1/1-\alpha$  on any risk value.

At an intuitive level, this shows that by varying  $\alpha$  in Eq. (F.1), one can interpolate between a range DRO problems. In particular, at level  $\alpha = 1$ , we recover the problem in (3.1), which can be viewed as a DRO problem which selects a Dirac distribution which places solely on the essential supremum of  $R \sim \mathbb{T}_f$ . On the other hand, at level  $\alpha = 0$ , we recover a problem which selects a distribution that equally weights each of the risks in different domains equally. A special case of this is the GroupDRO formulation in [45], wherein under the assumption that the data is partitioned into  $m$  groups, the inner maximum in (F.12) is taken over the  $(m-1)$ -dimensional simplex  $\Delta_m$  (see, e.g., equation (7) in [45]).

## G Additional analyses and experiments

### G.1 Linear regression

In this section we extend § 6.1 to provide further analyses and discussion of EQRM using linear regression datasets based on Ex. A.3. In particular, we: (i) extend Fig. 3 to include plots of the predictors' risk CDFs (G.1.1); and (ii) discuss the ability of EQRM to recover the causal predictor when  $\sigma_1^2$ ,  $\sigma_2^2$  and/or  $\sigma_7^2$  change over environments, compared to IRM [9] and VREx [41] (G.1.2).

Table 5: Recovering the causal predictor for linear regression tasks based on Ex. A.3. A tick means that it is *possible* to recover the causal predictor, under further assumptions.

Changing	Domain Scedasticity	Invariant		IRM	VREx	EQRM
		Risk	Function ( $\beta_{\text{cause}}$ )			
$\sigma_1$	<i>Homoscedastic</i>	✓	✓	✓	✓	✓
$\sigma_2$	<i>Homoscedastic</i>	✓	✓	✓	✓	✓
$\sigma_Y$	<i>Heteroscedastic</i>	✗	✓	✓	✗	✗

### G.1.1 Risk CDFs as risk-robustness curves

As an extension of Fig. 3, in particular the PDFs in Fig. 3 B, Fig. 6 depicts the risk CDFs for different predictors. Here we see that a predictor’s risk CDF depicts its risk-robustness curve, and also that each  $\alpha$  results in a predictor  $f_\alpha$  with minimal  $\alpha$ -quantile risk. That is, for each desired level of robustness (i.e. probability of the upper-bound on risk holding, y-axis), the corresponding  $\alpha$  has minimal risk (x-axis).

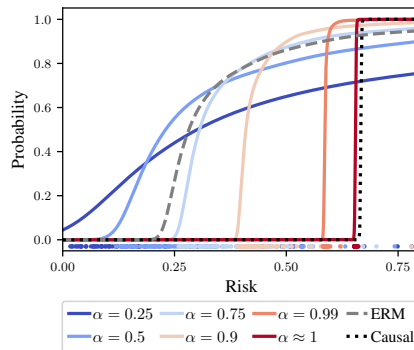


Figure 6: Extension of Fig. 3 showing the risk CDFs (i.e. risk-robustness curves) for different predictors. For each risk upper-bound ( $x$ ), we see the corresponding probability of it holding under the training domains ( $y$ ). Note that, for each level of robustness ( $y$ , i.e. probability that the risk upper-bound holds), the corresponding  $\alpha$  has the lowest upper-bound on risk ( $x$ ). Also note that these CDFs correspond to the PDFs of Fig. 3 (B).

### G.1.2 Invariant risks vs. invariant functions

We now compare seeking invariant *risks* to seeking invariant *functions* by analyzing linear regression datasets, based on Ex. A.3, in which  $\sigma_1^2$ ,  $\sigma_2^2$  and/or  $\sigma_Y^2$  change over domains. This in turn allows us to compare EQRM (invariant risks), VREx [41] (invariant risks), and IRM [9] (invariant functions).

**Domain-skedasticity.** For recovering the causal predictor, the key difference between using invariant *risks* and invariant *functions* lies in the assumption about *domain-skedasticity*, i.e. the “predictability” of  $Y$  across domains. In particular, the causal predictor only has invariant risks in *domain-homoskedastic* cases and not in *domain-heteroskedastic* cases, the latter describing scenarios in which the predictability of  $Y$  (i.e. the amount of irreducible error or intrinsic noise) varies across domains, meaning that the risk of the causal predictor will be smaller on some domains than others. Thus, methods seeking the causal predictor through invariant risks must assume domain homoskedasticity [41, 54]. In contrast, methods seeking the causal predictor through invariant *functions* need not make such a domain-homoskedasticity assumption, but instead the slightly weaker assumption of the conditional mean  $\mathbb{E}[Y|Pa(Y)]$  being invariant across domains. As explained in the next paragraph and summarized in Table 5, this translates into the coefficient  $\beta_{\text{cause}}$  being invariant across domains for the linear SEM of Ex. A.3.

**Mathematical analysis.** We now analyze the risk-invariant solutions of Ex. A.3. We start by expanding the structural equations of Ex. A.3 as:

$$\begin{aligned} X_1 &= N_1, \\ Y &= N_1 + N_Y, \\ X_2 &= N_1 + N_Y + N_2. \end{aligned}$$

We then note that the goal is to learn a model  $\hat{Y} = \beta_1 \cdot X_1 + \beta_2 \cdot X_2$ , which has residual error

$$\begin{aligned} \hat{Y} - Y &= \beta_1 \cdot N_1 + \beta_2 \cdot (N_1 + N_Y + N_2) - N_1 - N_Y \\ &= (\beta_1 + \beta_2 - 1) \cdot N_1 + (\beta_2 - 1) \cdot N_Y + \beta_2 \cdot N_2. \end{aligned}$$

Then, since all variables have zero mean and the noise terms are independent, the risk (i.e. the MSE loss) is simply the variance of the residuals, which can be written as

$$\mathbb{E}[(\hat{Y} - Y)^2] = (\beta_1 + \beta_2 - 1)^2 \cdot \sigma_1^2 + (\beta_2 - 1)^2 \cdot \sigma_Y^2 + \beta_2^2 \cdot \sigma_2^2.$$

Here, we have that, when:

- **Only  $\sigma_1$  changes:** the only way to keep the risk invariant across domains is to set  $\beta_1 + \beta_2 = 1$ . The minimal invariant-risk solution then depends on  $\sigma_Y$  and  $\sigma_2$ :
  - if  $\sigma_Y < \sigma_2$ , the minimal invariant-risk solution sets  $\beta_1 = 1$  and  $\beta_2 = 0$  (causal predictor);
  - if  $\sigma_Y > \sigma_2$ , the minimal invariant-risk solution sets  $\beta_1 = 0$  and  $\beta_2 = 1$  (anti-causal predictor);
  - if  $\sigma_Y = \sigma_2$ , then any solution  $(\beta_1, \beta_2) = (c, 1 - c)$  with  $c \in [0, 1]$  is a minimal invariant-risk solution, including the causal predictor  $c = 1$ , anti-causal predictor  $c = 0$ , and everything in-between.
- **Only  $\sigma_2$  changes:** the invariant-risk solutions set  $\beta_2 = 0$ , with the minimal invariant-risk solution also setting  $\beta_1 = 1$  (causal predictor).
- **$\sigma_1$  and  $\sigma_2$  change:** the invariant-risk solution sets  $\beta_1 = 1, \beta_2 = 0$  (causal predictor).
- **Only  $\sigma_Y$  changes:** the invariant-risk solutions set  $\beta_2 = 1$ , with the minimal invariant-risk solution also setting  $\beta_1 = 0$  (anti-causal predictor).
- **$\sigma_1$  and  $\sigma_Y$  change:** the invariant-risk solution sets  $\beta_1 = 0, \beta_2 = 1$  (anti-causal predictor).
- **$\sigma_2$  and  $\sigma_Y$  change:** there is no invariant-risk solution.
- **$\sigma_1, \sigma_2$  and  $\sigma_Y$  change:** there is no invariant-risk solution.

**Empirical analysis.** To see this empirically, we refer the reader to Table 5 of Krueger et al. [41, App. G.2], which compares the invariant-risk solution of VREx to the invariant-function solution of IRM on the synthetic linear-SEM tasks of Arjovsky et al. [9, Sec. 5.1], which calculate the MSE between the estimated coefficients  $(\hat{\beta}_1, \hat{\beta}_2)$  and those of the causal predictor  $(1, 0)$ .

**Different goals, solutions, and advantages.** We end by emphasizing the fact that the invariant-risk and invariant-function solutions have different pros and cons depending both on the goal and the assumptions made. If the goal is the recover the causal predictor or causes of  $Y$ , then the invariant-function solution has the advantage due to weaker assumptions on domain skedasticity. However, if the goal is learn predictors with stable or invariant performance, such that they perform well on new domains with high probability, then the invariant-risk solution has the advantage. For example, in the domain-heteroskedastic cases above where  $\sigma_Y$  changes or  $\sigma_Y$  and  $\sigma_1$  change, the invariant-function solution recovers the causal predictor  $\beta_1 = 1, \beta_2 = 0$  and thus has arbitrarily-large risk as  $\sigma_Y \rightarrow \infty$  (i.e. in the worst-case). In contrast, the invariant-risk solution recovers the anti-causal predictor  $\beta_1 = 0, \beta_2 = 1$  and thus has fixed risk  $\sigma_2^2$  in all domains.

## G.2 DomainBed

In this section, we include the full per-dataset DomainBed results. We consider the two most common model-selection methods of the DomainBed package—training-domain validation set and test-domain validation set (oracle)—and compare EQRM to a range of baselines. Implementation details for these experiments are provided in § E.3 and our open-source code.

**G.2.1 Model selection: training-domain validation set****VLCS**

Algorithm	C	L	S	V	Avg
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4
MLDG	97.4 ± 0.2	65.2 ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2
CORAL	98.3 ± 0.1	66.1 ± 1.2	73.4 ± 0.3	77.5 ± 1.2	78.8
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.5
DANN	99.0 ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6
CDANN	97.1 ± 0.3	65.1 ± 1.2	70.7 ± 0.8	77.1 ± 1.5	77.5
MTL	97.8 ± 0.4	64.3 ± 0.3	71.5 ± 0.7	75.3 ± 1.7	77.2
SagNet	97.9 ± 0.4	64.5 ± 0.5	71.4 ± 1.3	77.5 ± 0.5	77.8
ARM	98.7 ± 0.2	63.6 ± 0.7	71.3 ± 1.2	76.7 ± 0.6	77.6
VREx	98.4 ± 0.3	64.4 ± 1.4	74.1 ± 0.4	76.2 ± 1.3	78.3
RSC	97.9 ± 0.1	62.5 ± 0.7	72.3 ± 1.2	75.6 ± 0.8	77.1
EQRM	98.3 ± 0.0	63.7 ± 0.8	72.6 ± 1.0	76.7 ± 1.1	77.8

**PACS**

Algorithm	A	C	P	S	Avg
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	86.3
ARM	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1
VREx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
RSC	85.4 ± 0.8	79.7 ± 1.8	97.6 ± 0.3	78.2 ± 1.2	85.2
EQRM	86.5 ± 0.4	82.1 ± 0.7	96.6 ± 0.2	80.8 ± 0.2	86.5

**OfficeHome**

Algorithm	A	C	P	R	Avg
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
Mixup	62.4 ± 0.8	54.8 ± 0.6	76.9 ± 0.3	78.3 ± 0.2	68.1
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
CORAL	65.3 ± 0.4	54.4 ± 0.5	76.5 ± 0.1	78.4 ± 0.5	68.7
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4
SagNet	63.4 ± 0.2	54.8 ± 0.4	75.8 ± 0.4	78.3 ± 0.3	68.1
ARM	58.9 ± 0.8	51.0 ± 0.5	74.1 ± 0.1	75.2 ± 0.3	64.8
VREx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
RSC	60.7 ± 1.4	51.4 ± 0.3	74.8 ± 1.1	75.1 ± 1.3	65.5
EQRM	60.5 ± 0.1	56.0 ± 0.2	76.1 ± 0.4	77.4 ± 0.3	67.5

**TerraIncognita**

Algorithm	L100	L38	L43	L46	Avg
ERM	49.8 ± 4.4	42.1 ± 1.4	56.9 ± 1.8	35.7 ± 3.9	46.1
IRM	54.6 ± 1.3	39.8 ± 1.9	56.2 ± 1.8	39.6 ± 0.8	47.6
GroupDRO	41.2 ± 0.7	38.6 ± 2.1	56.7 ± 0.9	36.4 ± 2.1	43.2
Mixup	59.6 ± 2.0	42.2 ± 1.4	55.9 ± 0.8	33.9 ± 1.4	47.9
MLDG	54.2 ± 3.0	44.3 ± 1.1	55.6 ± 0.3	36.9 ± 2.2	47.7
CORAL	51.6 ± 2.4	42.2 ± 1.0	57.0 ± 1.0	39.8 ± 2.9	47.6
MMD	41.9 ± 3.0	34.8 ± 1.0	57.0 ± 1.9	35.2 ± 1.8	42.2
DANN	51.1 ± 3.5	40.6 ± 0.6	57.4 ± 0.5	37.7 ± 1.8	46.7
CDANN	47.0 ± 1.9	41.3 ± 4.8	54.9 ± 1.7	39.8 ± 2.3	45.8
MTL	49.3 ± 1.2	39.6 ± 6.3	55.6 ± 1.1	37.8 ± 0.8	45.6
SagNet	53.0 ± 2.9	43.0 ± 2.5	57.9 ± 0.6	40.4 ± 1.3	48.6
ARM	49.3 ± 0.7	38.3 ± 2.4	55.8 ± 0.8	38.7 ± 1.3	45.5
VREx	48.2 ± 4.3	41.7 ± 1.3	56.8 ± 0.8	38.7 ± 3.1	46.4
RSC	50.2 ± 2.2	39.2 ± 1.4	56.3 ± 1.4	40.8 ± 0.6	46.6
EQRM	47.9 ± 1.9	45.2 ± 0.3	59.1 ± 0.3	38.8 ± 0.6	47.8

**DomainNet**

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1 ± 0.3	18.8 ± 0.3	46.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9
IRM	48.5 ± 2.8	15.0 ± 1.5	38.3 ± 4.3	10.9 ± 0.5	48.2 ± 5.2	42.3 ± 3.1	33.9
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.3
Mixup	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.3	48.2 ± 0.5	39.2
MLDG	59.1 ± 0.2	19.1 ± 0.3	45.8 ± 0.7	13.4 ± 0.3	59.6 ± 0.2	50.2 ± 0.4	41.2
CORAL	59.2 ± 0.1	19.7 ± 0.2	46.6 ± 0.3	13.4 ± 0.4	59.8 ± 0.2	50.1 ± 0.6	41.5
MMD	32.1 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.8 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3
CDANN	54.6 ± 0.4	17.3 ± 0.1	43.7 ± 0.9	12.1 ± 0.7	56.2 ± 0.4	45.9 ± 0.5	38.3
MTL	57.9 ± 0.5	18.5 ± 0.4	46.0 ± 0.1	12.5 ± 0.1	59.5 ± 0.3	49.2 ± 0.1	40.6
SagNet	57.7 ± 0.3	19.0 ± 0.2	45.3 ± 0.3	12.7 ± 0.5	58.1 ± 0.5	48.8 ± 0.2	40.3
ARM	49.7 ± 0.3	16.3 ± 0.5	40.9 ± 1.1	9.4 ± 0.1	53.4 ± 0.4	43.5 ± 0.4	35.5
VREx	47.3 ± 3.5	16.0 ± 1.5	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.6
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.2 ± 0.2	55.7 ± 0.7	47.8 ± 0.9	38.9
EQRM	56.1 ± 1.3	19.6 ± 0.1	46.3 ± 1.5	12.9 ± 0.3	61.1 ± 0.0	50.3 ± 0.1	41.0

**Averages**

Algorithm	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet	Avg
ERM	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
IRM	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	61.6
GroupDRO	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	60.9
Mixup	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	63.4
MLDG	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	63.6
CORAL	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	64.6
MMD	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	63.3
DANN	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1	62.6
CDANN	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	45.8 ± 1.6	38.3 ± 0.3	62.0
MTL	77.2 ± 0.4	84.6 ± 0.5	66.4 ± 0.5	45.6 ± 1.2	40.6 ± 0.1	62.9
SagNet	77.8 ± 0.5	86.3 ± 0.2	68.1 ± 0.1	48.6 ± 1.0	40.3 ± 0.1	64.2
ARM	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	61.7
VREx	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9	61.9
EQRM	77.8 ± 0.6	86.5 ± 0.2	67.5 ± 0.1	47.8 ± 0.6	41.0 ± 0.3	64.1



**G.2.2 Model selection: test-domain validation set (oracle)****VLCS**

Algorithm	C	L	S	V	Avg
ERM	97.6 ± 0.3	67.9 ± 0.7	70.9 ± 0.2	74.0 ± 0.6	77.6
IRM	97.3 ± 0.2	66.7 ± 0.1	71.0 ± 2.3	72.8 ± 0.4	76.9
GroupDRO	97.7 ± 0.2	65.9 ± 0.2	72.8 ± 0.8	73.4 ± 1.3	77.4
Mixup	97.8 ± 0.4	67.2 ± 0.4	71.5 ± 0.2	75.7 ± 0.6	78.1
MLDG	97.1 ± 0.5	66.6 ± 0.5	71.5 ± 0.1	75.0 ± 0.9	77.5
CORAL	97.3 ± 0.2	67.5 ± 0.6	71.6 ± 0.6	74.5 ± 0.0	77.7
MMD	98.8 ± 0.0	66.4 ± 0.4	70.8 ± 0.5	75.6 ± 0.4	77.9
DANN	99.0 ± 0.2	66.3 ± 1.2	73.4 ± 1.4	80.1 ± 0.5	79.7
CDANN	98.2 ± 0.1	68.8 ± 0.5	74.3 ± 0.6	78.1 ± 0.5	79.9
MTL	97.9 ± 0.7	66.1 ± 0.7	72.0 ± 0.4	74.9 ± 1.1	77.7
SagNet	97.4 ± 0.3	66.4 ± 0.4	71.6 ± 0.1	75.0 ± 0.8	77.6
ARM	97.6 ± 0.6	66.5 ± 0.3	72.7 ± 0.6	74.4 ± 0.7	77.8
VREx	98.4 ± 0.2	66.4 ± 0.7	72.8 ± 0.1	75.0 ± 1.4	78.1
RSC	98.0 ± 0.4	67.2 ± 0.3	70.3 ± 1.3	75.6 ± 0.4	77.8
EQRM	98.2 ± 0.2	66.8 ± 0.8	71.7 ± 1.0	74.6 ± 0.3	77.8

**PACS**

Algorithm	A	C	P	S	Avg
ERM	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	82.7 ± 1.1	86.7
IRM	84.2 ± 0.9	79.7 ± 1.5	95.9 ± 0.4	78.3 ± 2.1	84.5
GroupDRO	87.5 ± 0.5	82.9 ± 0.6	97.1 ± 0.3	81.1 ± 1.2	87.1
Mixup	87.5 ± 0.4	81.6 ± 0.7	97.4 ± 0.2	80.8 ± 0.9	86.8
MLDG	87.0 ± 1.2	82.5 ± 0.9	96.7 ± 0.3	81.2 ± 0.6	86.8
CORAL	86.6 ± 0.8	81.8 ± 0.9	97.1 ± 0.5	82.7 ± 0.6	87.1
MMD	88.1 ± 0.8	82.6 ± 0.7	97.1 ± 0.5	81.2 ± 1.2	87.2
DANN	87.0 ± 0.4	80.3 ± 0.6	96.8 ± 0.3	76.9 ± 1.1	85.2
CDANN	87.7 ± 0.6	80.7 ± 1.2	97.3 ± 0.4	77.6 ± 1.5	85.8
MTL	87.0 ± 0.2	82.7 ± 0.8	96.5 ± 0.7	80.5 ± 0.8	86.7
SagNet	87.4 ± 0.5	81.2 ± 1.2	96.3 ± 0.8	80.7 ± 1.1	86.4
ARM	85.0 ± 1.2	81.4 ± 0.2	95.9 ± 0.3	80.9 ± 0.5	85.8
VREx	87.8 ± 1.2	81.8 ± 0.7	97.4 ± 0.2	82.1 ± 0.7	87.2
RSC	86.0 ± 0.7	81.8 ± 0.9	96.8 ± 0.7	80.4 ± 0.5	86.2
EQRM	88.3 ± 0.6	82.1 ± 0.5	97.2 ± 0.4	81.6 ± 0.5	87.3

**OfficeHome**

Algorithm	A	C	P	R	Avg
ERM	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4
IRM	56.4 ± 3.2	51.2 ± 2.3	71.7 ± 2.7	72.7 ± 2.7	63.0
GroupDRO	60.5 ± 1.6	53.1 ± 0.3	75.5 ± 0.3	75.9 ± 0.7	66.2
Mixup	63.5 ± 0.2	54.6 ± 0.4	76.0 ± 0.3	78.0 ± 0.7	68.0
MLDG	60.5 ± 0.7	54.2 ± 0.5	75.0 ± 0.2	76.7 ± 0.5	66.6
CORAL	64.8 ± 0.8	54.1 ± 0.9	76.5 ± 0.4	78.2 ± 0.4	68.4
MMD	60.4 ± 1.0	53.4 ± 0.5	74.9 ± 0.1	76.1 ± 0.7	66.2
DANN	60.6 ± 1.4	51.8 ± 0.7	73.4 ± 0.5	75.5 ± 0.9	65.3
CDANN	57.9 ± 0.2	52.1 ± 1.2	74.9 ± 0.7	76.2 ± 0.2	65.3
MTL	60.7 ± 0.8	53.5 ± 1.3	75.2 ± 0.6	76.6 ± 0.6	66.5
SagNet	62.7 ± 0.5	53.6 ± 0.5	76.0 ± 0.3	77.8 ± 0.1	67.5
ARM	58.8 ± 0.5	51.8 ± 0.7	74.0 ± 0.1	74.4 ± 0.2	64.8
VREx	59.6 ± 1.0	53.3 ± 0.3	73.2 ± 0.5	76.6 ± 0.4	65.7
RSC	61.7 ± 0.8	53.0 ± 0.9	74.8 ± 0.8	76.3 ± 0.5	66.5
EQRM	60.0 ± 0.8	54.4 ± 0.7	76.5 ± 0.4	77.2 ± 0.5	67.0

**TerraIncognita**

Algorithm	L100	L38	L43	L46	Avg
ERM	59.4 ± 0.9	49.3 ± 0.6	60.1 ± 1.1	43.2 ± 0.5	53.0
IRM	56.5 ± 2.5	49.8 ± 1.5	57.1 ± 2.2	38.6 ± 1.0	50.5
GroupDRO	60.4 ± 1.5	48.3 ± 0.4	58.6 ± 0.8	42.2 ± 0.8	52.4
Mixup	67.6 ± 1.8	51.0 ± 1.3	59.0 ± 0.0	40.0 ± 1.1	54.4
MLDG	59.2 ± 0.1	49.0 ± 0.9	58.4 ± 0.9	41.4 ± 1.0	52.0
CORAL	60.4 ± 0.9	47.2 ± 0.5	59.3 ± 0.4	44.4 ± 0.4	52.8
MMD	60.6 ± 1.1	45.9 ± 0.3	57.8 ± 0.5	43.8 ± 1.2	52.0
DANN	55.2 ± 1.9	47.0 ± 0.7	57.2 ± 0.9	42.9 ± 0.9	50.6
CDANN	56.3 ± 2.0	47.1 ± 0.9	57.2 ± 1.1	42.4 ± 0.8	50.8
MTL	58.4 ± 2.1	48.4 ± 0.8	58.9 ± 0.6	43.0 ± 1.3	52.2
SagNet	56.4 ± 1.9	50.5 ± 2.3	59.1 ± 0.5	44.1 ± 0.6	52.5
ARM	60.1 ± 1.5	48.3 ± 1.6	55.3 ± 0.6	40.9 ± 1.1	51.2
VREx	56.8 ± 1.7	46.5 ± 0.5	58.4 ± 0.3	43.8 ± 0.3	51.4
RSC	59.9 ± 1.4	46.7 ± 0.4	57.8 ± 0.5	44.3 ± 0.6	52.1
EQRM	57.0 ± 1.5	49.5 ± 1.2	59.0 ± 0.3	43.4 ± 0.6	52.2

**DomainNet**

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.6 ± 0.3	19.2 ± 0.2	47.0 ± 0.3	13.2 ± 0.2	59.9 ± 0.3	49.8 ± 0.4	41.3
IRM	40.4 ± 6.6	12.1 ± 2.7	31.4 ± 5.7	9.8 ± 1.2	37.7 ± 9.0	36.7 ± 5.3	28.0
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	34.2 ± 0.3	9.2 ± 0.4	51.9 ± 0.5	40.1 ± 0.6	33.4
Mixup	55.6 ± 0.1	18.7 ± 0.4	45.1 ± 0.5	12.8 ± 0.3	57.6 ± 0.5	48.2 ± 0.4	39.6
MLDG	59.3 ± 0.1	19.6 ± 0.2	46.8 ± 0.2	13.4 ± 0.2	60.1 ± 0.4	50.4 ± 0.3	41.6
CORAL	59.2 ± 0.1	19.9 ± 0.2	47.4 ± 0.2	14.0 ± 0.4	59.8 ± 0.2	50.4 ± 0.4	41.8
MMD	32.2 ± 13.3	11.2 ± 4.5	26.8 ± 11.3	8.8 ± 2.2	32.7 ± 13.8	29.0 ± 11.8	23.5
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.9 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3
CDANN	54.6 ± 0.4	17.3 ± 0.1	44.2 ± 0.7	12.8 ± 0.2	56.2 ± 0.4	45.9 ± 0.5	38.5
MTL	58.0 ± 0.4	19.2 ± 0.2	46.2 ± 0.1	12.7 ± 0.2	59.9 ± 0.1	49.0 ± 0.0	40.8
SagNet	57.7 ± 0.3	19.1 ± 0.1	46.3 ± 0.5	13.5 ± 0.4	58.9 ± 0.4	49.5 ± 0.2	40.8
ARM	49.6 ± 0.4	16.5 ± 0.3	41.5 ± 0.8	10.8 ± 0.1	53.5 ± 0.3	43.9 ± 0.4	36.0
VREx	43.3 ± 4.5	14.1 ± 1.8	32.5 ± 5.0	9.8 ± 1.1	43.5 ± 5.6	37.7 ± 4.5	30.1
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.5 ± 0.1	55.7 ± 0.7	47.8 ± 0.9	38.9
EQRM	55.5 ± 1.8	19.6 ± 0.1	45.9 ± 1.9	12.9 ± 0.3	61.1 ± 0.0	50.3 ± 0.1	40.9

**Averages**

Algorithm	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet	Avg
ERM	77.6 ± 0.3	86.7 ± 0.3	66.4 ± 0.5	53.0 ± 0.3	41.3 ± 0.1	65.0
IRM	76.9 ± 0.6	84.5 ± 1.1	63.0 ± 2.7	50.5 ± 0.7	28.0 ± 5.1	60.6
GroupDRO	77.4 ± 0.5	87.1 ± 0.1	66.2 ± 0.6	52.4 ± 0.1	33.4 ± 0.3	63.3
Mixup	78.1 ± 0.3	86.8 ± 0.3	68.0 ± 0.2	54.4 ± 0.3	39.6 ± 0.1	65.4
MLDG	77.5 ± 0.1	86.8 ± 0.4	66.6 ± 0.3	52.0 ± 0.1	41.6 ± 0.1	64.9
CORAL	77.7 ± 0.2	87.1 ± 0.5	68.4 ± 0.2	52.8 ± 0.2	41.8 ± 0.1	65.6
MMD	77.9 ± 0.1	87.2 ± 0.1	66.2 ± 0.3	52.0 ± 0.4	23.5 ± 9.4	61.4
DANN	79.7 ± 0.5	85.2 ± 0.2	65.3 ± 0.8	50.6 ± 0.4	38.3 ± 0.1	63.8
CDANN	79.9 ± 0.2	85.8 ± 0.8	65.3 ± 0.5	50.8 ± 0.6	38.5 ± 0.2	64.1
MTL	77.7 ± 0.5	86.7 ± 0.2	66.5 ± 0.4	52.2 ± 0.4	40.8 ± 0.1	64.8
SagNet	77.6 ± 0.1	86.4 ± 0.4	67.5 ± 0.2	52.5 ± 0.4	40.8 ± 0.2	65.0
ARM	77.8 ± 0.3	85.8 ± 0.2	64.8 ± 0.4	51.2 ± 0.5	36.0 ± 0.2	63.1
VREx	78.1 ± 0.2	87.2 ± 0.6	65.7 ± 0.3	51.4 ± 0.5	30.1 ± 3.7	62.5
RSC	77.8 ± 0.6	86.2 ± 0.5	66.5 ± 0.6	52.1 ± 0.2	38.9 ± 0.6	64.3
EQRM	77.8 ± 0.2	87.3 ± 0.2	67.0 ± 0.4	52.2 ± 0.7	40.9 ± 0.3	65.1

### G.3 WILDS

In Figure 7, we visualize the test-time risk distributions of IRM and GroupDRO relative to ERM, as well as EQRM $_{\alpha}$  for select values<sup>10</sup> of  $\alpha$ . In each of these figures, we see that IRM and GroupDRO tend to have heavier tails than any of the other algorithms.

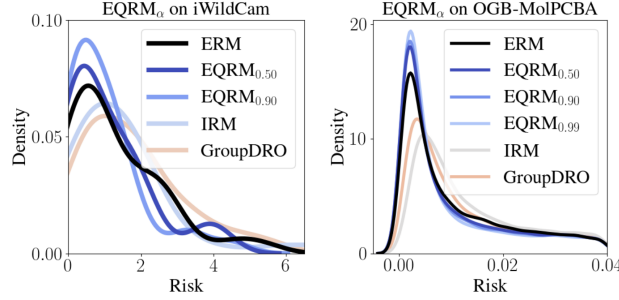


Figure 7: **Baseline test risk distributions on iWildCam and OGB-MolPCBA.** We supplement Figure 4 by providing comparisons to two baseline algorithms: IRM and GroupDRO. In each case, EQRM $_{\alpha}$  tends to display superior tail performance relative to ERM, IRM, and GroupDRO.

**Other performance metrics.** In the main text, we studied the tails of the *risk* distributions of predictors trained on iWildCam and OGB. However, in the broader DG literature, there are a number of other metrics that are used to assess performance or OOD-generalization. In particular, for iWildCam, past work has used the macro  $F_1$  score as well as the average accuracy across domains to assess OOD generalization; for OGB, the standard metric is a predictor’s average precision over test domains [12]. In Tables 6 and 7, we report these metrics and compare the performance of our algorithms to ERM, IRM, and GroupDRO. Below, we discuss the results in each of these tables.

To begin, consider Table 6. Observe that ERM achieves the best *in-distribution* (ID) scores relative to any of the other algorithms. However, when we consider the *out-of-distribution* columns, we see that EQRM offers better performance with respect to both the macro  $F_1$  score and the mean accuracy. Thus, although our algorithms are not explicitly trained to optimize these metrics, their strong performance on the tails of the risk distribution appears to be correlated with strong OOD performance with these alternative metrics. We also observe that relative to ERM, EQRM suffers smaller accuracy drops between ID and OOD mean accuracy. Specifically, ERM dropped 5.50 points, whereas EQRM dropped by an average of 2.38 points.

Next, consider Table 7. Observe again that ERM is the strongest-performing *baseline* (first band of the table). Also observe that EQRM performs similarly to ERM, with validation and test precision tending to cluster around 28 and 27 respectively. However, we stress that these metrics are *averaged* over their respective domains, whereas in Tables 2 and 3, we showed that EQRM performed well on the more difficult domains, i.e. when using *tail* metrics.

Table 6: WILDS metrics on iWildCam.

Algorithm	Macro $F_1$ ( $\uparrow$ )		Mean accuracy ( $\uparrow$ )	
	ID	OOD	ID	OOD
ERM	<b>49.8</b>	30.6	<b>77.0</b>	71.5
IRM	23.4	15.2	59.6	64.1
GroupDRO	34.3	22.1	66.7	67.7
QRM $_{0.25}$	18.3	11.4	54.3	58.3
QRM $_{0.50}$	48.1	33.8	76.2	73.5
QRM $_{0.75}$	49.5	31.8	76.1	72.0
QRM $_{0.90}$	48.6	32.9	77.1	73.3
QRM $_{0.99}$	45.9	30.8	76.6	71.3

Table 7: WILDS metrics on OGB-MolPCBA.

Algorithm	Mean precision ( $\uparrow$ )	
	Validation	Test
ERM	28.1	27.3
IRM	15.4	15.5
GroupDRO	23.5	22.3
QRM $_{0.25}$	28.1	27.3
QRM $_{0.50}$	<b>28.3</b>	<b>27.4</b>
QRM $_{0.75}$	28.1	27.1
QRM $_{0.90}$	27.9	27.2
QRM $_{0.99}$	28.1	27.4

<sup>10</sup>We display results for fewer values of  $\alpha$  in Figure 7 to keep the plots uncluttered.

## H Limitations of our work

As discussed in the first paragraph of § 7, the main limitation of our work is that, for  $\alpha$  to *precisely* approximate the probability of generalizing with risk below the associated  $\alpha$ -quantile value, we must have a large number of i.i.d.-sampled domains. Currently, this is rarely satisfied in practice, although § 7 describes how new data-collection procedures could help to better-satisfy this assumption. We believe that our work, and its promise of machine learning systems that generalize with high probability, provides sufficient motivation for collecting real-world datasets with a large number of i.i.d.-sampled domains. In addition, we hope that future work can explore ways to relax this assumption, e.g., by leveraging knowledge of domain dependencies like time.

### **A.3 Domain Generalisation: Harnessing Spurious Features (§ 5.2)**

# Appendices

## Table of Contents

---

<b>A Proof and Further Discussion of Theorem 4.4</b>	<b>17</b>
A.1 Proof of Theorem 4.4 . . . . .	17
A.2 Further discussion of Theorem 4.4 . . . . .	19
<b>B Proof of Theorem 4.6</b>	<b>20</b>
<b>C Multiclass Case</b>	<b>22</b>
<b>D Supplementary Results</b>	<b>24</b>
D.1 Trivial solution to joint-risk minimization . . . . .	24
D.2 Causal perspectives . . . . .	25
<b>E Datasets</b>	<b>26</b>
<b>F Further Experiments</b>	<b>27</b>
F.1 ColorMNIST . . . . .	28
F.2 Camelyon17 . . . . .	29
<b>G Implementation Details</b>	<b>30</b>
G.1 Adaptive baselines . . . . .	30
G.2 Synthetic experiments . . . . .	31
G.3 ColorMNIST experiments . . . . .	31
G.4 PACS experiments . . . . .	31
G.5 Camelyon17 experiments . . . . .	32
<b>H Further Related Work</b>	<b>32</b>
<b>I Performance When Complementarity is Violated</b>	<b>33</b>

---

## A Proof and Further Discussion of Theorem 4.4

### A.1 Proof of Theorem 4.4

In this section, we prove our main results regarding the marginal generalization problem presented in Section 4, namely Thm. 4.4. For the reader's convenience, we restate Thm. 4.4 here:

**Theorem 4.4** (Marginal generalization with for binary labels and complementary features). *Consider three random variables  $X_S$ ,  $X_U$ , and  $Y$ , where*

1.  $Y$  is binary ( $\{0, 1\}$ -valued),
2.  $X_S$  and  $X_U$  are complementary features for  $Y$  (i.e.,  $X_S \perp\!\!\!\perp X_U | Y$ ), and
3.  $X_S$  is informative of  $Y$  ( $X_S \not\perp\!\!\!\perp Y$ ).

Then, the joint distribution of  $(X_S, X_U, Y)$  can be written in terms of the joint distributions of  $(X_S, Y)$  and  $(X_S, X_U)$ . Specifically, if  $\hat{Y}|X_S \sim \text{Bernoulli}(\Pr[Y = 1|X_S])$  is pseudo-label and

$$\epsilon_0 := \Pr[\hat{Y} = 0|Y = 0] \quad \text{and} \quad \epsilon_1 := \Pr[\hat{Y} = 1|Y = 1] \quad (\text{A.1})$$

are the conditional probabilities that  $\hat{Y}$  and  $Y$  agree, given  $Y = 0$  and  $Y = 1$ , respectively, then,

1.  $\epsilon_0 + \epsilon_1 > 1$ ,
2.  $\Pr[Y = 1|X_U] = \frac{\Pr[\hat{Y} = 1|X_U] + \epsilon_0 - 1}{\epsilon_0 + \epsilon_1 - 1}$ , and
3.  $\Pr[Y = 1|X_S, X_U] = \sigma(\text{logit}(\Pr[Y = 1|X_S]) + \text{logit}(\Pr[Y = 1|X_U]) - \text{logit}(\Pr[Y = 1]))$ .

Before proving Thm. 4.4, we provide some examples demonstrating that the complementarity and informativeness assumptions in Thm. 4.4 cannot be dropped.

**Example A.1.** Suppose  $X_S$  and  $X_U$  have independent Bernoulli(1/2) distributions. Then,  $X_S$  is informative of both of the binary variables  $Y_1 = X_S X_U$  and  $Y_2 = X_S(1 - X_U)$  and both have identical conditional distributions given  $X_S$ , but  $Y_1$  and  $Y_2$  have different conditional distributions given  $X_U$ :

$$\Pr[Y_1 = 1|X_U = 0] = 0 \neq 1/2 = \Pr[Y_2 = 1|X_U = 0].$$

Thus, the complementarity condition cannot be omitted.

On the other hand,  $X_S$  and  $X_U$  are complementary for both  $Y_3 = X_U$  and an independent  $Y_4 \sim \text{Bernoulli}(1/2)$  and both  $Y_3$  and  $Y_4$  both have identical conditional distributions given  $X_S$ , but  $Y_1$  and  $Y_2$  have different conditional distributions given  $X_U$ :

$$\Pr[Y_3 = 1|X_U = 1] = 1/2 \neq 1 = \Pr[Y_4 = 1|X_U = 1].$$

Thus, the informativeness condition cannot be omitted.

Before proving Thm. 4.4, we prove Lemma 4.5, which allows us to safely divide by the quantity  $\epsilon_0 + \epsilon_1 - 1$  in the formula for  $\Pr[Y = 1|X_U]$ , under the condition that  $X_S$  is informative of  $Y$ .

**Lemma 4.5.** *In the setting of Thm. 4.4, let  $\epsilon_0$  and  $\epsilon_1$  be the class-wise pseudo-label accuracies defined in as in Eq. (A.1). Then,  $\epsilon_0 + \epsilon_1 = 1$  if and only if  $X_S$  and  $Y$  are independent.*

Note that the entire result also holds, with almost identical proof, in the multi-environment setting of Sections 3 and 5, conditioned on a particular environment  $E$ .

*Proof.* We first prove the forward implication. Suppose  $\epsilon_0 + \epsilon_1 = 1$ . If  $\Pr[Y = 1] \in \{0, 1\}$ , then  $X_S$  and  $Y$  are trivially independent, so we may assume  $\Pr[Y = 1] \in (0, 1)$ . Then,

$$\begin{aligned} \mathbb{E}[\hat{Y}] &= \epsilon_1 \Pr[Y = 1] + (1 - \epsilon_0)(1 - \Pr[Y = 1]) && \text{(Law of Total Expectation)} \\ &= (\epsilon_0 + \epsilon_1 - 1) \Pr[Y = 1] + 1 - \epsilon_0 \\ &= 1 - \epsilon_0 && (\epsilon_0 + \epsilon_1 = 1) \\ &= \mathbb{E}[\hat{Y}|Y = 0]. && \text{(Definition of } \epsilon_0) \end{aligned}$$

Since  $Y$  is binary and  $\Pr[Y = 1] \in (0, 1)$ , it follows that  $\mathbb{E}[\hat{Y}] = \mathbb{E}[\hat{Y}|Y = 0] = \mathbb{E}[\hat{Y}|Y = 1]$ ; i.e.,  $\mathbb{E}[\hat{Y}|Y] \perp\!\!\!\perp Y$ . Since  $\hat{Y}$  is binary, its distribution is specified entirely by its mean, and so  $\hat{Y} \perp\!\!\!\perp Y$ . It follows that the covariance between  $\hat{Y}$  and  $Y$  is 0:

$$\begin{aligned} 0 &= \mathbb{E}[(Y - \mathbb{E}[Y])(\hat{Y} - \mathbb{E}[\hat{Y}])] \\ &= \mathbb{E}[\mathbb{E}[(Y - \mathbb{E}[Y])(\hat{Y} - \mathbb{E}[\hat{Y}])|X_S]] && \text{(Law of Total Expectation)} \\ &= \mathbb{E}[\mathbb{E}[Y - \mathbb{E}[Y]|X_S] \mathbb{E}[\hat{Y} - \mathbb{E}[\hat{Y}]|X_S]] && (Y \perp\!\!\!\perp \hat{Y}|X_S) \\ &= \mathbb{E}[(\mathbb{E}[Y - \mathbb{E}[Y]|X_S])^2], \end{aligned}$$

where the final equality holds because  $\hat{Y}$  and  $Y$  have identical conditional distributions given  $X_S$ . Since the  $\mathcal{L}_2$  norm of a random variable is 0 if and only if the variable is 0 almost surely, it follows that,  $P_{X_S}$ -almost surely,

$$0 = \mathbb{E}[Y - \mathbb{E}[Y]|X_S] = \mathbb{E}[Y|X_S] - \mathbb{E}[Y],$$

so that  $\mathbb{E}[Y|X_S] \perp\!\!\!\perp X_S$ . Since  $Y$  is binary, its distribution is specified entirely by its mean, and so  $Y \perp\!\!\!\perp X_S$ , proving the forward implication.

To prove the reverse implication, suppose  $X_S$  and  $Y$  are independent. Then  $\hat{Y}$  and  $Y$  are also independent. Hence,

$$\epsilon_1 = \mathbb{E}[\hat{Y}|Y = 1] = \mathbb{E}[\hat{Y}|Y = 0] = 1 - \epsilon_0,$$

so that  $\epsilon_0 + \epsilon_1 = 1$ .  $\square$

We now use Lemma 4.5 to prove Thm. 4.4:

*Proof.* To begin, note that  $\hat{Y}$  has the same conditional distribution given  $X_S$  as  $Y$  (i.e.,  $P_{\hat{Y}|X_S} = P_{Y|X_S}$ ) and that  $\hat{Y}$  is conditionally independent of  $Y$  given  $X_S$  ( $\hat{Y} \perp\!\!\!\perp Y|X_S$ ). Then, since

$$\Pr[\hat{Y} = 1] = \mathbb{E}[\Pr[Y = 1|X_S]] = \Pr[Y = 1], \quad (\text{A.2})$$

we have

$$\begin{aligned} \epsilon_1 = \Pr[\hat{Y} = 1|Y = 1] &= \frac{\Pr[Y = 1, \hat{Y} = 1]}{\Pr[Y = 1]} && \text{(Definition of } \epsilon_1) \\ &= \frac{\Pr[Y = 1, \hat{Y} = 1]}{\Pr[\hat{Y} = 1]} && \text{(Eq. (A.2))} \\ &= \frac{\mathbb{E}_{X_S}[\Pr[Y = 1, \hat{Y} = 1|X_S]]}{\mathbb{E}_{X_S}[\Pr[\hat{Y} = 1|X_S]]} && \text{(Law of Total Expectation)} \\ &= \frac{\mathbb{E}_{X_S}[\Pr[Y = 1|X_S] \Pr[\hat{Y} = 1|X_S]]}{\mathbb{E}_{X_S}[\Pr[\hat{Y} = 1|X_S]]} && (\hat{Y} \perp\!\!\!\perp Y|X_S) \\ &= \frac{\mathbb{E}_{X_S}[(\Pr[Y = 1|X_S])^2]}{\mathbb{E}_{X_S}[\Pr[Y = 1|X_S]]} && (P_{\hat{Y}|X_S} = P_{Y|X_S}) \end{aligned}$$

entirely in terms of the conditional distribution  $P_{Y|X_S}$  and the marginal distribution  $P_{X_S}$ . Similarly,

$\epsilon_0$  can be written as  $\epsilon_0 = \frac{\mathbb{E}_{X_S}[(\Pr[Y=0|X_S])^2]}{\mathbb{E}_{X_S}[\Pr[Y=0|X_S]]}$ . Meanwhile, by the law of total expectation, and the

assumption that  $X_S$  (and hence  $\hat{Y}$ ) is conditionally independent of  $X_U$  given  $Y$ , the conditional distribution  $P_{\hat{Y}|X_U}$  of  $\hat{Y}$  given  $X_U$  can be written as

$$\begin{aligned} &\Pr[\hat{Y} = 1|X_U] \\ &= \Pr[\hat{Y} = 1|Y = 0, X_U] \Pr[Y = 0|X_U] + \Pr[\hat{Y} = 1|Y = 1, X_U] \Pr[Y = 1|X_U] \\ &= \Pr[\hat{Y} = 1|Y = 0] \Pr[Y = 0|X_U] + \Pr[\hat{Y} = 1|Y = 1] \Pr[Y = 1|X_U] \\ &= (1 - \epsilon_0)(1 - \Pr[Y = 1|X_U]) + \epsilon_1 \Pr[Y = 1|X_U] \\ &= (\epsilon_0 + \epsilon_1 - 1) \Pr[Y = 1|X_U] + 1 - \epsilon_0. \end{aligned}$$



By Lemma 4.5, the assumption  $X_S \not\perp\!\!\!\perp Y$  implies  $\epsilon_0 + \epsilon_1 \neq 1$ . Hence, re-arranging the above equality gives us the conditional distribution  $P_{Y|X_U}$  of  $Y$  given  $X_U$  purely in terms of the conditional  $P_{Y|X_S}$  and  $P_{X_S, X_U}$ :

$$\Pr[Y = 1|X_U = X_U] = \frac{\Pr[\hat{Y} = 1|X_U = X_U] + \epsilon_0 - 1}{\epsilon_0 + \epsilon_1 - 1}.$$

It remains now to write the conditional distribution  $P_{Y|X_S, X_U}$  in terms of the conditional distributions  $P_{Y|X_S}$  and  $P_{Y|X_U}$  and the marginal  $P_Y$ . Note that

$$\begin{aligned} \frac{\Pr[Y = 1|X_S, X_U]}{\Pr[Y = 0|X_S, X_U]} &= \frac{\Pr[X_S, X_U|Y = 1] \Pr[Y = 1]}{\Pr[X_S, X_U|Y = 0] \Pr[Y = 0]} && \text{(Bayes' Rule)} \\ &= \frac{\Pr[X_S|Y = 1] \Pr[X_U|Y = 1] \Pr[Y = 1]}{\Pr[X_S|Y = 0] \Pr[X_U|Y = 0] \Pr[Y = 0]} && \text{(Complementarity)} \\ &= \frac{\Pr[Y = 1|X_S] \Pr[Y = 1|X_U] \Pr[Y = 0]}{\Pr[Y = 0|X_S] \Pr[Y = 0|X_U] \Pr[Y = 1]}. && \text{(Bayes' Rule)} \end{aligned}$$

It follows that the logit of  $\Pr[Y = 1|X_S, X_U]$  can be written as the sum of a term depending only on  $X_S$ , a term depending only on  $X_U$ , and a constant term:

$$\begin{aligned} \text{logit}(\Pr[Y = 1|X_S, X_U]) &= \log \frac{\Pr[Y = 1|X_S, X_U]}{1 - \Pr[Y = 1|X_S, X_U]} \\ &= \log \frac{\Pr[Y = 1|X_S, X_U]}{\Pr[Y = 0|X_S, X_U]} \\ &= \log \frac{\Pr[Y = 1|X_S]}{\Pr[Y = 0|X_S]} + \log \frac{\Pr[Y = 1|X_U]}{\Pr[Y = 0|X_U]} - \log \frac{\Pr[Y = 1]}{\Pr[Y = 0]} \\ &= \text{logit}(\Pr[Y = 1|X_S]) + \text{logit}(\Pr[Y = 1|X_U]) - \text{logit}(\Pr[Y = 1]). \end{aligned}$$

Since the sigmoid  $\sigma$  is the inverse of logit,

$$\Pr[Y = 1|X_S, X_U] = \sigma(\text{logit}(\Pr[Y = 1|X_S]) + \text{logit}(\Pr[Y = 1|X_U]) - \text{logit}(\Pr[Y = 1])),$$

which, by Eq. (4.3), can be written in terms of the conditional distribution  $P_{Y|X_S}$  and the joint distribution  $P_{X_S, X_U}$ .  $\square$

## A.2 Further discussion of Theorem 4.4

**Connections to learning from noisy labels.** Thm. 4.4 leverages two theoretical insights about the special structure of pseudo-labels that complement results in the literature on learning from noisy labels. First, Blanchard et al. [7] showed that learning from noisy labels is possible if and only if the total noise level is below the critical threshold  $\epsilon_0 + \epsilon_1 > 1$ ; in the case of learning from pseudo-labels, we show (see Lemma 4.5 in Appendix A.1) that this is satisfied if and only if  $X_S$  is informative of  $Y$  (i.e.,  $Y \not\perp\!\!\!\perp X_S$ ). Second, methods for learning under label noise commonly assume knowledge of  $\epsilon_0$  and  $\epsilon_1$  [44, 75], which may be unrealistic in applications where we have absolutely no true (i.e., test-domain) labels; however, for pseudo-labels sampled from a known conditional probability distribution  $P_{Y|X_S}$ , one can express these noise levels in terms of  $P_{Y|X_S}$  and  $P_{X_S}$  and thereby estimate them *without any true labels*, as on line 3 of Alg. 1.

**Possible applications of Thm. 4.4 beyond domain adaptation** The reason we wrote Thm. 4.4 in the more general setting of the marginal problem rather than in the specific context of domain adaptation is that we envision possible applications to a number of problems besides domain adaptation. For example, suppose that, after learning a calibrated machine learning model  $M_1$  using a feature  $X_S$ , we observe an additional feature  $X_U$ . In the case that  $X_S$  and  $X_U$  are complementary, Thm. 4.4 justifies using the student-teacher paradigm [11, 2, 27] to train a model for predicting  $Y$  from  $X_U$  (or from  $(X_S, X_U)$  jointly) based on predictions from  $M_1$ . This could be useful if we don't have access to labeled pairs  $(X_U, Y)$ , or if retraining a model using  $X_S$  would require substantial computational resources or access to sensitive or private data. Exploring such approaches could be a fruitful direction for future work

## B Proof of Theorem 4.6

This appendix provides a proof of Thm. 4.6, which provides conditions under which our proposed domain adaptation procedure (Alg. 1) is consistent.

We state a formal version of Thm. 4.6:

**Theorem 4.6** (Consistency of the bias-corrected classifier). *Assume*

1.  $X_S$  is stable,
2.  $X_S$  and  $X_U$  are complementary, and
3.  $X_S$  is informative of  $Y$  (i.e.,  $X_S \not\perp\!\!\!\perp Y$ ).

Let  $\hat{\eta}_n : \mathcal{X}_S \times X_U \rightarrow [0, 1]$  given by

$$\hat{\eta}_n(x_S, x_U) = \sigma \left( f_S(x_S) + \text{logit} \left( \frac{\hat{\eta}_{U,n}(x_U) + \hat{\epsilon}_{0,n} - 1}{\hat{\epsilon}_{0,n} + \hat{\epsilon}_{1,n} - 1} \right) - \beta_1 \right), \quad \text{for all } (x_S, x_U) \in \mathcal{X}_S \times \mathcal{X}_U,$$

denote the bias-corrected regression function estimate proposed in Alg. 1, and let  $\hat{h}_n : \mathcal{X}_S \times \mathcal{X}_U \rightarrow \{0, 1\}$  given by

$$\hat{h}_n(x_S, x_U) = 1\{\hat{\eta}_n(x_S, x_U) > 1/2\}, \quad \text{for all } (x_S, x_U) \in \mathcal{X}_S \times \mathcal{X}_U,$$

denote the corresponding hard classifier. Let  $\eta_U : \mathcal{X}_U \rightarrow [0, 1]$ , given by  $\eta_U(x_U) = \Pr[Y = 1 | X_U = x_U, E = 1]$  for all  $x_U \in \mathcal{X}_U$ , denote the true regression function over  $X_U$ , and let  $\hat{\eta}_{U,n}$  denote its estimate as assumed in Line 4 of Alg. 1. Then, as  $n \rightarrow \infty$ ,

- (a) if, for  $P_{X_U}$ -almost all  $x_U \in \mathcal{X}_U$ ,  $\hat{\eta}_{U,n}(x_U) \rightarrow \eta_U(x_U)$  in probability, then  $\hat{\eta}_n$  and  $\hat{h}_n$  are weakly consistent (i.e.,  $\hat{\eta}_n(x_S, x_U) \rightarrow \eta(x_S, x_U)$   $P_{X_S, X_U}$ -almost surely and  $R(\hat{h}_n) \rightarrow R(h^*)$  in probability).
- (b) if, for  $P_{X_U}$ -almost all  $x_U \in \mathcal{X}_U$ ,  $\hat{\eta}_{U,n}(x_U) \rightarrow \eta_U(x_U)$  almost surely, then  $\hat{\eta}_n$  and  $\hat{h}_n$  are strongly consistent (i.e.,  $\hat{\eta}_n(x_S, x_U) \rightarrow \eta(x_S, x_U)$   $P_{X_S, X_U}$ -almost surely and  $R(\hat{h}_n) \rightarrow R(h^*)$  a.s.).

Before proving Thm. 4.6, we provide a few technical lemmas. The first shows that almost-everywhere convergence of regression functions implies convergence of the corresponding classifiers in classification risk:

**Lemma B.1.** *Consider a sequence of regression functions  $\eta, \eta_1, \eta_2, \dots : \mathcal{X} \rightarrow [0, 1]$ . Let  $h, h_1, h_2, \dots : \mathcal{X} \rightarrow \{0, 1\}$  denote the corresponding classifiers*

$$h(x) = 1\{\eta(x) > 1/2\} \quad \text{and} \quad h_i(x) = 1\{\eta_i(x) > 1/2\}, \quad \text{for all } i \in \mathbb{N}, x \in \mathcal{X}.$$

- (a) *If  $\eta_n(x) \rightarrow \eta(x)$  for  $P_X$ -almost all  $x \in \mathcal{X}$  in probability, then  $R(h_n) \rightarrow R(h^*)$  in probability.*
- (b) *If  $\eta_n(x) \rightarrow \eta(x)$  for  $P_X$ -almost all  $x \in \mathcal{X}$  almost surely as  $n \rightarrow \infty$ , then  $R(h_n) \rightarrow R(h)$  almost surely.*

*Proof.* Note that, since  $h_n(x) \neq h(x)$  implies  $|\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2|$ ,

$$1\{h_n(x) \neq h(x)\} \leq 1\{|\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2|\}. \quad (\text{B.1})$$

We utilize this observation to prove both (a) and (b).

**Proof of (a)** Let  $\delta > 0$ . By Inequality (B.1) and partitioning  $\mathcal{X}$  based on whether  $|2\eta(X) - 1| \leq \delta/2$ ,

$$\begin{aligned} & \mathbb{E}_X [ |2\eta(X) - 1| 1\{h_n(X) \neq h(X)\} ] \\ & \leq \mathbb{E}_X [ |2\eta(X) - 1| 1\{|\eta_n(X) - \eta(X)| \geq |\eta(X) - 1/2|\} ] \\ & = \mathbb{E}_X [ |2\eta(X) - 1| 1\{|\eta_n(X) - \eta(X)| \geq |\eta(X) - 1/2|\} 1\{|2\eta(X) - 1| > \delta/2\} ] \\ & \quad + \mathbb{E}_X [ |2\eta(X) - 1| 1\{|\eta_n(X) - \eta(X)| \geq |\eta(X) - 1/2|\} 1\{|2\eta(X) - 1| \leq \delta/2\} ] \\ & \leq \mathbb{E}_X [ 1\{|\eta_n(X) - \eta(X)| > \delta/2\} ] + \delta/2. \end{aligned}$$

Hence,

$$\begin{aligned}
& \lim_{n \rightarrow \infty} \Pr_{\eta_n} [\mathbb{E}_X [|2\eta(X) - 1| 1\{h_n(X) \neq h(X)\}] > \delta] \\
& \leq \lim_{n \rightarrow \infty} \Pr_{\eta_n} [\mathbb{E}_X [1\{|\eta_n(X) - \eta(X)| > \delta/2\}] > \delta/2] \\
& \leq \lim_{n \rightarrow \infty} \frac{2}{\delta} \mathbb{E}_{\eta_n} [\mathbb{E}_X [1\{|\eta_n(X) - \eta(X)| > \delta/2\}]] \quad (\text{Markov's Inequality}) \\
& = \lim_{n \rightarrow \infty} \frac{2}{\delta} \mathbb{E}_X [\mathbb{E}_{\eta_n} [1\{|\eta_n(X) - \eta(X)| > \delta/2\}]] \quad (\text{Fubini's Theorem}) \\
& = \frac{2}{\delta} \mathbb{E}_X \left[ \lim_{n \rightarrow \infty} \Pr_{\eta_n} [|\eta_n(X) - \eta(X)| > \delta/2] \right] \quad (\text{Dominated Convergence Theorem}) \\
& = 0. \quad (\eta_n(X) \rightarrow \eta(X), P_X\text{-a.s., in probability})
\end{aligned}$$

**Proof of (b)** For any  $x \in \mathcal{X}$  with  $\eta(x) \neq 1/2$ , if  $\eta_n(x) \rightarrow \eta(x)$  then  $1\{|\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2|\} \rightarrow 0$ . Hence, by Inequality (B.1), the dominated convergence theorem (with  $|2\eta(x) - 1| 1\{|\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2|\} \leq 1$ ), and the assumption that  $\eta_n(x) \rightarrow \eta(x)$  for  $P_X$ -almost all  $x \in \mathcal{X}$  almost surely,

$$\begin{aligned}
& \lim_{n \rightarrow \infty} \mathbb{E}_X [|2\eta(X) - 1| 1\{h_n(X) \neq h(X)\}] \\
& \leq \lim_{n \rightarrow \infty} \mathbb{E}_X [|2\eta(X) - 1| 1\{|\eta_n(X) - \eta(X)| \geq |\eta(X) - 1/2|\}] \\
& = \mathbb{E}_X \left[ \lim_{n \rightarrow \infty} |2\eta(X) - 1| 1\{|\eta_n(X) - \eta(X)| \geq |\eta(X) - 1/2|\} \right] \\
& = 0, \quad \text{almost surely.}
\end{aligned}$$

□

Our next lemma concerns an edge case in which the features  $X_S$  and  $X_U$  provide perfect but contradictory information about  $Y$ , leading to Equation (4.4) being ill-defined. We show that this can happen only with probability 0 over  $(X_S, X_U) \sim P_{X_S, X_U}$  can thus be safely ignored:

**Lemma B.2.** Consider two predictors  $X_S$  and  $X_U$  of a binary label  $Y$ . Then,

$$\Pr_{X_S, X_U} [\mathbb{E}[Y|X_S] = 1 \text{ and } \mathbb{E}[Y|X_U] = 0] = \Pr_{X_S, X_U} [\mathbb{E}[Y|X_S] = 0 \text{ and } \mathbb{E}[Y|X_U] = 1] = 0.$$

*Proof.* Suppose, for sake of contradiction, that the event

$$A := \{(x_S, x_U) : \mathbb{E}[Y|X_S = x_S] = 1 \text{ and } \mathbb{E}[Y|X_U = x_U] = 0\}$$

has positive probability. Then, the conditional expectation  $\mathbb{E}[Y|A]$  is well-defined, giving the contradiction

$$1 = \mathbb{E}_{X_S} [\mathbb{E}[Y|E, X_S]] = \mathbb{E}[Y|A] = \mathbb{E}_{X_U} [\mathbb{E}[Y|E, X_U]] = 0.$$

The case  $\mathbb{E}[Y|X_S] = 0$  and  $\mathbb{E}[Y|X_U] = 1$  is similar. □

We now utilize Lemmas B.1 and B.2 to prove Thm. 4.6.

*Proof.* By Lemma B.1, it suffices to prove that  $\hat{\eta}(x_S, x_U) \rightarrow \eta(x_S, x_U)$ , for  $P_{X_S, X_U}$ -almost all  $(x_S, x_U) \in \mathcal{X}_S \times \mathcal{X}_U$ , in probability (to prove (a)) and almost surely (to prove (b)).

**Finite case** We first consider the case when both  $\Pr[Y|X_S = x_S], \Pr[Y|X_U = x_U] \in (0, 1)$ , so that  $f_S(x_S)$  and  $\text{logit} \left( \frac{\tilde{\eta}(x_U) + \epsilon_0 - 1}{\epsilon_0 + \epsilon_1 - 1} \right)$  are both finite. Since

$$\begin{aligned}
& \hat{\eta}_{S,U}(x_S, x_U) - \eta_{S,U}(x_S, x_U) \\
& = \sigma \left( f_S(x_S) + \text{logit} \left( \frac{\hat{\eta}_{U,1}(x_U) + \hat{\epsilon}_0 - 1}{\hat{\epsilon}_0 + \hat{\epsilon}_1 - 1} \right) - \hat{\beta}_{1,n} \right) - \sigma \left( f_S(x_S) + \text{logit} \left( \frac{\tilde{\eta}(x_U) + \epsilon_0 - 1}{\epsilon_0 + \epsilon_1 - 1} \right) - \beta_1 \right),
\end{aligned}$$

where the sigmoid  $\sigma : \mathbb{R} \rightarrow [0, 1]$  is continuous, by the continuous mapping theorem and the assumption that  $\hat{\eta}_{U,1}(x_U) \rightarrow \tilde{\eta}(x_U)$ , to prove both of these, it suffices to show:

- (i)  $\widehat{\epsilon}_0 \rightarrow \epsilon_0$  and  $\widehat{\epsilon}_1 \rightarrow \epsilon_1$  almost surely as  $n \rightarrow \infty$ .
- (ii)  $\widehat{\beta}_{1,n} \rightarrow \beta_1 \in (-\infty, \infty)$  almost surely as  $n \rightarrow \infty$ .
- (iii) The mapping  $(a, b, c) \mapsto \text{logit}\left(\frac{a+b-1}{b+c-1}\right)$  is continuous at  $(\widetilde{\eta}(x_U), \epsilon_0, \epsilon_1)$ .

We now prove each of these in turn.

**Proof of (i)** Since  $\widehat{Y}_i \perp\!\!\!\perp Y_i | X_S$  and  $0 < \Pr[\widehat{Y} = 1]$ , by the strong law of large numbers and the continuous mapping theorem,

$$\widehat{\epsilon}_1 = \frac{1}{n_1} \sum_{i=1}^n \widehat{Y}_i \sigma(f_S(X_i)) = \frac{\frac{1}{n} \sum_{i=1}^n \widehat{Y}_i \sigma(f_S(X_i))}{\frac{1}{n} \sum_{i=1}^n \widehat{Y}_i} \rightarrow \frac{\mathbb{E}[\sigma(f_S(X)) \mathbf{1}\{\widehat{Y} = 1\}]}{\Pr[\widehat{Y} = 1]} = \mathbb{E}[\sigma(f_S(X)) | \widehat{Y} = 1] = \epsilon_1,$$

almost surely as  $n \rightarrow \infty$ . Similarly, since  $\Pr[\widehat{Y} = 0] = 1 - \Pr[\widehat{Y} = 1] > 0$ ,  $\widehat{\epsilon}_0 \rightarrow \epsilon_0$  almost surely.

**Proof of (ii)** Recall that

$$\widehat{\beta}_{1,n} = \text{logit}\left(\frac{1}{n} \sum_{i=1}^n \widehat{Y}_i\right).$$

By the strong law of large numbers,  $\frac{1}{n} \sum_{i=1}^n \widehat{Y}_i \rightarrow \Pr[\widehat{Y} = 1 | E = 1] = \Pr[Y = 1 | E = 1]$ . Since we assumed  $\Pr[Y = 1 | E = 1] \in (0, 1)$ , it follows that the mapping  $a \mapsto \text{logit}(a)$  is continuous at  $a = \Pr[Y = 1 | E = 1]$ . Hence, by the continuous mapping theorem,  $\widehat{\beta}_{1,n} \rightarrow \text{logit}(\Pr[Y = 1 | E = 1]) = \beta_1$  almost surely.

**Proof of (iii)** Since the logit function is continuous on the open interval  $(0, 1)$  and we assumed  $\epsilon_0 + \epsilon_1 > 1$ , it suffices to show that  $0 < \widetilde{\eta}(x_U) + \epsilon_0 - 1 < \epsilon_0 + \epsilon_1 - 1$ . Since, according to Thm. 4.4,

$$\widetilde{\eta}(x_U) = (\epsilon_0 + \epsilon_1 - 1)\eta^*(x_U) + 1 - \epsilon_0,$$

this holds as long as  $0 < \eta^*(x_U) < 1$ , as we assumed for  $P_{X_U}$ -almost all  $x_U \in \mathcal{X}_U$ .

**Infinite case** We now address the case where either  $\Pr[Y | X_S = x_S] \in \{0, 1\}$  or  $\Pr[Y | X_U = x_U] \in \{0, 1\}$ . By Lemma B.2, only one of these can happen at once,  $P_{X_S, X_U}$ -almost surely. Hence, since  $\lim_{n \rightarrow \infty} \widehat{\beta}_{1,n}$  is also finite almost surely, if  $\Pr[Y | X_S = x_S] \in \{0, 1\}$ , then  $\widehat{\eta}(x_S, x_U) = \sigma(\text{logit}(\Pr[Y | X_S = x_S])) = \eta(x_S, x_U)$ , while, if  $\Pr[Y | X_U = x_U] \in \{0, 1\}$ , then  $\widehat{\eta}(x_S, x_U) \rightarrow \sigma(\text{logit}(\Pr[Y | X_U = x_U])) = \eta(x_S, x_U)$ , in probability or almost surely, as appropriate.  $\square$

## C Multiclass Case

In the main paper, to simplify notation, we presented our unsupervised test-domain adaptation method in the case of binary labels  $Y$ . However, in many cases, including several of our experiments in Section 6, the label  $Y$  can take more than 2 distinct values. Hence, in this section, we show how to generalize our method to the multiclass setting and then present the exact procedure (Alg. 2) used in our multiclass experiments in Section 6.

Suppose we have  $K \geq 2$  classes. We “one-hot encode” these classes, so that  $Y$  takes values in the set

$$\mathcal{Y} = \{(1, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (0, \dots, 0, 1)\} \subseteq \{0, 1\}^K.$$

Let  $\epsilon \in [0, 1]^{\mathcal{Y} \times \mathcal{Y}}$  with

$$\epsilon_{y,y'} = \Pr[\widehat{Y} = y | Y = y']$$

denote the class-conditional confusion matrix of the pseudo-labels. Then, we have

$$\begin{aligned} \mathbb{E}[\widehat{Y} | X_U] &= \sum_{y \in \mathcal{Y}} \mathbb{E}[\widehat{Y} | Y = y, X_U] \Pr[Y = y | X_U] && \text{(Law of Total Expectation)} \\ &= \sum_{y \in \mathcal{Y}} \mathbb{E}[\widehat{Y} | Y = y] \Pr[Y = y | X_U] && \text{(Complementary)} \\ &= \epsilon \mathbb{E}[Y | X_U]. && \text{(Definition of } \epsilon) \end{aligned}$$

When  $\epsilon$  is non-singular, this has the unique solution  $\mathbb{E}[Y|X_U] = \epsilon^{-1} \mathbb{E}[\widehat{Y}|X_U]$ , giving a multiclass equivalent of Eq. (4.3) in Thm. 4.4. In practice, however, it is numerically more stable to estimate  $\mathbb{E}[Y|X_U]$  by the least-squares solution

$$\arg \min_{p \in \Delta^{\mathcal{Y}}} \left\| \epsilon p - \mathbb{E}[\widehat{Y}|X_U] \right\|_2,$$

which is what we will do in Algorithm 2. To estimate  $\epsilon$  without observing the label  $Y$  in the test domain, note that

$$\begin{aligned} \epsilon_{y,y'} &= \Pr[\widehat{Y} = y | Y = y'] = \frac{\Pr[\widehat{Y} = y, Y = y']}{\Pr[Y = y']} \\ &= \frac{\mathbb{E}[\Pr[\widehat{Y} = y, Y = y' | X_S]]}{\mathbb{E}[\Pr[Y = y' | X_S]]} \\ &= \frac{\mathbb{E}[\Pr[\widehat{Y} = y | X_S] \Pr[Y = y' | X_S]]}{\mathbb{E}[\Pr[Y = y' | X_S]]} \\ &= \frac{\mathbb{E}[f_{1,y}(X_S) f_{1,y'}(X_S)]}{\mathbb{E}[f_{1,y'}(X_S)]}. \end{aligned}$$

This suggests the estimate

$$\widehat{\epsilon}_{y,y'} = \frac{\sum_{i=1}^n \widehat{f}_{S,y}(X_{S,i}) \widehat{f}_{S,y'}(X_{S,i})}{\sum_{i=1}^n \widehat{f}_{S,y'}(X_{S,i})} = \sum_{i=1}^n \widehat{f}_{S,y}(X_{S,i}) \frac{\widehat{f}_{S,y'}(X_{S,i})}{\sum_{i=1}^n \widehat{f}_{S,y'}(X_{S,i})}$$

of each  $\epsilon_{y,y'}$ , or, in matrix notation,

$$\widehat{\epsilon} = f_S^T(X_S) \text{Normalize}(f_S(X_S)),$$

where  $\text{Normalize}(X)$  scales each column of  $X$  to sum to 1. This gives us a multiclass equivalent of Line 3 in Alg. 1.

The multiclass versions of Eq. (4.4) and Line 6 of Alg. 1 are slightly less straightforward. Specifically, whereas, in the binary case, we used the fact that  $\Pr[X_S, X_U | Y \neq 1] = \Pr[X_S, X_U | Y = 0] = \Pr[X_S | Y = 0] \Pr[X_U | Y = 0] = \Pr[X_S | Y \neq 1] \Pr[X_U | Y \neq 1]$  (by complementarity), in the multiclass case, we do not have  $\Pr[X_S, X_U | Y \neq 1] = \Pr[X_S | Y \neq 1] \Pr[X_U | Y \neq 1]$ . However, following similar reasoning as in the proof of Thm. 4.4, we have

$$\begin{aligned} \frac{\Pr[Y = y | X_S, X_U, E]}{\Pr[Y \neq y | X_S, X_U, E]} &= \frac{\Pr[Y = y | X_S, X_U, E]}{\sum_{y' \neq y} \Pr[Y = y' | X_S, X_U, E]} \\ &= \frac{\Pr[X_S, X_U | Y = y, E] \Pr[Y = y | E]}{\sum_{y' \neq y} \Pr[X_S, X_U | Y = y', E] \Pr[Y = y' | E]} && \text{(Bayes' Rule)} \\ &= \frac{\Pr[X_S | Y = y, E] \Pr[X_U | Y = y, E] \Pr[Y = y | E]}{\sum_{y' \neq y} \Pr[X_S | Y = y', E] \Pr[X_U | Y = y', E] \Pr[Y = y' | E]} && (X_S \perp\!\!\!\perp X_U | Y) \\ &= \frac{\Pr[Y = y | X_S, E] \Pr[Y = y | X_U, E]}{\sum_{y' \neq y} \Pr[Y = y' | X_S, E] \Pr[Y = y' | X_U, E]} \cdot \frac{\Pr[Y = y | E]}{\Pr[Y = y' | E]}. && \text{(Bayes' Rule)} \end{aligned}$$

Hence,

$$\begin{aligned} \text{logit}(\Pr[Y = y | X_S, X_U, E]) &= \log \left( \frac{\Pr[Y = y | X_S, E] \Pr[Y = y | X_U, E]}{\sum_{y' \neq y} \Pr[Y = y' | X_S, E] \Pr[Y = y' | X_U, E]} \cdot \frac{\Pr[Y = y | E]}{\Pr[Y = y' | E]} \right) \\ &= \log \left( \frac{Q_y}{\sum_{y' \neq y} Q_{y'}} \right) = \log \left( \frac{\frac{Q_y}{\|Q\|_1}}{\sum_{y' \neq y} \frac{Q_{y'}}{\|Q\|_1}} \right) = \text{logit} \left( \frac{Q_y}{\|Q\|_1} \right), \end{aligned}$$

for  $Q \in \mathbb{R}^{\mathcal{Y}}$  defined by

$$Q_y = \frac{f_{S,y}(X_S)f_{U,y}(X_U)}{\Pr[Y=y]} \quad \text{for each } y \in \mathcal{Y}.$$

In particular, applying the sigmoid function to each side, we have

$$\Pr[Y|X_S, X_U] = \frac{Q}{\|Q\|_1}.$$

We can estimate  $Q_y$  by

$$\hat{Q}_y = \frac{f_{S,y}(X_S)f_{U,y}(X_U)}{\frac{1}{n} \sum_{i=1}^n f_{S,y}(X_{S,i})}.$$

In matrix notation, this is

$$\hat{Q} = \frac{f_S(X_S) \circ f_U(X_U)}{\frac{1}{n} \sum_{i=1}^n f_S(X_{S,i})},$$

where  $\circ$  denotes element-wise multiplication. It follows that, for  $p \in \Delta^{\mathcal{Y}}$  (we will use  $p_y = \Pr[Y=y]$ ), we can use the multiclass combination function  $C : \Delta^{\mathcal{Y}} \times \Delta^{\mathcal{Y}} \rightarrow \Delta^{\mathcal{Y}}$  with

$$C_p(p_S, p_U) = \text{Normalize} \left( \frac{p_S p_U}{p} \right), \quad (\text{C.1})$$

where the multiplication and division are performed element-wise and  $\text{Normalize}(x) = \frac{x}{\|x\|_1}$ , to generalize Eq. (4.5). Putting these derivations together gives us our multiclass version of Alg. 1, presented in Alg. 2, where  $\Delta^{\mathcal{Y}} = \{z \in [0, 1]^K : \sum_{y \in \mathcal{Y}} z_y = 1\}$  denotes the standard probability simplex over  $\mathcal{Y}$ .

---

**Algorithm 2:** Multiclass bias-corrected adaptation procedure.

---

**Input:** Calibrated stable classifier  $f_S : \mathcal{X} \rightarrow \Delta^{\mathcal{Y}}$  with  $f_{S,y}(x_S) = \Pr[Y=y|X_S=x_S]$ ,  $n$  unlabeled samples  $\{(X_{S,i}, X_{U,i})\}_{i=1}^n$

**Output:** Joint classifier  $\hat{f} : \mathcal{X}_S \times \mathcal{X}_U \rightarrow \Delta^{\mathcal{Y}}$  estimating  $\Pr[Y=y|X_S=x_S, X_U=x_U]$

- 1 Compute soft pseudo-labels  $\{\hat{Y}_i\}_{i=1}^n$  with  $\hat{Y}_i = f_S(X_{S,i})$
  - 2 Compute soft class counts  $\hat{n} = \sum_{i=1}^n \hat{Y}_i$
  - 3 Estimate class-conditional pseudo-label confusion matrix  $\hat{\epsilon} \leftarrow f_S^T(X_S) \text{Normalize}(f_S^T(X_S))$
  - 4 Fit unstable classifier  $\tilde{f}_U(x_U)$  to pseudo-labelled data  $\{(X_{U,i}, \hat{Y}_i)\}_{i=1}^n$  //  $\approx \Pr[\hat{Y}=y|X_U]$
  - 5 Bias-correction  $\hat{f}_U(x_U) \mapsto \arg \min_{p \in \Delta^{\mathcal{Y}}} \|\epsilon p - \tilde{f}_U(x_U)\|_2$  //  $\approx \Pr[Y=y|X_U]$
  - 6 **return**  $\hat{f}(x_S, x_U) \mapsto C_{\hat{n}/n}(f_S(x_S), \hat{f}_U(x_U))$  // Eq. (C.1),  $\approx \Pr[Y=y|X_S, X_U]$
- 

## D Supplementary Results

### D.1 Trivial solution to joint-risk minimization

In Prop. D.1 below, we assume that the stable  $f_S(X)$  and unstable  $f_U(X)$  predictors output *logits*. In contrast, throughout the rest of the paper, we assume that  $f_S(X)$  and  $f_U(X)$  output *probabilities* in  $[0, 1]$ .

**Proposition D.1.** *Suppose  $\hat{Y}|f_S(X) \sim \text{Bernoulli}(\sigma(f_S(X)))$ , such that  $\hat{Y} \perp\!\!\!\perp f_U(X)|f_S(X)$ . Then,*

$$0 \in \arg \min_{f_U: \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}[\ell(\hat{Y}, \sigma(f_S(X) + f_U(X)))],$$

where  $\ell(x, y) = -x \log y - (1-x) \log(1-y)$  denotes the cross-entropy loss.

*Proof.* Suppose  $\hat{Y}|f_S(X) \sim \text{Bernoulli}(\sigma(f_S(X)))$ , such that  $\hat{Y} \perp\!\!\!\perp f_U(X)|f_S(X)$ . Then,

$$\begin{aligned}
& -\mathbb{E}[\ell(\hat{Y}, \sigma(f_S(X) + f_U(X)))] \\
&= \mathbb{E}[\mathbb{E}[\ell(\hat{Y}, \sigma(f_S(X) + f_U(X)))] \quad (\text{Law of Total Expectation}) \\
&= \mathbb{E}[\mathbb{E}[\hat{Y} \log \sigma(f_S(X) + f_U(X)) \\
&\quad + (1 - \hat{Y}) \log(1 - \sigma(f_S(X) + f_U(X)) | f_S(X))] \\
&= \mathbb{E}[\mathbb{E}[\hat{Y} | f_S(X_S)] \mathbb{E}[\log \sigma(f_S(X) + f_U(X)) | f_S(X_S)] \\
&\quad + \mathbb{E}[(1 - \hat{Y}) | f_S(X_S)] \mathbb{E}[\log(1 - \sigma(f_S(X) + f_U(X)) | f_S(X))] \quad (\hat{Y} \perp\!\!\!\perp f_U(X) | f_S(X)) \\
&= \mathbb{E}[\sigma(f_S(X)) \log \sigma(f_S(X) + f_U(X)) \\
&\quad + (1 - \sigma(f_S(X))) \log(1 - \sigma(f_S(X) + f_U(X)))] \quad (\hat{Y} | f_S(X) \sim \text{Bernoulli}(\sigma(f_S(X)))).
\end{aligned}$$

Since the cross-entropy loss is differentiable and convex, any  $f_U(X)$  satisfying  $0 = \frac{d}{df_U(X)} \mathbb{E}[\ell(\hat{Y}, \sigma(f_S(X) + f_U(X)))]$  is a minimizer. Indeed, under the mild assumption that the expectation and derivative commute, for  $f_U(X) = 0$ ,

$$\begin{aligned}
\frac{d}{df_U(X)} \mathbb{E}[\ell(\hat{Y}, \sigma(f_S(X) + f_U(X)))] &= -\mathbb{E} \left[ \frac{\sigma(f_S(X))}{\sigma(f_S(X) + f_U(X))} + \frac{1 - \sigma(f_S(X))}{1 - \sigma(f_S(X) + f_U(X))} \right] \\
&= -\mathbb{E} \left[ \frac{\sigma(f_S(X))}{\sigma(f_S(X))} + \frac{1 - \sigma(f_S(X))}{1 - \sigma(f_S(X))} \right] = 0.
\end{aligned}$$

□

## D.2 Causal perspectives

The stability, complementarity, and informativeness assumptions in Thm. 4.4 can be interpreted as constraints on the causal relationships between the variables  $X_S$ ,  $X_U$ ,  $Y$ , and  $E$ . We conclude this section with a result with a characterization of causal, directed acyclic graphs (DAGs) that are consistent with these assumptions. In particular, this result shows that our assumptions are satisfied in the “anti-causal” and “cause-effect” settings assumed in prior work [49, 68, 31], as well as work assuming only covariate shift (i.e., changes in the distribution of  $X$  without changes in the conditional  $P_{Y|X}$ ).

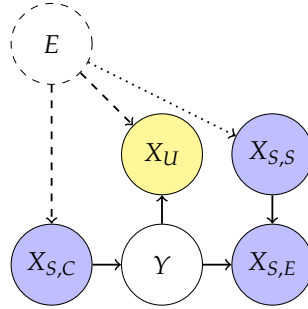


Figure 3: Causal DAGs over the environment  $E$ , three types of stable features (causes  $X_{S,C}$ , effects  $X_{S,E}$ , and spouses  $X_{S,S}$ ), unstable features  $X_U$ , and label  $Y$ , under conditions 1)-6). At least one, and possibly both, of the dashed edges  $E \rightarrow X_{S,C}$  and  $E \rightarrow X_U$  must be included. The dotted edge  $E \rightarrow X_{S,S}$  may or may not be included.

**Proposition D.2** (Possible Causal DAGs). *Consider an environment variable  $E$ , two covariates  $X_U$  and  $X_S$ , and a label  $Y$ . Assume there are no other hidden confounders (i.e., causal sufficiency). First, assume:*

- 1)  $E$  is a root (i.e., none of  $X_U$ ,  $X_S$ , and  $Y$  is an ancestor of  $E$ ).
- 2)  $X_S$  is informative of  $Y$  (i.e.,  $X_S \not\perp\!\!\!\perp Y|E$ ).
- 3)  $X_S$  and  $X_U$  are complementary predictors of  $Y$ ; i.e.,  $X_S \perp\!\!\!\perp X_U|(Y, E)$ .
- 4)  $X_S$  is stable (i.e.,  $E \perp\!\!\!\perp Y|X_S$ ).

These are the four structural assumptions under which Theorems 4.4 and 4.6 show that the SFB algorithm learns the conditional distribution  $P_{Y|X_S, X_U}$  in the test domain. Additionally, suppose

- 5)  $X_U$  is unstable (i.e.,  $E \not\perp\!\!\!\perp Y|X_U$ ). This is the case in which empirical risk minimization [ERM; 65] may suffer bias due to distribution shift, and hence when SFB may outperform ERM.
- 6)  $X_U$  contains some information about  $Y$  that is not included in  $X_S$  (i.e.,  $X_U \not\perp\!\!\!\perp Y|X_S$ ). This is information we expect invariant risk minimization [IRM; 1] is unable to learn, and hence when we expect SFB to outperform IRM.

Then,  $X_U$  consists of causal descendants (“effects”) of  $Y$ , while three types of stable features are possible:

1. causal ancestors  $X_{S,C}$  of  $Y$ ,
2. causal descendants  $X_{S,E}$  of  $Y$  that are not also descendants of  $E$ ,
3. causal spouses  $X_{S,S}$  of  $Y$  (i.e., causal ancestors of  $X_{S,E}$ ).

Notable special cases of the DAG in Figure 3 include:

1. the “cause-effect” settings, studied by Rojas-Carulla et al. [49], von Kügelgen et al. [68, 69], where  $X_S$  is a cause of  $Y$ ,  $X_U$  is an effect of  $Y$ , and  $E$  may affect both  $X_S$  and  $X_U$  but may affect  $Y$  only indirectly through  $X_S$ . Note that this generalizes the commonly used “covariate shift” assumption, as not only the covariate distribution  $P_{X_S, X_U}$  but also the conditional distribution  $P_{Y|X_U}$  can change between environments.
2. the “anti-causal” setting, studied by Jiang and Veitch [31], where  $X_S$  and  $X_U$  are both effects of  $Y$ , but  $X_S$  is unaffected by  $E$ .
3. the widely studied “covariate shift” setting [62, 23, 6, 61], which corresponds (see Sections 3 and 5 of Schölkopf [55]) to a causal factorization  $P(X, Y) = P(X)P(Y|X)$  (i.e., in which the only stable components  $X_S$  are causes  $X_{S,C}$  of  $Y$  or unconditionally independent (e.g., causal spouses  $X_{S,S}$ ) of  $Y$ ).

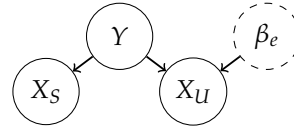
However, this model is more general than these special cases. Also, for sake of simplicity, we assumed causal sufficiency here; however, in the presence of unobserved confounders, other types of stable features are also possible; for example, if we consider the possibility of unobserved confounders  $U$  influencing  $Y$  that are independent of  $E$  (i.e., invariant across domains), then our method can also utilize stable features that are descendants of  $U$  (i.e., “siblings” of  $Y$ ).

## E Datasets

In our experiments, we consider five datasets: two (synthetic) numerical datasets and three image datasets. We now describe each dataset.

**Synthetic: Anti-causal (AC).** We consider an anti-causal synthetic dataset based on that of Jiang and Veitch [31, §6.1] where data is generated according to the following structural equations:

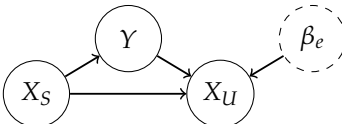
$$\begin{aligned} Y &\leftarrow \text{Rad}(0.5); \\ X_S &\leftarrow Y \cdot \text{Rad}(0.75); \\ X_U &\leftarrow Y \cdot \text{Rad}(\beta^e), \end{aligned}$$



where the input  $X = (X_S, X_U)$  and  $\text{Rad}(\beta)$  denotes a Rademacher random variable that is  $-1$  with probability  $1 - \beta$  and  $+1$  with probability  $\beta$ . Following Jiang and Veitch [31, §6.1], we create two training domains with  $\beta_e \in \{0.95, 0.7\}$ , one validation domain with  $\beta_e = 0.6$  and one test domain with  $\beta_e = 0.1$ .

**Synthetic: Cause-effect with a direct  $X_S$ - $X_U$  dependence (CE-DD).** We also consider a synthetic cause-effect dataset in which there is a direct dependence between  $X_S$  and  $X_U$ . In particular, following Jiang and Veitch [31, App. B], data is generated according to the following structural equations:



$$\begin{aligned}
X_S &\leftarrow \text{Bern}(0.5); \\
Y &\leftarrow \text{XOR}(X_S, \text{Bern}(0.75)); \\
X_U &\leftarrow \text{XOR}(\text{XOR}(Y, \text{Bern}(\beta_e)), X_S),
\end{aligned}$$


where the input  $X = (X_S, X_U)$  and  $\text{Bern}(\beta)$  denotes a Bernoulli random variable that is 1 with probability  $\beta$  and 0 with probability  $1 - \beta$ . Note that  $X_S \not\perp\!\!\!\perp X_U|Y$ , since  $X_S$  directly influences  $X_U$ . Following Jiang and Veitch [31, App. B], we create two training domains with  $\beta_e \in \{0.95, 0.8\}$ , one validation domain with  $\beta_e = 0.2$ , and one test domain with  $\beta_e = 0.1$ .

**ColorMNIST.** We next consider the CoLoRmNIST dataset [1]. This transforms the original MNIST dataset into a binary classification task (digit in 0–4 or 5–9) and then: (i) flips the label with probability 0.25, meaning that, across all 3 domains, digit shape correctly determines the label with probability 0.75; and (ii) colorizes the digit such that digit color (red or green) is a more informative but spurious feature (see Fig. 4).

**PACS.** We next consider the PACS dataset [37]—a 7-class image-classification dataset consisting of 4 domains: photos (P), art (A), cartoons (C) and sketches (S), with examples shown in Fig. 4. Model performances are reported for each domain after training on the other 3 domains.

**Camelyon17.** Finally, in the additional experiments of App. F.2, we consider the Camelyon17 [3] dataset from the WILDS benchmark [33]: a medical dataset with histopathology images from 5 hospitals which use different staining and imaging techniques (see Fig. 4). The goal is to determine whether or not a given image contains tumor tissue, making it a binary classification task across 5 domains (3 training, 1 validation, 1 test).

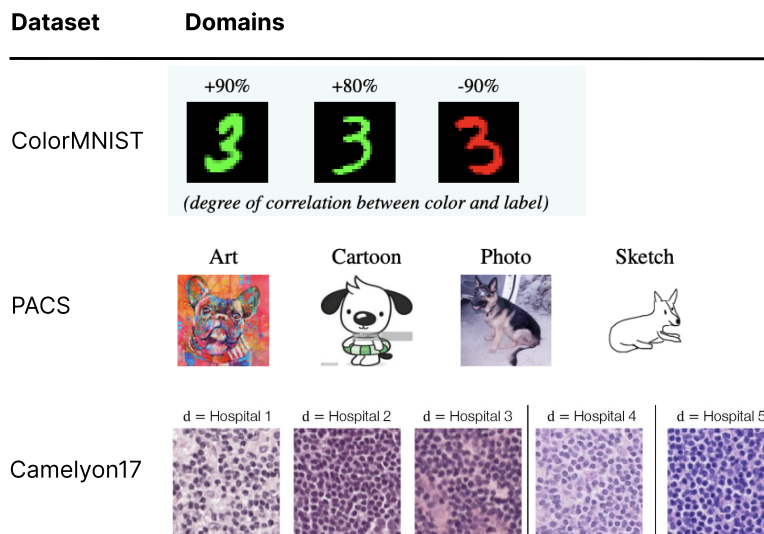


Figure 4: Examples from CoLoRmNIST [1], PACS [37] and Camelyon17 [3]. Figure and examples based on Gulrajani and Lopez-Paz [24, Table 3] and Koh et al. [33, Figure 4]. For CoLoRmNIST, we follow the standard approach [1] and use the first two domains for training and the third for testing. For PACS [37], we follow the standard approach [37, 24] and use each domain in turn for testing, using the remaining three domains for training. For Camelyon17 [3], we follow WILDS [33] and use the first three domains for training, the fourth for validation, and the fifth for testing.

## F Further Experiments

This appendix provides further experiments which supplement those in the main text. In particular, it provides: (i) an ablation on the CoLoRmNIST dataset showing the effects of bias correction, post-hoc calibration and multiple rounds of pseudo-labelling on SFB’s performance (F.1.1); (ii) the

performance of SFB on the CoLoRmNIST dataset when using different stability penalties (F.1.2); and (iii) results on a real-world medical dataset, Came1yon17 [3], where we find that all methods perform similarly *when properly tuned* (F.2).

## F.1 ColorMNIST

We now provide ablations on the CoLoRmNIST dataset to illustrate the effectiveness of the different components of SFB. In particular, we focus on bias correction and calibration, while also showing how multiple rounds of pseudo-labeling can improve performance in practice.

### F.1.1 Ablations

**Bias correction.** To adapt the unstable classifier in the test domain, SFB employs the bias-corrected adaptation algorithm of Alg. 1 (or Alg. 2 for the multi-class case) which corrects for biases caused by possible disagreements between the stable-predictor pseudo-labels  $\hat{Y}$  and the true label  $Y$ . In this (sub)section, we investigate the performance of SFB with and without bias correction (BC).

**Calibration.** As discussed in § 4.2, correctly combining the stable and unstable predictions post-adaptation requires them to be properly calibrated. In particular, it requires the stable predictor  $f_S$  to be calibrated with respect to the true labels  $Y$  and the unstable predictor  $f_U$  to be calibrated with respect to the pseudo-labels  $\hat{Y}$ . In this (sub)section, we investigate the performance of SFB with and without post-hoc calibration (in particular, simple temperature scaling [25]). More specifically, we investigate the effect of calibrating the stable predictor (CS) and calibrating the unstable predictor (CU).

**Multiple rounds of pseudo-labeling.** While SFB learns the optimal unstable classifier  $h_U^e$  in the test domain *given enough unlabelled data*, § 4.1 discussed how more accurate pseudo-labels  $\hat{Y}$  improve the sample efficiency of SFB. In particular, in a restricted-sample setting, more accurate pseudo-labels result in an unstable classifier  $h_U^e$  which better harnesses  $X_U$  in the test domain. With this in mind, note that, after adapting, we expect the joint predictions of SFB to be more accurate than its stable-only predictions. This raises the question: can we use these improved predictions to form more accurate pseudo-labels, and, in turn, an unstable classifier  $h_U^e$  that leads to even better performance? Furthermore, can we repeat this process, using multiple rounds of pseudo-labeling to refine our pseudo-labels and ultimately  $h_U^e$ ? While this multi-round approach loses the asymptotic guarantees of § 4.2, we found it to work quite well in practice. In this (sub)section, we thus investigate the performance of SFB with and without multiple rounds of pseudo-labeling (PL rounds).

Table 4: SFB ablations on CMNIST. Means and standard errors are over 3 random seeds. *BC*: bias correction. *CS*: post-hoc calibration of the stable classifier. *CU*: post-hoc calibration of the unstable classifier. *PL Rounds*: Number of pseudo-labeling rounds used. *GT adpt*: “ground-truth” adaptation using true labels in the test domain.

Algorithm	Bias	Calibration		PL Rounds	Test Acc.
	Correction	Stable	Unstable		
SFB no adpt.				1	$70.6 \pm 1.8$
SFB				1	$78.0 \pm 2.9$
+BC	✓			1	$83.4 \pm 2.8$
+CS		✓		1	$80.6 \pm 3.4$
+CU			✓	1	$76.6 \pm 2.4$
+BC+CS+CU	✓	✓	✓	1	$84.4 \pm 2.2$
+BC+CS	✓	✓		1	$84.9 \pm 2.6$
+BC+CS	✓	✓		2	$87.4 \pm 1.9$
+BC+CS	✓	✓		3	$88.1 \pm 1.8$
+BC+CS	✓	✓		4	$88.6 \pm 1.3$
+BC+CS	✓	✓		5	$88.7 \pm 1.3$
SFB GT adpt.	✓	✓		1	$89.0 \pm 0.3$

**Results.** Table 4 reports the ablations of SFB on CoLoRMNIST. Here we see that: (i) bias correction significantly boosts performance (+BC); (ii) calibrating the stable predictor also boosts performance without (+CS) and with (+BC+CS) bias correction, with the latter leading to the best performance; (iii) calibrating the unstable predictor (with respect to the pseudo-labels) slightly hurts performance without (+CU) and with (+BC+CS+CU) bias correction and stable-predictor calibration; (iv) multiple rounds of pseudo-labeling boosts performance, while also reducing the performance variation across random seeds; (v) using bias correction, stable-predictor calibration and 5 rounds of pseudo-labeling results in near-optimal adaptation performance, as indicated by the similar performance of SFB when using true labels  $Y$  to adapt  $h_U^e$  (denoted “SFB GT adpt.” in Table 4).

### F.1.2 Different stability penalties

In our experiments of § 6, we used IRM for the stability term of our SFB method, given in Eq. (5.1). However, as discussed in § 5, many other approaches exist for enforcing stability [35, 58, 47, 15, 67, 40, 78], and, in principle, any of these could be used. To illustrate this point, we now evaluate the performance of SFB when using different stability penalties, namely IRM [1], VREx [35], EQRM [15] and CLOvE [70]. For all penalties, we use SFB with bias correction, post-hoc calibration of the stable predictor, and 5 rounds of pseudo-labeling (see the ablation study of App. F.1.1).

Table 5: CMNIST test-domain accuracies for SFB with different stability penalties. Shown are the mean and standard error over 10 seeds.

Algorithm	Without Adaptation	With Adaptation
SFB w. IRM	70.6 ± 1.8	88.7 ± 1.3
SFB w. VREx	72.5 ± 1.0	88.7 ± 1.5
SFB w. EQRM	69.0 ± 2.8	88.2 ± 2.5
SFB w. CLOvE	67.0 ± 3.7	77.0 ± 6.6

### F.1.3 Full results

We now provide extended/full results of those provided in the main text. In particular, Table 6 represents an extended version of Table 3 in the main text, comparing against many more baseline methods. In addition, Table 7 provides the full numerical results for all adaptive baseline methods (described in App. G.1), which correspond to the plots of Fig. 2 in the main text.

Table 6: CMNIST test-domain accuracies. Mean and standard error are over 10 seeds. Extended/full version of Table 3 in the main text.

Algorithm	Test Acc.
ERM	27.9 ± 1.5
GroupDRO [53]	29.0 ± 1.1
IRM [1]	69.7 ± 0.9
SD [46]	70.3 ± 0.6
IGA [57]	57.7 ± 3.3
Fishr [48]	70.1 ± 0.7
V-REx [35]	71.6 ± 0.5
EQRM [15]	71.4 ± 0.4
SFB no adpt.	70.6 ± 1.8
SFB	<b>88.1 ± 1.8</b>
Oracle no adpt.	72.1 ± 0.7
Oracle	89.9 ± 0.1

## F.2 Camelyon17

We now provide results on the Camelyon17 [3] dataset. See App. E for a description of the dataset, and App. G.5 for implementation details.

Algorithm	Domain (Color-Label Correlation)										
	1.0	0.9	0.8	0.7	0.6	0.5	-0.6	-0.7	-0.8	-0.9	-1.0
ERM	97.5 ± 0.3	88.5 ± 0.4	79.7 ± 0.4	70.6 ± 0.5	61.4 ± 0.7	52.5 ± 0.4	43.5 ± 0.7	34.7 ± 0.7	25.1 ± 0.5	16.4 ± 0.4	7.6 ± 0.6
ERM+T3A	98.1 ± 0.2	88.9 ± 0.4	79.8 ± 0.4	70.4 ± 0.5	61.0 ± 0.8	51.7 ± 0.4	42.3 ± 0.7	33.0 ± 0.6	23.1 ± 0.4	13.8 ± 0.5	4.5 ± 0.5
ERM+PL (last)	97.6 ± 0.2	88.6 ± 0.3	79.7 ± 0.4	70.6 ± 0.5	61.4 ± 0.8	52.5 ± 0.4	43.4 ± 0.7	34.6 ± 0.7	25.0 ± 0.5	16.2 ± 0.3	7.4 ± 0.6
ERM+PL (all)	<b>100.0 ± 0.0</b>	90.0 ± 0.4	<b>80.2 ± 0.4</b>	70.1 ± 0.4	59.9 ± 0.8	50.1 ± 0.4	40.0 ± 0.5	30.1 ± 0.6	19.6 ± 0.3	9.9 ± 0.3	0.0 ± 0.1
IRM	70.6 ± 2.1	70.3 ± 1.9	70.4 ± 1.7	70.2 ± 1.1	69.9 ± 0.7	<b>69.9 ± 0.7</b>	70.1 ± 0.5	69.7 ± 0.6	69.8 ± 1.2	69.6 ± 1.6	69.4 ± 1.7
IRM+T3A	72.3 ± 1.7	71.2 ± 1.6	70.7 ± 1.6	70.2 ± 1.0	69.8 ± 0.7	<b>69.9 ± 0.7</b>	<b>70.3 ± 0.6</b>	70.6 ± 0.5	71.6 ± 1.1	72.4 ± 1.7	73.4 ± 1.9
IRM+PL (last)	70.8 ± 2.2	70.5 ± 1.9	70.5 ± 1.7	70.2 ± 1.1	69.9 ± 0.7	<b>69.9 ± 0.6</b>	70.0 ± 0.6	69.7 ± 0.6	69.9 ± 1.2	69.8 ± 1.6	69.7 ± 1.7
IRM+PL (all)	99.6 ± 1.2	89.4 ± 1.2	79.5 ± 1.3	68.7 ± 3.8	63.3 ± 4.7	63.5 ± 5.3	63.8 ± 4.5	67.5 ± 2.8	76.4 ± 4.2	87.2 ± 5.0	98.2 ± 3.0
SFB	<b>100.0 ± 0.1</b>	<b>90.5 ± 0.5</b>	79.8 ± 0.8	<b>71.0 ± 1.1</b>	<b>70.9 ± 0.3</b>	69.2 ± 0.4	68.4 ± 1.3	<b>71.2 ± 0.3</b>	<b>79.3 ± 1.2</b>	<b>88.7 ± 1.3</b>	<b>98.9 ± 1.5</b>

Table 7: CMNIST comparison with other test-time/source-free unsupervised domain adaptation methods. Means and standard errors are over 10 seeds. The largest mean per column/domain is in bold. “last”: only last-layer updated. “all”: all layers updated. Fig. 2 gives the corresponding plot.

Table 2 shows that ERM, IRM and SFB perform similarly on CameLyon17. In line with [24], we found that a properly-tuned ERM model can be difficult to beat on real-world datasets, particularly when the model is pre-trained on ImageNet and the dataset doesn’t contain severe distribution shift. While we conducted this proper tuning for ERM, IRM, and SFB (see App. G.5), doing so for ACTIR was non-trivial. We thus report the result from their paper [31, Table 1], which is likely lower due to sub-optimal hyperparameters. In particular, we found that, for ERM and IRM, using a lower learning rate (1e-5 vs 1e-4) and early stopping (1 vs 25 epochs) improved performance by 20 percentage points, from around 70% [31, Table 1] to around 90% (Table 8 below). It remains to be seen whether or not ACTIR can improve over a properly-tuned ERM model on CameLyon17.

While it may seem disappointing that SFB does not outperform the simpler methods of IRM and ERM on CameLyon17, we note that SFB can only be expected to do well when there is some gain in out-of-distribution performance from enforcing stability, e.g., when IRM outperforms ERM. The identical performances of IRM and ERM in Table 8 indicate that, with ImageNet pre-training and proper hyperparameter tuning, this is not the case for CameLyon17. Finally, despite the similar performances, we note that adapting SFB on CameLyon17 still gives a small performance boost and reduces the variance across random seeds.

Table 8: CameLyon17 test-domain accuracies. Mean and standard errors are over 5 random seeds. †: Result taken from [31, Tab. 1] and is likely lower due to sub-optimal hyperparameters (they report  $\approx 70\%$  for ERM and IRM).

Algorithm	Accuracy
ERM	90.2 ± 1.1
IRM	90.2 ± 1.1
ACTIR	77.7 ± 1.7 <sup>†</sup>
SFB no adpt.	89.8 ± 1.2
SFB	<b>90.3 ± 0.7</b>

## G Implementation Details

Below we provide further implementation details for the experiments of this work. Code is available at: <https://github.com/cianeastwood/sfb>.

### G.1 Adaptive baselines

For both the synthetic and CMNIST datasets, we compare against adaptive baseline methods by using pseudo-labeling (PL, [36]) and test-time classifier adjustment (T3A, [30]) on top of both ERM and IRM, choosing all adaptation hyperparameters using leave-one-domain-out cross-validation:

- *ERM/IRM + PL (last)*: After training with ERM/IRM, we update the last layer using the model’s own pseudo-labels [36].
- *ERM/IRM + PL (all)*: After training with ERM/IRM, we update all layers using the model’s own pseudo-labels [36].

- *ERM/IRM + T3A*: After training with ERM/IRM, we replace the classifier (final layer) with the template-based classifier of T3A [30]. This means: (i) computing template representations for each class using pseudo-labeled test-domain data; and (ii) classifying each example based on its distance to these templates.

## G.2 Synthetic experiments

Following Jiang and Veitch [31], we use a simple three-layer network with 8 units in each hidden layer and the Adam optimizer, choosing hyperparameters using the validation domain.

For SFB, we sweep over  $\lambda_S$  in  $\{0.01, 0.1, 1, 5, 10, 20\}$  and  $\lambda_C$  in  $\{0.01, 0.1, 1\}$ . For SFB’s unsupervised adaptation, we employ the bias correction of Alg. 1 and calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature to minimize the expected calibration error (ECE, [25]) on the validation domain. In addition, we use the Adam optimizer with an adaptation learning rate of 0.01, choosing the number of adaptation steps in  $[1, 20]$  (via early stopping) using the validation domain. Finally, we report the mean and standard error over 100 random seeds.

## G.3 ColorMNIST experiments

**Training details.** We follow the setup of Eastwood et al. [15, §6.1] and build on their open-source code<sup>5</sup>. In particular, we use the original MNIST training set to create training and validation sets for each domain, and the original MNIST test set for the test sets of each domain. For all methods, we use a 2-hidden-layer MLP with 390 hidden units, the Adam optimizer, a learning rate of 0.0001 with cosine scheduling, and dropout with  $p=0.2$ . In addition, we use full batches (size 25000), 400 steps for ERM pre-training (which directly corresponds to the delicate penalty “annealing” or warm-up periods used by penalty-based methods on CoLoRMNIST [1, 35, 15, 74]), and 600 total steps. We sweep over stability-penalty weights in  $\{50, 100, 500, 1000, 5000\}$  for IRM, VREx and SFB and  $a$ ’s in  $1 - \{e^{-100}, e^{-250}, e^{-500}, e^{-750}, e^{-1000}\}$  for EQRM. As the stable (shape) and unstable (color) features are conditionally independent given the label, we fix SFB’s conditional-independence penalty weight  $\lambda_C = 0$ . As is the standard for CoLoRMNIST, we use a test-domain validation set to select the best settings (after the total number of steps), and then report the mean and standard error over 10 random seeds on a test-domain test set. As in previous works, the hyperparameter ranges of all methods are selected by peeking at test-domain performance. While far from ideal, this is quite difficult to avoid with CoLoRMNIST and highlights a core problem with hyperparameter selection in DG—as discussed by many previous works [1, 35, 24, 74, 15].

**SFB adaptation details.** For SFB’s unsupervised adaptation in the test domain, we use a batch size of 2048 and employ the bias correction of Alg. 1. In addition, we calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature to minimize the expected calibration error (ECE, [25]) across the two training domains. Again using the two training domains for hyperparameter selection, we sweep over adaptation learning rates in  $\{0.1, 0.01\}$ , choose the best adaptation step in  $[5, 20]$  (via early stopping), and sweep over the number of pseudo-labeling rounds in  $[1, 5]$ . Finally, we report the mean and standard error over 3 random seeds for adaptation.

## G.4 PACS experiments

We follow the setup of Jiang and Veitch [31, § 6.4] and build on their open-source code<sup>6</sup>. This means using an ImageNet-pretrained ResNet-18, the Adam optimizer with a learning rate of  $10^{-4}$ , and choosing hyperparameters using leave-one-domain-out cross-validation (akin to K-fold cross-validation, except with domains). In particular, for each held-out test domain, we train 3 models—each time leaving out 1 of the 3 training domains for validation—and then select hyperparameters based on the best average performance across the held-out validation domains. Finally, we use the selected hyperparameters to retrain the model using all 3 training domains.

For SFB, we sweep over  $\lambda_S$  in  $\{0.01, 0.1, 1, 5, 10, 20\}$ ,  $\lambda_C$  in  $\{0.01, 0.1, 1\}$ , and learning rates in  $\{10^{-4}, 50^{-4}\}$ . For SFB’s unsupervised adaptation, we employ the multi-class bias correction of Alg. 2 and calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature

<sup>5</sup><https://github.com/cianeastwood/qrm/tree/main/CMNIST>

<sup>6</sup><https://github.com/ybjiaang/ACTIR>.

to minimize the expected calibration error (ECE, [25]) across the three training domains. In addition, we use the Adam optimizer with an adaptation learning rate of 0.01, choosing the number of adaptation steps in  $[1, 20]$  (via early stopping) using the training domains. Finally, we report the mean and standard error over 5 random seeds.

### G.5 Camelyon17 experiments

We follow the setup of Jiang and Veitch [31, § 6.3] and build on their open-source code<sup>7</sup>. This means using an ImageNet-pretrained ResNet-18, the Adam optimizer, and, following [33], choosing hyperparameters using the validation domain (hospital 4). In contrast to [31], we use a learning rate of  $10^{-5}$  for all methods, rather than  $10^{-4}$ , and employ early stopping using the validation domain. We found this to significantly improve all methods. E.g., the baselines of ERM and IRM improve by approximately 20 percentage points, jumping from  $\approx 70\%$  to  $\approx 90\%$ .

For SFB, we sweep over  $\lambda_S$  in  $\{0.01, 0.1, 1, 5, 10, 20\}$  and  $\lambda_C$  in  $\{0.01, 0.1, 1\}$ . For SFB’s unsupervised adaptation, we employ the bias correction of Alg. 1 and calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature to minimize the expected calibration error (ECE, [25]) on the validation domain. In addition, we use the Adam optimizer with an adaptation learning rate of 0.01, choosing the number of adaptation steps in  $[1, 20]$  (via early stopping) using the validation domain. Finally, we report the mean and standard error over 5 random seeds.

## H Further Related Work

**Learning with noisy labels.** An intermediate goal in our work, namely learning a model to predict  $Y$  from  $X_U$  using pseudo-labels based on  $X_S$ , is an instance of *learning with noisy labels*, a widely studied problem [56, 44, 7, 60, 38, 64, 75]. Specifically, under the complementarity assumption ( $X_S \perp\!\!\!\perp X_U|Y$ ), the accuracy of the pseudo-labels on each class is independent of  $X_U$ , placing us in the so-called *class-conditional random noise model* [56, 44, 7, 75]. As we discuss in Section 4, our theoretical insights about the special structure of pseudo-labels complement existing results on learning under this model. Our bias-correction (Eq. (4.3)) for  $P_{Y|X_U}$  is also closely related to the “method of unbiased estimators” [44] and to the bias correction proposed in Eq. (1) of Zhang et al. [75]. However, rather than correcting the loss used in ERM, our post-hoc bias correction applies to any calibrated classifier. Moreover, our ultimate goal, learning a predictor of  $Y$  *jointly* using  $X_S$  and  $X_U$ , is not captured by learning with noisy labels.

**Co-training.** Our use of stable-feature pseudo-labels to train a classifier based on a disjoint subset of (unstable) features is reminiscent of co-training [10]. Both methods benefit from conditional independence of the two feature subsets given the label to ensure that they provide complementary information.<sup>8</sup> The key difference is that while co-training requires (a small number of) labeled samples from the *same distribution as the test data*, our method instead uses labeled data from a *different distribution* (training domains), along with the assumption of a stable feature. Additionally, while co-training iteratively refines two pre-trained classifiers symmetrically based on each other’s predictions, our method only trains the unstable classifier, in a single iteration, using the stable classifier’s predictions.

**Boosting.** Our method of building a strong (albeit unstable) classifier using a weak (but stable) one is reminiscent of boosting, in which one ensembles weak classifiers to create a single strong classifier [54] and which inspires the name of our approach, “stable feature boosting (SFB)”. However, whereas traditional boosting improves weak classifiers by examining how their predictions differ from true labels, our adaptation method utilizes only pseudo-labels and needs no true labels from the test domain. For example, while traditional boosting only refines functions of existing features, SFB can utilize new features that are only available in the test domain.

**Learning theory for domain generalization.** In addition to often assuming particular kinds of distribution shifts (e.g., covariate shift), existing error bounds for domain generalization often depend on some notion of distance between training and test domains (which does not vanish as more data is collected within domains) [9, 5, 77, 76] or assume that the test domain has a particular structural

<sup>7</sup>See Footnote 6.

<sup>8</sup>See Krogel and Scheffer [34], Blum and Mitchell [10, Theorem 1] for discussion of this assumption.



relationship with the training domains (e.g., is a convex combination of training domains [41]). In contrast, under the structure of invariant and complementary features, we show that consistent generalization (i.e., with generalization error vanishing as more data is collected within domains) is possible in *any* test domain. Additionally, whereas these prior works derive uniform convergence bounds (implying good generalization for ERM), our results demonstrate the benefit of an additional bias-correction step after training. We also note that, in much of this literature, “invariance” refers to invariance of the covariate marginal distribution  $P_X$  across domains; in contrast, our notion of stable features (Defn. 4.1) refers to invariance of the conditional  $P_{Y|X}$ .

## I Performance When Complementarity is Violated

Thm. 4.4 justifies the bias correction of Eq. (4.3) under the assumption that stable  $X_S$  and unstable  $X_U$  features are complementary, i.e., conditionally independent given the label  $Y$ . In this section, we discuss what happens if this assumption is relaxed and provide some intuition for why the bias correction appears to help even when complementarity is violated (as we observed in some of our experiments). In particular, we provide an argument that, in most cases, the bias correction should improve the accuracy of a naive classifier by making it agree more often with the Bayes-optimal classifier. While not a rigorous proof, we believe that this argument provides some insight into SFB’s strong performance even when complementarity is violated.

In the absence of complementarity, the quantity  $\Pr[\hat{Y} = 1|Y = 1, X_U = x_U]$  no longer reduces to the class-wise accuracy  $\Pr[\hat{Y} = 1|Y = 1]$ ; thus we write more generally  $\epsilon_1(x_U) = \Pr[\hat{Y} = 1|Y = 1, X_U = x_U]$ , and we write  $\bar{\epsilon}_1 = \mathbb{E}_{X_U}[\epsilon_1(X_U)] = \Pr[\hat{Y} = 1|Y = 1]$  instead of simply  $\epsilon_1$  for the accuracy on class 1. Similarly, we write  $\epsilon_0(x_U) = \Pr[\hat{Y} = 0|Y = 0, X_U = x_U]$ , and we write  $\bar{\epsilon}_0 = \mathbb{E}_{X_U}[\epsilon_0(X_U)] = \Pr[\hat{Y} = 0|Y = 0]$  instead of simply  $\epsilon_0$  for the accuracy on class 0.

Let  $f_*(x_U) = \Pr[Y = 1|X_U = x_U]$  denote the true regression function, and let  $h_*(x_U) = 1\{f_*(x_U) > 0.5\}$  denote the Bayes-optimal classifier. It is well known that the Bayes-optimal classifier  $h_*$  has the maximum possible accuracy out of all classifiers. Thus, the sub-optimality of a classifier  $h$  can be measured by the probability  $S(h) = \Pr_{X_U}[h(X_U) \neq h_*(X_U)]$  that it disagrees with the Bayes-optimal classifier. Our next result expresses  $S(h)$  in terms of the true regression function  $f_*$ , the functions  $\epsilon_0$  and  $\epsilon_1$ , and the distribution of  $X_U$ , when  $h$  is the bias-corrected classifier

$$h_{BC}(x_U) := 1 \left\{ \frac{\Pr[\hat{Y} = 1|X_U = x_U] + \bar{\epsilon}_0 - \bar{\epsilon}_1}{\bar{\epsilon}_0 + \bar{\epsilon}_1 - 1} > 0.5 \right\}$$

from Thm. 4.4 or when  $h$  is the “naive” classifier

$$h_{Naive}(x_U) := 1 \left\{ \Pr[\hat{Y} = 1|X_U = x_U] > 0.5 \right\}$$

that simply treats the pseudo-labels as true labels.

### Proposition I.1.

$$S(h_{BC}) = \Pr_{X_U} \left[ |f_*(X_U) - 0.5| \leq \frac{|\epsilon_0(X_U) - \epsilon_1(X_U) - \mathbb{E}_{X_U}[\epsilon_0(X_U) - \epsilon_1(X_U)]|}{2(\epsilon_0(X_U) + \epsilon_1(X_U) - 1)} \right],$$

and

$$S(h_{Naive}) = \Pr_{X_U} \left[ |f_*(X_U) - 0.5| \leq \frac{|\epsilon_0(X_U) - \epsilon_1(X_U)|}{2(\epsilon_0(X_U) + \epsilon_1(X_U) - 1)} \right].$$

These two formulae for  $S(h_{BC})$  and  $S(h_{Naive})$  differ only in the numerator of the right-hand side; letting  $Z := \epsilon_0(X_U) - \epsilon_1(X_U)$ , the sub-optimality of  $h_{BC}$  scales with  $|Z - \mathbb{E}[Z]|$ , whereas the sub-optimality of  $h_{Naive}$  scales with  $|Z|$ . Intuitively, for all except very pathological random variables  $Z$ ,  $|Z - \mathbb{E}[Z]|$  is typically smaller than  $|Z|$ . Although not a rigorous proof that the bias correction is always better than the naive classifier, this analysis provides an argument that, in most cases, the bias correction should improve on the accuracy of the naive classifier, by making it agree more often with the Bayes-optimal classifier.

We conclude by sketching the proof of Proposition I.1:

*Proof.* By construction, a thresholding classifier  $h(x) = 1\{f(x) > 0.5\}$  disagrees with the Bayes-optimal classifier if and only if

$$f(x) \leq 0.5 < f_*(x) \quad \text{or} \quad f_*(x) \leq 0.5 < f(x).$$

Expanding these inequalities in the cases  $f(x) = \frac{\Pr[\hat{Y}=1|X_U=x] + \bar{\epsilon}_0 - \bar{\epsilon}_1}{\bar{\epsilon}_0 + \bar{\epsilon}_1 - 1}$  and  $f(x) = \Pr[\hat{Y} = 1|X_U = x]$  and solving for the quantity  $f_*(x) - 0.5$  in each case gives Proposition I.1.  $\square$



## **A.4 Disentangled Representations (§ 6.2)**

## A PROOFS

## A.1 PROOF OF PROPOSITION 3.3

**Proposition 3.3.** *If  $D = C = 1$  and  $K = L$  (i.e.,  $\dim(\mathbf{c}) = \dim(\mathbf{z})$ ), then  $\mathbf{R}$  is a permutation matrix.*

*Proof.* First, by Defn. 2.1, we have  $0 \leq R_{ij}$  and  $\sum_{i=1}^L R_{ij} = 1 \forall i, j$ , so  $0 \leq R_{ij} \leq 1$ . It follows that  $\forall i, j : P_i, \tilde{P}_j \in \Delta_{K-1}$ , where  $\Delta_{K-1}$  denotes the  $K$ -dim. probability simplex, i.e.,  $P_i$  and  $\tilde{P}_j$  are valid probability vectors. Hence, the Shannon entropies  $H_K(P_i), H_K(\tilde{P}_j)$  are well-defined  $\forall i, j$ , and, due to using  $\log_K$  in the definition of  $H_K$  (see § 2), are bounded in  $[0, 1]$ . It follows that  $\forall i, j : 0 \leq D_i \leq 1$  and  $0 \leq C_j \leq 1$ . Since  $D$  and  $C$  are convex combinations of the  $D_i$  and  $C_j$ , we have

$$\begin{aligned} D = 1 &\iff \forall i : D_i = 1 \iff \forall i : H_K(P_i) = 0, \\ C = 1 &\iff \forall j : C_j = 1 \iff \forall j : H_K(\tilde{P}_j) = 0. \end{aligned}$$

Now for any  $\mathbf{p} = (p_1, \dots, p_K) \in \Delta_{K-1}$ , we have that

$$H_K(\mathbf{p}) = -\sum_{k=1}^K p_k \log_K p_k = 0 \iff \forall k : p_k \log_K p_k = 0 \iff \forall k : p_k \in \{0, 1\}$$

where  $p_k \log p_k := 0$  for  $p_k = 0$ , consistent with  $\lim_{x \rightarrow 0^+} x \log x = 0$ . Together with the simplex constraint, this implies that  $\mathbf{p}$  must be a standard basis vector  $\mathbf{p} = \mathbf{e}_l$  for some  $l$ , i.e.,  $p_l = 1$  and  $p_k = 0$  for  $k \neq l$ . Hence,  $P_i, \tilde{P}_j$  must be standard basis vectors for all  $i, j$ , and so each row and column of  $\mathbf{R}$  contains exactly one non-zero element. Since columns of  $\mathbf{R}$  sum to one, these non-zero elements must all be one.  $\square$

## A.2 PROOF OF COROLLARY 3.4

**Corollary 3.4.** *Under the same conditions as Prop. 3.3, if  $\mathbf{z} = \mathbf{W}^\top \mathbf{c}$  (so that  $I = 1$ ) for some  $\mathbf{W}$  with  $R_{ij} = \frac{|w_{ij}|}{\sum_{i=1}^L |w_{ij}|}$ , then  $\mathbf{c}$  identifies  $\mathbf{z}$  up to permutation and sign (Defn. 3.1).*

*Proof.* First, we show that  $R_{ij} = \frac{|w_{ij}|}{\sum_{i=1}^L |w_{ij}|}$  is a well-defined feature importance matrix. Suppose for a contradiction, that  $\sum_{i=1}^L |w_{il}| = 0$  for some  $l$ . Since  $|w_{il}| \geq 0$ , this implies  $w_{il} = 0$  for all  $i$ . Consider  $z_l = \sum_{i=1}^L w_{il} c_i$ . Taking the covariance, we obtain  $\text{Var}[z_l] = \sum_{i,j=1}^L w_{il} w_{jl} \text{Cov}(c_i, c_j) = 0$ , which is a contradiction since  $z_l$  has positive (unit) variance by the normalisation assumption (see footnote 1). Hence,  $\sum_{i=1}^L |w_{il}| > 0$  for all  $l$ . Thus  $\mathbf{R}$  is well-defined, with its elements being non-negative and its columns summing to one by construction, so it is a valid feature importance matrix.

Next, note that we can write  $\mathbf{R} = |\mathbf{W}|\mathbf{D}$  where  $\mathbf{D}$  is the invertible diagonal matrix with positive diagonal entries  $D_{jj} = \frac{1}{\sum_{i=1}^L |w_{ij}|} > 0$ .

By Prop. 3.3,  $\mathbf{R}$  is a permutation matrix, so  $\mathbf{R} = \mathbf{P} = |\mathbf{W}|\mathbf{D}$  for some permutation matrix  $\mathbf{P}$ . Right multiplication by  $\mathbf{D}^{-1}$  yields  $\mathbf{P}\mathbf{D}^{-1} = |\mathbf{W}|$ , that is  $|\mathbf{W}|$  has exactly one non-zero, positive element in each row and each column (and zeros elsewhere). Thus  $\mathbf{W}$  and therefore also  $\mathbf{W}^\top$  are generalised permutation matrices. Hence  $(\mathbf{W}^\top)^{-1}$  exists and is also a generalised permutation matrix.

Finally, consider  $\mathbf{c} = (\mathbf{W}^\top)^{-1} \mathbf{z}$ . Since all but one element in each row of  $(\mathbf{W}^\top)^{-1}$  are zero, we have for any  $i : c_i = \tilde{w}_{ij} z_j$  for some  $j$ , where  $\tilde{w}_{ij}$  denotes the  $(i, j)$  element of  $(\mathbf{W}^\top)^{-1}$ . By considering the variances of both sides and recalling that all  $c_i$ 's and  $z_j$ 's are normalised to unit variance, it follows that  $1 = \text{Var}(c_i) = \tilde{w}_{ij}^2 \text{Var}(z_j) = \tilde{w}_{ij}^2$ . Hence,  $\tilde{w}_{ij}^2 = \pm 1$  and so  $(\mathbf{W}^\top)^{-1}$  is, in fact, a signed permutation matrix, which concludes the proof.  $\square$

## A.3 PROOF OF COROLLARY 3.5

**Corollary 3.5.** *Under the same conditions as Prop. 3.3, let  $\mathbf{z} = f(\mathbf{c})$  (so that  $I = 1$ ) with  $f$  an invertible and differentiable nonlinear function, and let  $\mathbf{R}$  be a matrix of relative feature importances*

Published as a conference paper at ICLR 2023

for  $f$  (Defn. 2.1) with the property that  $R_{ij} = 0$  if and only if  $f_j$  does not depend on  $c_i$ , i.e.,  $\|\partial_i f_j\|_2 = 0$ . Then  $c$  identifies  $\mathbf{z}$  up to permutation and element-wise reparametrisation (Defn. 3.2).

*Proof.* For any  $j$  consider  $z_j = f_j(\mathbf{c})$ . By Prop. 3.3,  $R$  is a permutation matrix, so column  $j$  of  $R$  contains exactly one non-zero entry in row  $\pi(j)$  for some permutation  $\pi$  of  $\{1, \dots, K\}$ . Hence, by the assumed property of  $R$ ,  $f_j(\mathbf{c})$  does not depend on  $c_i$  for all  $i \neq \pi(j)$ , and thus  $z_j = f_j(c_{\pi(j)}) \forall j$ . By invertibility of  $f$ , we obtain  $c_j = h_j(z_{j'})$  with  $h_j = f_{j'}^{-1}$  and  $j' = \pi^{-1}(j)$ .  $\square$

## B ADDITIONAL EXPERIMENTAL RESULTS

### B.1 DOWNSTREAM CORRELATIONS

Here we present the full results of the correlations between the DCIE scores and downstream performance, the latter with low-capacity probes (as discussed in § 6.3).

In Tab. 3 and Tab. 4 we show the values of the Pearson and Spearman correlations alongside the corresponding  $p$ -values<sup>5</sup>. Note that some of the assumptions behind these  $p$ -values, e.g. that the DCIE scores and downstream performances are normally distributed, likely do not hold. Thus, these  $p$ -values should not be interpreted as precise probabilities but rather as rough indications of statistical significance. In Tab. 5 we show the correlations for regression and classification tasks separately, with both task types exhibiting similar correlations. We note that E has the strongest correlation with the downstream performance (when using low-capacity probes for the downstream task).

Table 3: Pearson correlation coefficient  $\rho$  between the D, C, I, and E scores and downstream performance, along with the corresponding  $p$ -values (in parentheses). See § 6.3 for experimental details.

Probe $f$	D	C	I	E
MLP	0.15 ( $4 \times 10^{-1}$ )	0.28 ( $9 \times 10^{-1}$ )	0.47 ( $9 \times 10^{-3}$ )	<b>0.96</b> ( $8 \times 10^{-18}$ )
RF	0.8 ( $1 \times 10^{-7}$ )	0.70 ( $2 \times 10^{-5}$ )	0.4 ( $3 \times 10^{-2}$ )	<b>0.88</b> ( $2 \times 10^{-10}$ )

Table 4: Spearman rank correlation between the D, C, I, and E scores and downstream performance, along with the corresponding  $p$ -values (in parentheses).

Probe $f$	D	C	I	E
MLP	0.12 ( $5 \times 10^{-1}$ )	-0.07 ( $7 \times 10^{-1}$ )	0.55 ( $2 \times 10^{-3}$ )	<b>0.94</b> ( $1 \times 10^{-14}$ )
RF	<b>0.81</b> ( $6 \times 10^{-8}$ )	0.75 ( $2 \times 10^{-6}$ )	0.28 ( $1 \times 10^{-1}$ )	0.78 ( $3 \times 10^{-7}$ )

Table 5: Pearson correlation coefficient  $\rho$  between the D,C,I, E scores and downstream performance for each task type (regression and classification). Correlations are similar across both task types.

Probe $f$	Task	D	C	I	E
MLP	Regression	0.16	0.04	0.46	<b>0.96</b>
	Classification	0.14	0.00	0.48	<b>0.96</b>
RF	Regression	0.76	0.66	0.42	<b>0.84</b>
	Classification	0.81	0.72	0.35	<b>0.89</b>

**Score-by-score analysis.** To get a deeper insight into the correlations reported in Tabs. 3 and 4, we plot each of the D, C, I and E scores against downstream performance for each of the 30 models considered in § 6.3. As shown in Figs. 6 to 9, only E correlates strongly with downstream performance

<sup>5</sup>Computed using <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html>

Published as a conference paper at ICLR 2023

for both probe types, again highlighting: (i) the value that E adds to the existing DCI framework; and (ii) the practical usefulness of reporting E when comparing/evaluating learned representations.

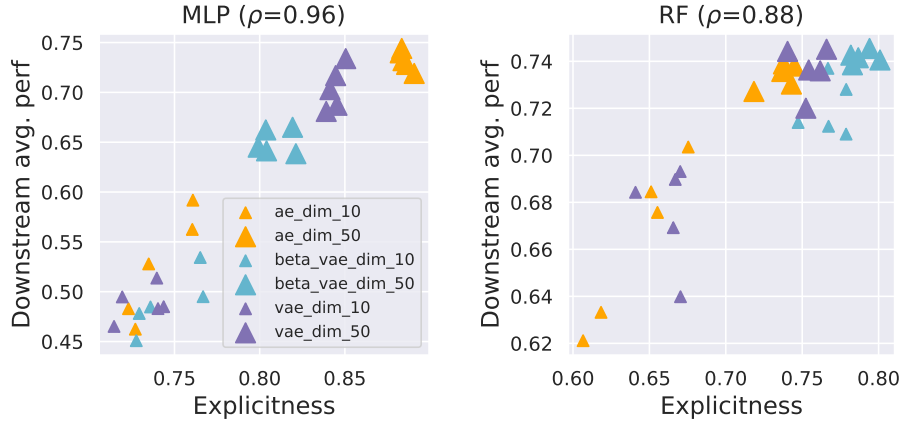


Figure 6: **Explicitness (E) vs. downstream performance.** Scatter plots show 30 data points: 3 models (AEs, VAEs,  $\beta$ -VAEs)  $\times$  2 latent dimensionalities ( $L = 10$  and  $L = 50$ )  $\times$  5 random seeds.

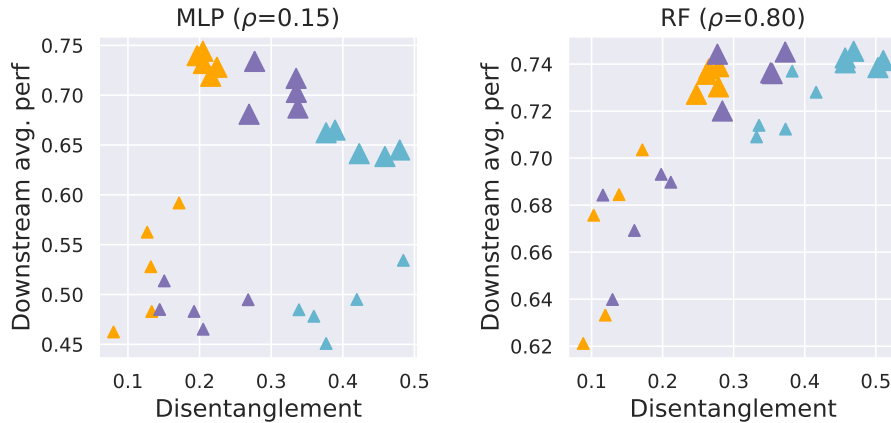


Figure 7: **Disentanglement (D) vs. downstream performance.** Scatter plots show 30 data points: 3 models (AEs, VAEs,  $\beta$ -VAEs)  $\times$  2 latent dimensionalities ( $L = 10$  and  $L = 50$ )  $\times$  5 random seeds.

## B.2 DIFFERENT AMOUNTS OF DATA

In Fig. 10 we present loss-capacity curves obtained when using different amounts of data to train the MLP probes. As shown, larger datasets have smaller performance gaps between (i) synthetic and learned representations; and (ii) small and large representations.

Published as a conference paper at ICLR 2023

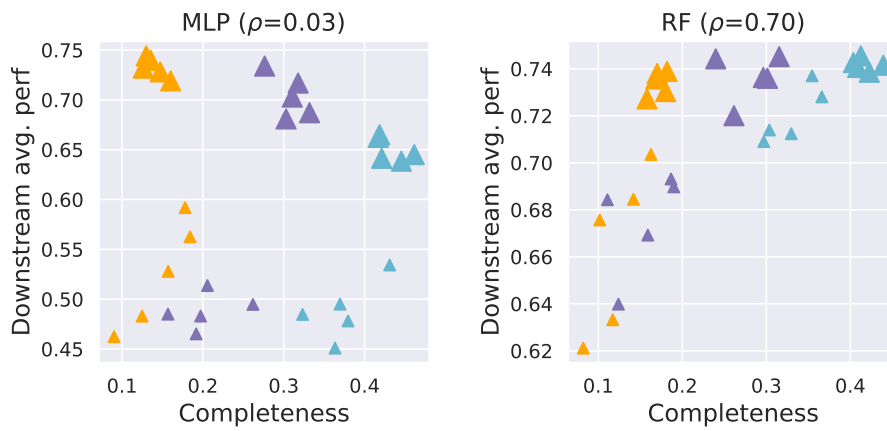


Figure 8: **Completeness (C) vs. downstream performance.** Scatter plots show 30 data points: 3 models (AEs, VAEs,  $\beta$ -VAEs)  $\times$  2 latent dimensionalities ( $L = 10$  and  $L = 50$ )  $\times$  5 random seeds.

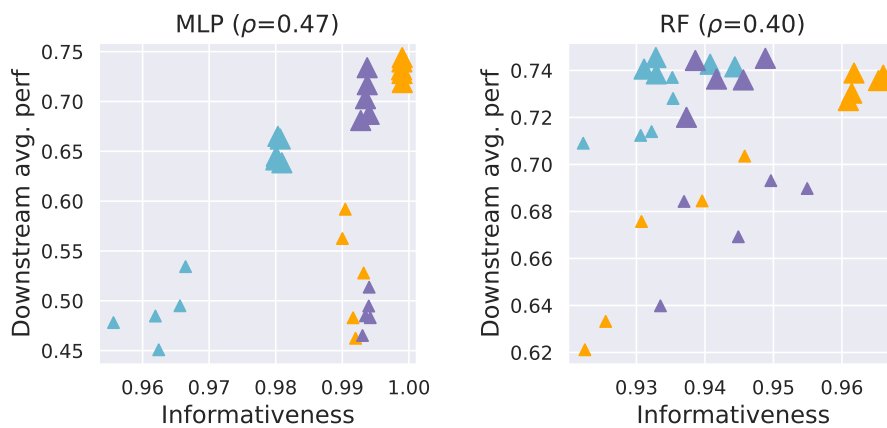


Figure 9: **Informativeness (I) vs. downstream performance.** Scatter plots show 30 data points: 3 models (AEs, VAEs,  $\beta$ -VAEs)  $\times$  2 latent dimensionalities ( $L = 10$  and  $L = 50$ )  $\times$  5 random seeds.

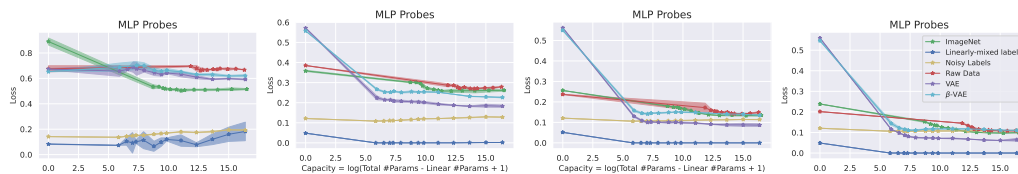


Figure 10: Loss-capacity curves for MPI3D-Real subsets of size 100, 1000, 5000 and 10000 respectively.