Unlocking AI outside the training distribution: Generalization, Causality, and Coronary Risk Modeling

by

Aahlad Manas Puli

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

September, 2024

Prof. Rajesh H. Ranganath

Dedication

To Amma and Nanna.

Acknowledgements

I continue to count my blessings for having Rajesh Ranganath as my advisor. Acting as a mentor and role model over the last decade, he has unquestionably had the most significant effect on my growth. I can only hope to approximate the care, patience, and resilience with which he approaches both life and research. I want to thank my committee, Kyunghyun Cho, Rob Fergus, Rich Zemel, and Kevin Murphy, for their support and feedback. I want to thank Kyunghyun for his invaluable advice and for sharing his enthusiasm for research, as well as Uri Shalit and David Sontag for their gentle guidance during my first foray into ML research and continued help in my career.

I owe a great deal to my family for their unwavering support in my non-traditional ventures. I want to thank my parents for their constant love and guidance, my cousins for filling the void where siblings would have been, my aunts, uncles, and grandparents for their presence in my life, and finally my late grandfather, Seshadri Reddy, for being an inspiration.

There are many at NYU without whom my PhD journey would have been much worse off. Chief amongst them is Mark Goldstein; we met on Day 1 of the program, and he has since been my roommate, office mate, and research collaborator, becoming prime counterexamples to the adage "familiarity breeds contempt." I consider myself lucky to have known, collaborated with, and learned from Mukund, Wouter, Raghav, Xintian, Neil, Lily, Yoav, Adriel, Wanqian, Vincent, Walter, Hao, and Nhi. My PhD cohort, David, Minjae, Aaron, Katrina, and Mimee, taught me various things during the PhD while calling me out on things not said, thought of, or considered. I want to thank Nikita, Katrina, and Andrew for their constant friendship. Others at NYU, Yunfan, Anthony, Will, Alex, Alfredo, Mahi, Irmak, Vaibhav, and Lerrel, have always been around to share coffee, stories, hikes, and insights.

I am grateful for the many people in my life who blur the lines between friends and family. I want to thank Veena, Deva, Prabhat, Prasanth, Hema, Nalini, and Gunaa for being a collective ball of positive energy, and Polly, Bharath, Pranjali, Trical for being adults when appropriate and children otherwise. I also want to thank Arjun, Arun, and Mardava for sitting with me during my many non-trivially-improvised DMing sessions. I cannot name all the friends from before grad school that have made my life better, but I can start the list here: Amrutha, Moc, Sreekar, Venkat, Subbu, Sailu, RLV, Pandu, Akshay, Hemanth, Nivvedan, Bomma, Dheeraj, Amanda, Naren, Abhijith, Abhik, Kavya, Sidharth, Suthirth, Harsha have all brought a much-needed combination of goblin energy and fulfilling conversation to my life. Know that if you are not on this list, you will be in my thoughts.

Funding acknowledgement

The work presented in this dissertation was partly supported by the Apple Scholars in AIML PhD fellowship 2022, MacCracken Fellowship, NIH/NHLBI Award R01HL148248, NSF Award 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science, NSF CAREER Award 2145542, ONR N00014-23-1-2634.

Abstract

Modern AI models make it easy to exploit the correlations in a dataset to predict a target of interest from a given set of inputs. However, the primary use of these models often lies outside the training data. For example, while one can train a Transformer to correlate a patient's medical history to their chances of developing coronary heart disease (CHD), the goal would be to estimate risks on populations elsewhere or in the future. Challenges arise if the model relies on correlations that shift between training and test times or capture non-causal relationships. Predictions based on unstable relationships can degrade outside the training distribution, and basing treatment decisions on non-causal relationships can result in harm. This thesis first develops a methodology for generalizing out-of-distribution (OOD) and estimating causal effects. It closes with an empirical study of building and transporting CHD risk models at two large hospital systems.

The first part begins by defining a class of distribution shifts where standard training or balancing the data yield models can perform worse than random guessing. We characterize representations that generalize across such shifts and derive an algorithm to build models with such representations. Next, we develop an approach to encode knowledge of features used by humans into building robust models. The last work in this part identifies biases implicit in the standard way of training — gradient-based optimization of cross-entropy — that force models to depend more on unstable features than on the more informative stable ones. We develop a class of loss functions to encourage dependence on the more informative features.

The second part of this thesis studies cases where common assumptions that enable causal estimation are violated. We provide an algorithm to estimate causal effects with deep models from confounded data where instrumental variables are available. This algorithm generalizes the control function method and works without the separability assumptions required by popular algorithms like two-stage least-squares and generalized method of moments. Then, we consider tasks where the confounders are known to equal a function of the variables whose effects we want to estimate; this setup violates an assumption known as overlap or positivity, commonly made to uniquely determine (identify) causal effects from non-randomized data. In this setting, we derive nonparametric conditions for identifiability and derive an estimator that solves a gradient flow equation to answer general causal queries from the data without overlap.

The last part of this thesis performs an empirical study of building and transporting CHD risk models between two large hospitals. Departing from the standard approach of constructing risk scores from carefully chosen features, we use broad feature sets available in the electronic health records (EHRs). We train AI models to predict time-to-CHD from minimally curated EHR data that outperform existing risk scores both at the institution where they were trained and when transported externally.

Contents

List of Figures

List of Tables

List of Appendices

1 Introduction

Prediction is fundamental to making better decisions. Consider sepsis — a life-threatening reaction to infection — which is responsible for at least 350,000 deaths each year[*]. With the aid of a model that predicts if a patient is at risk of sepsis, care providers acted to reduce sepsis-related mortality by 17% [1]. At its core, prediction requires on extracting correlations between a target of interest and given inputs or covariates. Advances in supervised learning and pre-training make it easy to extract complex input-output correlations from data. The modern recipe is simple: gradient-based optimization of a loss function with an appropriately chosen model class such as, for example, residual neural networks or transformers.

Trouble begins when it is no longer suitable to assume that the data on which a model is built comes from the same data generating process as the data on which the model is to be used. For example, while one can train a transformer to correlate a patient's medical history to their chances of developing coronary heart disease (CHD), the goal would then be to estimate risks on populations elsewhere or in the future. The challenge is that correlations that the model picked up can shift between training and test times or capture non-causal relationships. Predictions based on unstable relationships can degrade outside the training distribution and basing treatment decisions on non-causal relationships can result in harm. In this thesis, we develop methodology for generalizing out-of-distribution (OOD) and estimating causal effects. We then perform an em-

[*] https://www.nigms.nih.gov/education/fact-sheets/Pages/sepsis.aspx

pirical study in building and transporting CHD risk models at two large hospital systems. The settings vary, sometimes subtly, across chapters and so we defer their descriptions to the individual chapters.

The first part of this thesis tackles the problem of OOD generalization. Chapter 2 considers spurious correlations that are induced by a changing relationship between the label and a nuisance variable that is also correlated with the covariates. For example, in classifying animals in natural images, the background, which is a nuisance, can predict the type of animal. This nuisance-label relationship does not always hold, and the performance of a model trained under one such relationship may be poor on data with a different nuisance-label relationship.

To build predictive models that perform well regardless of the nuisance-label relationship, we develop Nuisance-Randomized Distillation (NuRD). We introduce the nuisance-randomized distribution, a distribution where the nuisance and the label are independent. Under this distribution, we define the set of representations such that conditioning on any member, the nuisance and the label remain independent. We prove that the representations in this set always perform better than chance, while representations outside of this set may not. NuRD finds a representation from this set that is most informative of the label under the nuisance-randomized distribution, and we prove that this representation achieves the highest performance regardless of the nuisance-label relationship. We evaluate NuRD on several tasks including chest X-ray classification where, using non-lung patches as the nuisance, NuRD produces models that predict pneumonia under strong spurious correlations.

Chapter 3 studies how knowledge about features that are related to the label in the same way across different settings for that task, called semantic features or semantics, can help the process of building robust models. Features with varying relationships to the label, like the background in the animal classification task, are nuisances. Models that exploit nuisance-label relationships face performance degradation when these relationships change. Building models robust to such changes requires additional knowledge beyond samples of the features and labels. For example, existing work uses annotations of nuisances or assumes ERM-trained models depend on nuisances. Approaches to integrate new kinds of additional knowledge enlarge the settings where robust models can be built.

We develop an approach to use knowledge about the semantics by corrupting them in data, and then using the corrupted data to produce models which identify correlations between nuisances and the label. Once these correlations are identified, they can be used to adjust for where nuisances drive predictions. We study semantic corruptions in powering different spurious-correlation avoiding methods on multiple OOD tasks like classifying waterbirds, NLI, and detecting cardiomegaly in chest X-rays.

Chapter 4 studies the problem of shortcut learning from the lens of implicit biases in training. Common explanations for shortcut learning assume that the shortcut improves prediction under the training distribution but not in the test distribution. Thus, models trained via the typical gradient-based optimization of cross-entropy, which we call default-ERM, utilize the shortcut. However, even when the stable feature determines the label in the training distribution and the shortcut does not provide any additional information, like in perception tasks, default-ERM still exhibits shortcut learning.

Why are such solutions preferred when the loss for default-ERM can be driven to zero using the stable feature alone? By studying a linear perception task, we show that default-ERM's preference for maximizing the margin leads to models that depend more on the shortcut than the stable feature, even without overparameterization. This insight suggests that default-ERM's implicit inductive bias towards max-margin is unsuitable for perception tasks. Instead, we develop an inductive bias toward uniform margins and show that this bias guarantees dependence only on the perfect stable feature in the linear perception task. We develop loss functions that encourage uniform-margin solutions, called MARG-CTRL. MARG-CTRL mitigates shortcut learning on a variety of vision and language tasks, showing that better inductive biases can remove the need for expensive two-stage shortcut-mitigating methods in perception tasks.

The second part of this thesis studies cases where common assumptions that enable causal estimation are violated. Chapter 5 considers causal estimation using external sources of randomness that only influence the treatment called IVs. We study variables constructed from treatment and IV that help estimate effects, called control functions. We characterize general control functions for effect estimation in a meta-identification result. Then, we show that structural assumptions on the treatment process allow the construction of general control functions, thereby guaranteeing identification. To construct general control functions and estimate effects, we develop the general control function method (GCFN).

GCFN's first stage called variational decoupling (VDE) constructs general control functions by recovering the residual variation in the treatment given the IV. Using VDE's control function, GCFN's second stage estimates effects via regression. Further, we develop semi-supervised GCFN to construct general control functions using subsets of data that have both IV and confounders observed as supervision; this needs no structural treatment process assumptions. We evaluate GCFN on low and high dimensional simulated data and on recovering the causal effect of slave export on modern community trust [2].

Chapter 6 considers tasks where the confounders are known to equal a function of the very variables whose effects we want to estimate. Causal inference relies on two fundamental assumptions: *ignorability* and *positivity*. We study causal inference when the true confounder value can be expressed as a function of the observed data; we call this setting *EFC*. In this setting ignorability is satisfied, however positivity is violated, and causal inference is impossible in general.

We consider two scenarios where causal effects are estimable. First, we discuss interventions on a part of the treatment called *functional interventions* and a sufficient condition for effect estimation of these interventions called *functional positivity*. Second, we develop conditions for nonparametric effect estimation based on the gradient fields of the functional confounder and the true outcome function. To estimate effects under these conditions, we develop LODE. Further, we prove error bounds on LODE's effect estimates, evaluate our methods on simulated and real data, and empirically demonstrate the value of EFC.

The third part of this thesis, chapter 7 covers an empirical study in transporting risk models for coronary heart disease (CHD). Accurate risk models for CVD improve primary prevention of the disease. Unlike the standard approach of constructing risk scores from carefully chosen features, training flexible survival models built on data from rich sources, like the EHR, has lead to better risk models in populations on which they were trained. However, EHR data can shift and exhibit variability due to changing hospital processes necessitates checking the *transportability* of a model, that is whether it remains valid at external institutions and robust to possible data shifts.

We train transformer-based models to predict time-to-CHD on from minimally curated electronic health record (EHR) data that outperform existing risk scores both at the institution they were trained on and when transported externally. We find that the patient's diagnoses and demographic information to be key features in achieving high internal performance and transportability, while including measurements may hurt transportability. Comparing the models' external performance against that of an externally trained on, we find that variation in external performance across demographic subgroups is driven by the differences between the data within the subgroups rather than disproportionate model transport across subgroups.

The papers covered in this thesis are listed here in the order they appear in the chapters:

1. OOD Generalization in the Presence of Nuisance-Induced Spurious Correlations

   Aahlad Puli, Lily Zhang, Eric Oermann, Rajesh Ranganath.

   ICLR, 2022

2. Nuisances via Negativa: Adjusting for Spurious Correlations via Data Augmentation

   Aahlad Puli , Nitish Joshi, Yoav Wald, He He, Rajesh Ranganath

   TMLR, June 2024

3. Don't blame Dataset Shift! Shortcut Learning due to Gradients and Cross Entropy

   Aahlad Puli, Lily Zhang, Yoav Wald, Rajesh Ranganath

   NeurIPS, 2023

4. Causal Estimation with Functional Confounders

   Aahlad Puli, Adler J Perotte, Rajesh Ranganath.

   NeurIPS, 2020.

5. General Control Functions for Causal Effect Estimation from IVs

   Aahlad Puli, Rajesh Ranganath.

   NeurIPS, 2020.

6. Performant and Transportable modeling of CHD risk from minimally curated Hospital-Scale EHRs

   Shreyas Bhave, **Aahlad Puli**, Mark Goldstein, Mert Ketenci, Noémie Elhadad, Adler Perotte, and Rajesh Ranganath

   Working paper.

Part I

Out-of-distribution Generalization

2 Out-of-distribution generalization in the presence of nuisance-induced spurious correlations

2.1 Introduction

Spurious correlations are relationships between the label and the covariates that are prone to change between training and test distributions [3]. Predictive models that exploit spurious correlations can perform worse than even predicting without covariates on the test distribution [4]. Discovering spurious correlations requires more than the training distribution because any single distribution has a fixed label-covariate relationship. Often, spurious correlations are discovered by noticing different relationships across multiple distributions between the label and *nuisance* factors correlated with the covariates. We call these *nuisance-induced spurious correlations.*

For example, in classifying cows vs. penguins, typical images have cows appear on grasslands and penguins appear near snow, their respective natural habitats [4, 5], but these animals can be photographed outside their habitats. In classifying hair color from celebrity faces on CelebA [6], gender is correlated with the hair color. This relationship may not hold in different countries [7]. In language, sentiment of a movie review determines the types of words used in the review to convey attitudes and opinions. However, directors' names appear in the reviews and are correlated with positive sentiment in time periods where directors make movies that are well-liked [8]. In X-ray classification, conditions like pneumonia are spuriously correlated with non-physiological traits of X-ray images due to the association between the label and hospital X-ray collection protocols [9]. Such factors are rarely recorded in datasets but produce subtle differences in X-ray images that convolutional networks easily learn [10].

We formalize nuisance-induced spurious correlations in a nuisance-varying family of distributions where any two distributions are different only due to the differences in the nuisance-label relationship. As the nuisance is informative of the label, predictive models exploit the nuisance-label relationship to achieve the best performance on any single member of the family. However, predictive models that perform best on one member can perform even worse than predicting without any covariates on another member, which may be out-of-distribution (OOD). We develop NuRD to use data collected under one nuisance-label relationship to build predictive models that perform well on other members of the family regardless of the nuisance-label relationship in that member. Specifically, NuRD estimates a conditional distribution which has OOD generalization guarantees across the nuisance-varying family.

In section 2.2, we motivate and develop ideas that help guarantee performance on every member of the family. The first is the *nuisance-randomized distribution*: a distribution where the nuisance is independent of the label. An example is the distribution where cows and penguins have equal chances of appearing on backgrounds of grass or snow. The second is an *uncorrelating representation*: a representation of the covariates such that under the nuisance-randomized distribution, the nuisance remains independent of the label after conditioning on the representation. The set of such representations is the *uncorrelating set*. We show that the nuisance-randomized conditional of the label given an uncorrelating representation has performance guarantees: such conditionals perform as well or better than predicting without covariates on every member in the family while other conditionals may not. Within the uncorrelating set, we characterize one that is optimal on every member of the nuisance-varying family *simultaneously*. We then prove that the same optimal performance can be realized by uncorrelating representations that are most informative of the label under the nuisance-randomized distribution.

Following the insights in section 2.2, we develop Nuisance-Randomized Distillation (NuRD) in section 2.3. NuRD finds an uncorrelating representation that is maximally informative of the label under the nuisance-randomized distribution. NuRD's first step, *nuisance-randomization*, breaks the nuisance-label dependence to produce nuisance-randomized data. We provide two nuisance randomization methods based on generative models and reweighting. The second step, *distillation*, maximizes the information a representation has with the label on the nuisance-randomized data over the uncorrelating set. We evaluate NuRD on class-conditional Gaussians, labeling colored MNIST images [4], distinguishing waterbirds from landbirds, and classifying chest X-rays. In the latter, using the non-lung patches as the nuisance, NuRD produces models that predict pneumonia under strong spurious correlations.

2.2 Nuisance-Randomization and Uncorrelating Sets

We formalize nuisance-induced spurious correlations via a family of data generating processes. Let $\mathbf{y}$ be the label, $\mathbf{z}$ be the nuisance, and $\mathbf{x}$ be the covariates (i.e. features). The family consists of distributions where the only difference in the members of the family comes from the difference in their nuisance-label relationships. Let D index a family of distributions $\mathcal{F} = \{p_D\}_D$; a member p_D in the nuisance-varying family of distributions $\mathcal{F}$ takes the following form:

$$p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y}) p_D(\mathbf{z} \mid \mathbf{y}) p(\mathbf{x} \mid \mathbf{z}, \mathbf{y}), \tag{2.1}$$

where $p_D(\mathbf{z} \mid \mathbf{y})$ is positive and bounded for any $\mathbf{y}$ where $p(\mathbf{y}) > 0$ and any $\mathbf{z}$ in the family's nuisance space $S_{\mathcal{F}}$. This family is called the nuisance-varying family. Due to changing nuisance-label relationships in this family, the conditional distribution of the label $\mathbf{y}$ given the covariates $\mathbf{x}$ in one member, e.g. the training distribution, can perform worse than predicting without covariates on another member of the family, e.g. a test distribution with a different nuisance-label relationship. We define performance of a model $\hat{p}(\mathbf{y} \mid \mathbf{x})$ on a distribution p_{te} as the negative expected KL-divergence from the true conditional $p_{te}(\mathbf{y} \mid \mathbf{x})$:

$$\texttt{Perf}_{p_{te}}(\hat{p}(\mathbf{y} \mid \mathbf{x})) = -\mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel \hat{p}(\mathbf{y} \mid \mathbf{x})\right].$$

Higher is better. This performance equals the expected log-likelihood up to a constant, $C_{p_{te}} = \mathbf{H}_{p_{te}}(\mathbf{y} \mid \mathbf{x})$, that only depends on the p_{te}:

$$\texttt{Perf}_{p_{te}}(\hat{p}(\mathbf{y} \mid \mathbf{x})) = \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \hat{p}(\mathbf{y} \mid \mathbf{x}) + C_{p_{te}}.$$

Consider the following example family $\{q_a\}_{a \in \mathbb{R}}$:

$$\mathbf{y} \sim \mathcal{N}(0, 1) \quad \mathbf{z} \sim \mathcal{N}(a\mathbf{y}, 0.5) \quad \mathbf{x} = [\mathbf{x}_1 \sim \mathcal{N}(\mathbf{y} - \mathbf{z}, 1.5), \mathbf{x}_2 \sim \mathcal{N}(\mathbf{y} + \mathbf{z}, 0.5)]. \tag{2.2}$$

Given training distribution $p_{tr} = q_1$ and test distribution $p_{te} = q_{-1}$, the conditional $p_{tr}(\mathbf{y} \mid \mathbf{x})$ performs even worse than predicting without covariates, $\texttt{Perf}_{p_{te}}(p(\mathbf{y})) \geq \texttt{Perf}_{p_{te}}(p_{tr}(\mathbf{y} \mid \mathbf{x}))$; see appendix A.1.9 for the proof. The problem is that $p_{tr}(\mathbf{y} \mid \mathbf{x})$ utilizes label-covariate relationships that do not hold when the nuisance-label relationships change. When the changing nuisance-label relationship makes the conditional $p_D(\mathbf{y} \mid \mathbf{x})$ of one member unsuitable for another $p'_D \in \mathcal{F}$, the family exhibits *nuisance-induced spurious correlations.*

Next, we identify a conditional distribution with performance guarantees across all members of the family. We develop two concepts to guarantee performance on every member of the nuisance-varying family: the nuisance-randomized distribution and uncorrelating representations.

Definition 1. *The **nuisance-randomized distribution** is* $p_{\perp\!\!\!\perp}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) p_{tr}(\mathbf{z}) p(\mathbf{y})$.[*]

In the cows vs. penguins example, $p_{\perp\!\!\!\perp}$ is the distribution where either animal has an equal chance to appear on backgrounds of grass or snow. The motivation behind the nuisance-randomized distribution is that when the nuisance is independent of the label, (noisy[†]) functions of only the nuisance are not predictive of the label. If the covariates only consist of (noisy) functions of either the nuisance or the label but never a mix of the two (an example of mixing is $\mathbf{x}_1 = \mathbf{y} - \mathbf{z} + \textit{noise}$), then the conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ does not vary with the parts of $\mathbf{x}$ that are (noisy) functions of just the nuisance. Thus, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ ignores the features which have changing relationships with the label.

How about nuisance-varying families where the covariates contain functions that mix the label and the nuisance? Equation (2.2) is one such family, where the covariates $\mathbf{x}_1$ and $\mathbf{x}_2$ are functions of both the label and the nuisance. In such nuisance-varying families, the conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ can use functions that mix the label and the nuisance even though the nuisance is not predictive of the label by itself. These mixed functions have relationships with the label which change across the family; for example in eq. (2.2), the coordinate $\mathbf{x}_1$ is correlated positively with the label under q_0 but negatively under q_{-2}. Then, under changes in the nuisance-label relationship, the conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ can perform worse than predicting without covariates because it utilizes a relationship, via these mixed features, that no longer holds. See appendix A.1.9 for details.

We address this performance degradation of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ by introducing representations that help avoid reliance on functions that mix the label and the nuisance. We note that when the conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ uses functions that mix the label and the nuisance, knowing the exact value of the

[*] Different marginal distributions $p_{\perp\!\!\!\perp}(\mathbf{z})$ produce different distributions where the label and nuisance are independent. The results are insensitive to the choice as long as $p_{\perp\!\!\!\perp}(\mathbf{z}) > 0$ for any $\mathbf{z} \in S_{\mathcal{F}}$. One distribution that satisfies this requirement is $p_{\perp\!\!\!\perp}(\mathbf{z}) = p_{tr}(\mathbf{z})$. See lemma 2.

[†] Noisy functions of a variable are functions of that variable and exogenous noise.

nuisance should improve the prediction of the label, i.e. $\mathbf{y} \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{x}$. [‡] Therefore, to avoid reliance on mixed functions, we define *uncorrelating* representations $r(\mathbf{x})$, where the nuisance does not provide any extra information about the label given the representation:

Definition 2. *An uncorrelating set of representations is $\mathcal{R}(p_{\perp\!\!\!\perp})$ s.t. $\forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid r(\mathbf{x})$.*

In the example in eq. (2.2), $r(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$ is an uncorrelating representation because it is purely a function of the label and the noise. Conditional distributions $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ for any uncorrelating $r(\mathbf{x})$ only depend on properties that are shared across all distributions in the nuisance-varying family. Specifically, for $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$, the conditional distribution $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ uses $p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})$ and $p(\mathbf{y})$ which are both shared across all members of the family $\mathcal{F}$. For $\mathbf{z}'$ such that $p_{\perp\!\!\!\perp}(\mathbf{z}' \mid r(\mathbf{x})) > 0$,

$$p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x})) = p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}') = \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{z}')p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}')}{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z}')} = \frac{p(\mathbf{y})p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}')}{\mathbb{E}_{p(\mathbf{y})}p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}')}. \tag{2.3}$$

This fact helps characterize the performance of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ on any member $p_{te} \in \mathcal{F}$:

$$\mathtt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) = \mathtt{Perf}_{p_{te}}(p(\mathbf{y})) + \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \,\|\, \mathbb{E}_{p(\mathbf{y})}p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right]. \tag{2.4}$$

As KL-divergence is non-negative, for any uncorrelating representation r, the conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ does at least as well as predicting without covariates for all members $p_{te} \in \mathcal{F}$: $\mathtt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) \geq \mathtt{Perf}_{p_{te}}(p(\mathbf{y}))$. See appendix A.1.3 for the formal derivation. In fact, we show in appendix A.1.5 that when the identity representation $r(\mathbf{x}) = \mathbf{x}$ is uncorrelating, then $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ is minimax optimal for a family with sufficiently diverse nuisance-label relationships.

Equation (2.4) lower bounds the performance of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ for any representation in the uncor-

[‡] This is because the nuisance is independent of the label under the nuisance-randomized distribution and can be thought of as a source of noise in $p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{y})$; consequently, conditional on $\mathbf{x}$ containing these mixed functions, knowing $\mathbf{z}$ provides extra information about the label by decreasing noise.

relating set across all $p_{te} \in \mathcal{F}$. However, it does not specify which of these representations leads to the best performing conditional. For example, between two uncorrelating representations like the shape of the animal and whether the animal has horns, which predicts better? Next, we characterize uncorrelating representations that are *simultaneously* optimal for all test distributions $p_{te} \in \mathcal{F}$.

Optimal uncorrelating representations. As we focus on nuisance-randomized conditionals, henceforth, by performance of $r(\mathbf{x})$, we mean the performance of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$: $\texttt{Perf}_{p_{te}}(r(\mathbf{x})) = \texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x})))$. Consider two uncorrelating representations r, r_2, where the pair (r, r_2) is also uncorrelating. How can $r_2(\mathbf{x})$ dominate $r(\mathbf{x})$ in performance across the nuisance-varying family? Equation (2.3) shows that

$$p_{\perp\!\!\!\perp}(\mathbf{y} \mid [r(\mathbf{x}), r_2(\mathbf{x})]) \propto p(\mathbf{y})p(r(\mathbf{x}) \mid r_2(\mathbf{x}), \mathbf{y}, \mathbf{z} = z)p(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z} = z).$$

If $r_2(\mathbf{x})$ **blocks** the dependence between the label and $r(\mathbf{x})$, i.e. $r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r_2(\mathbf{x}), \mathbf{z}$, then knowing r does not change the performance when r_2 is known, suggesting that blocking relates to performance. In theorem 1, we show that the *maximally blocking* uncorrelating representation is simultaneously optimal: its performance is as good or better than every other uncorrelating representation on every distribution in the nuisance-varying family. We state the theorem first:

Theorem 1. *Let $r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})$ be **maximally blocking**:*

$$\forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r^*(\mathbf{x}).$$

Then,

1. *(Simultaneous optimality)* $\forall p_{te} \in \mathcal{F}, \forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \texttt{Perf}_{p_{te}}(r^*(\mathbf{x})) \geq \texttt{Perf}_{p_{te}}(r(\mathbf{x}))$.
2. *(Information maximality)* $\forall r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) \geq \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r(\mathbf{x}))$.

3. *(Information maximality implies simultaneous optimality)* $\forall r' \in \mathcal{R}(p_{\perp\!\!\!\perp})$,

$$\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r'(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) \quad \Longrightarrow \quad \forall p_{te} \in \mathcal{F}, \quad \texttt{Perf}_{p_{te}}(r^*(\mathbf{x})) = \texttt{Perf}_{p_{te}}(r'(\mathbf{x})).$$

The proof is in appendix A.1.4. The first part of theorem 1, *simultaneous optimality*, says that a maximally blocking uncorrelating representation r^* dominates every other $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$ in performance on *every* test distribution in the family. In the cows vs. penguins example, the segmented foreground that contains only the animal is a maximally blocking representation because the animal blocks the dependence between the label and any other semantic feature of the animal.

The second and third parts of theorem 1 are useful for algorithm building. The second part proves that a maximally blocking r^* is also maximally informative of the label under $p_{\perp\!\!\!\perp}$, indicating how to find a simultaneously optimal uncorrelating representation. What about other information-maximal uncorrelating representations? The third part shows that if an uncorrelating representation r' has the same mutual information with the label (under $p_{\perp\!\!\!\perp}$) as the maximally blocking r^*, then r' achieves the same simultaneously optimal performance as r^*. In the cows vs. penguins example, an example of a maximally informative uncorrelating representation is the number of legs of the animal because the rest of the body does not give more information about the label. The second and third parts of theorem 1 together show that finding an uncorrelating representation that maximizes information under the nuisance-randomized distribution finds a simultaneously optimal uncorrelating $r(\mathbf{x})$.

2.3 Nuisance-Randomized Distillation (NuRD)

Theorem 1 says a representation that maximizes information with the label under the nuisance-randomized distribution has the best performance within the uncorrelating set. We develop a representation learning algorithm to maximize the mutual information between the label and a representation in the uncorrelating set under the nuisance-randomized distribution. We call this algorithm Nuisance-Randomized Distillation (NuRD). NuRD has two steps. The first step, called nuisance randomization, creates an estimate of the nuisance-randomized distribution. The second step, called distillation, finds a representation in the uncorrelating set with the maximum information with the label under the estimate of the nuisance-randomized distribution from step one.

Nuisance Randomization. We estimate the nuisance-randomized distribution with generative models or by reweighting existing data. Generative-NuRD uses the fact that $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ is the same for each member of the nuisance-varying family $\mathcal{F}$. With an estimate of this conditional denoted $\hat{p}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$, generative-NuRD's estimate of the nuisance-randomized distribution is $\mathbf{z} \sim p_{tr}(\mathbf{z}), \mathbf{y} \sim p(\mathbf{y}), \mathbf{x} \sim \hat{p}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$. For high dimensional $\mathbf{x}$, the estimate $\hat{p}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ can be constructed with deep generative models. Reweighting-NuRD importance weights the data from p_{tr} by $p(\mathbf{y})/p_{tr}(\mathbf{y} \mid \mathbf{z})$, making it match the nuisance-randomized distribution:

$$p_{\perp\!\!\!\perp}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{y})p_{tr}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = p(\mathbf{y})p_{tr}(\mathbf{z})\frac{p_{tr}(\mathbf{y} \mid \mathbf{z})}{p_{tr}(\mathbf{y} \mid \mathbf{z})}p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = \frac{p(\mathbf{y})}{p_{tr}(\mathbf{y} \mid \mathbf{z})}p_{tr}(\mathbf{x}, \mathbf{y}, \mathbf{z}).$$

Reweighting-NuRD uses a model trained on samples from p_{tr} to estimate $\frac{p(\mathbf{y})}{p_{tr}(\mathbf{y} \mid \mathbf{z})}$.

Distillation. Distillation seeks to find the representation in the uncorrelating set that maximizes the information with the label under $\hat{p}_{\perp\!\!\!\perp}$, the estimate of the nuisance-randomized distribution. Maximizing the information translates to maximizing likelihood because the entropy $\mathbf{H}_{\hat{p}_{\perp\!\!\!\perp}}(\mathbf{y})$ is constant with respect to the representation r_γ parameterized by γ:

$$\mathbf{I}_{\hat{p}_{\perp\!\!\!\perp}}(r_\gamma(\mathbf{x}); \mathbf{y}) - \mathbf{H}_{\hat{p}_{\perp\!\!\!\perp}}(\mathbf{y}) = \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, r_\gamma(\mathbf{x}))} \log \hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x})) = \max_\theta \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, r_\gamma(\mathbf{x}))} \log p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x})).$$

Theorem 1 requires the representations be in the uncorrelating set. When conditioning on representations in the uncorrelating set, the nuisance has zero mutual information with the label: $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{z} \mid r_\gamma(\mathbf{x})) = 0$. We operationalize this constraint by adding a conditional mutual information penalty to the maximum likelihood objective with a tunable scalar parameter λ

$$\max_{\theta,\gamma} \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})} \log p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x})) - \lambda \mathbf{I}_{\hat{p}_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{z} \mid r_\gamma(\mathbf{x})). \tag{2.5}$$

The objective in eq. (2.5) can have local optima when the representation is a function of the nuisance and exogenous noise (noise that generates the covariates given the nuisance and the label). The intuition behind these local optima is that the value of introducing information that predicts the label does not exceed the cost of the introduced conditional dependence. Appendix A.1.6 gives a formal discussion and an example with such local optima. Annealing λ, which controls the cost of conditional dependence, can mitigate the local optima issue at the cost of setting annealing schedules.

Instead, we restrict the distillation step to search over representations $r_\gamma(\mathbf{x})$ that are also marginally independent of the nuisance $\mathbf{z}$ under $p_{\perp\!\!\!\perp}$, i.e. $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r_\gamma(\mathbf{x})$. This additional independence removes representations that depend on the nuisance but are not predictive of the label; in turn, this removes local optima that correspond to functions of the nuisance and exogenous noise. In the cows vs. penguins example, representations that are functions of the background only, like the presence of snow, are uncorrelating but do not satisfy the marginal independence. Together, the conditional independence $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid r_\gamma(\mathbf{x})$ and marginal independence $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r_\gamma(\mathbf{x})$ hold if and only if the representation and the label are jointly independent of the nuisance : $(\mathbf{y}, r_\gamma(\mathbf{x})) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$. Using mutual information to penalize joint *dependence* (instead of the penalty in eq. (2.5)), the distillation step in NuRD is

$$\max_{\theta,\gamma} \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})} \log p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x})) - \lambda \mathbf{I}_{\hat{p}_{\perp\!\!\!\perp}}([\mathbf{y}, r_\gamma(\mathbf{x})]; \mathbf{z}). \tag{2.6}$$

We show in lemma 4 that within the set of representations that satisfy joint independence, NuRD learns a representation that is simultaneously optimal in performance on all members of the nuisance-varying family. To learn representations using gradients, the mutual information needs to be estimated in a way that is amenable to gradient optimization. To achieve this, we estimate the mutual information in NuRD via the classification-based density-ratio estimation trick [11]. We use a *critic model* p_ϕ to estimate said density ratio. We describe this technique in appendix A.1.1 for completeness. We implement the distillation step as a bi-level optimization where the outer loop optimizes the predictive model $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ and the inner loop optimizes the critic model p_ϕ which helps estimate the mutual information.

Algorithm. We give the full algorithm boxes for both reweighting-NuRD and generative-NuRD in appendix A.1.1. In reweighting-NuRD, to avoid poor weight estimation due to models memorizing the training data, we use cross-fitting; see algorithm 1. The setup of nuisance-induced spurious correlations in eq. (2.1) assumes $p(\mathbf{y})$ is fixed across distributions within the nuisance-varying family $\mathcal{F}$. This condition can be relaxed when $p_{te}(\mathbf{y})$ is known; see appendix A.1.1.

2.4 Related Work

In table 2.1, we summarize key differences between NuRD and the related work: invariant learning [4, 12], distribution matching [13, 14], shift-stable prediction [15], group-DRO [16], and causal regularization [17, 18]. We detail the differences here.

Nuisance versus Environment. In general, an environment is a distribution with a specific spurious correlation [16]. When the training and test distributions are members of the same nuisance-varying family, environments denote specific nuisance-label relationships. In contrast, nuisances are variables whose changing relationship with the label induces spurious correlations. While obtaining data from diverse environments requires data collection from sufficiently different sources, one can specify nuisances from a single source of data via domain knowledge.

Domain generalization, domain-invariant learning, and subgroup robustness We briefly mention existing methods that aim to generalize to unseen test data and focus on how these methods can suffer in the presence of nuisance-induced spurious correlations; for a more detailed presentation, see appendix A.1.2. Domain generalization and domain-invariant learning methods assume the training data consists of multiple *sufficiently different* environments to generalize to unseen test data that is related to the given environments or subgroups [4, 12, 13, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]. Due to its focus on nuisances, NuRD works with data from a single environment. Taking a distributional robustness [31] approach, Sagawa et al. [16] used group-DRO to build models that perform well on every one of a finite set of known subgroups. Other work also aims to minimize worst subgroup error with a finite number of fixed but unknown subgroups [32, 33]; as subgroups are unknown, they only find an approximate minimizer of the worst subgroup error in general even with infinite data. While these methods [32, 33] were developed to enforce fairness with respect to a sensitive attribute, they can be applied to ood generalization with the nuisance treated as the sensitive attribute; see [34]. Given the nuisance, existence of a finite number of subgroups maps to an additional discreteness assumption on the nuisance variable; in contrast, NuRD works with general nuisances. Given a high dimensional $\mathbf{z}$, as in our experiments, defining groups based on the value of the nuisance like in [16] typically results in groups with at most one sample; with the resulting groups, methods that minimize worst subgroup error will encourage memorizing the training data.

Nuisance as the environment label for domain generalization. Domain generalization methods are inapplicable when the training data consists only of a single environment. In this work, the training data comes from only one member of the nuisance-varying family, i.e. from a single environment. What if one treats groups defined by nuisance values as environments?

Table 2.1: NuRD vs. methods that use nuisances or environments. In this work, the training data comes from a single member of the family $\mathcal{F}$, i.e. a single environment. For methods that require multiple environments, values of the nuisance can be treated as environment labels. Unlike existing methods, NuRD works with high-dimensional nuisances without requiring them at test time.

	Invariant	Dist. match	Shift-stable	Group-DRO	Causal reg.	NuRD
High-dim **z**	✗	✗	✓	✗	✗	✓
No test-time **z**	✓	✓	✗	✓	✓	✓

Using the nuisance as the environment label can produce non-overlapping supports (over the covariates) between environments. In Colored-MNIST for example, splitting based on color produces an environment of green images and an environment of red images. When the covariates do not overlap between environments, methods such as Arjovsky et al. [4], Krueger et al. [12] will not produce invariant representations because the model can segment out the covariate space and learn separate functions for each environment.

Methods based on conditional distribution matching [13, 14] build representations that are conditionally independent of the environment variable given the label. When the training data is split into groups based on the nuisance value, representations built by these methods are independent of the nuisance given the label. However, splitting on a high-dimensional nuisance like image patches tends to yield many groups with only a single image. Matching distributions of representations for the same label across all environments is not possible when some environments only have one label.

Causal learning and shift-stable prediction. Anticausal learning [35] assumes a causal generative process for a class of distributions like the nuisance-varying family in eq. (2.1). In such an interpretation, the label $\mathbf{y}$ and nuisance $\mathbf{z}$ cause the image $\mathbf{x}$, and under independence of cause and mechanism, $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ is fixed — in other words, independent — regardless of the distribution $p_D(\mathbf{y}, \mathbf{z})$. A closely related idea to NuRD is that of Shift-Stable Prediction [15, 36]. Subbaswamy et al. [15] perform *graph surgery* to learn $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$ assuming access to $\mathbf{z}$ during test time. Shift-stable models are not applicable without nuisances at test time while NuRD only requires nuisances during training. However, if the nuisance is available at test time, the combined covariate-set $[\mathbf{x}, \mathbf{z}]$ is 1) uncorrelating because $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid [\mathbf{x}, \mathbf{z}]$ and 2) maximally blocking because $r([\mathbf{x}, \mathbf{z}]) \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}, [\mathbf{x}, \mathbf{z}]$, and theorem 1 says $p_{\perp\!\!\!\perp}(\mathbf{y} \mid [\mathbf{x}, \mathbf{z}])$ is optimal.

Two concurrent works also build models using the idea of a nuisance variable. Makar et al. [18] assume that there exists a stochastic function of $\mathbf{y}$ but not $\mathbf{z}$, called $\mathbf{x}^*$, such that $\mathbf{y} \perp\!\!\!\perp_{p_D} \mathbf{x} \mid \mathbf{x}^*, \mathbf{z}$; they use a marginal independence penalty $r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$. Veitch et al. [17] use counterfactual invariance to derive a conditional independence penalty $r(\mathbf{x}) \perp\!\!\!\perp_{p_{tr}} \mathbf{z} \mid \mathbf{y}$. The theory in these works requires the nuisance to be discrete, and their algorithms require the nuisance to be both discrete and low-cardinality; NuRD and its theory work with general high-dimensional nuisances. Counterfactual invariance promises that a representation will not vary with the nuisance but it does not produce optimal models in general because it rejects models that depend on functions of the nuisance. On the other hand, the uncorrelating property allows using functions of only the nuisance that are in the covariates to extract further information about the label from rest of the covariates; this leads to better performance in some nuisance-varying families, as we show using the theory of minimal sufficient statistics [37] in appendix A.1.7.

2.5 Experiments

We evaluate the implementations of NuRD on class-conditional Gaussians, Colored-MNIST [4], Waterbirds [16], and chest X-rays [38, 39]. See appendix A.2 for implementation details and further evaluations of NuRD.

Model selection, baselines, and metrics. Models in both steps of NuRD are selected using heldout subsets of the training data. We split the training data into training and validation datasets with an $80-20$ split. For nuisance-randomization, this selection uses standard measures

Table 2.2: Accuracy of NuRD versus ERM on class conditional Gaussians.

Method	Heldout p_{tr}	Heldout $\hat{p}_{\perp\!\!\!\perp}$	p_{te}
ERM	84 ± 0%	–	39 ± 0%
generative-NuRD	71 ± 0%	67 ± 0%	58 ± 0%
reweighting-NuRD	71 ± 1%	66 ± 0%	58 ± 0%

of held-out performance. Selection in the distillation step picks models that give the best value of the distillation objective on a held out subset of the *nuisance-randomized* data from NuRD's first step.

We compare against ERM because, as discussed in section 2.4, existing methods that aim to generalize under spurious correlations require assumptions, such as access to multiple environments or discrete nuisance of small cardinality, that do not hold in the experiments. When possible, we report the oracle accuracy or build gold standard models using data that does not have a nuisance-label relationship to exploit. For every method, we report the average accuracy and standard error across 10 runs each with a different random seed. We report the accuracy of each model for each experiment on the test data (p_{te}) and on heldout subsets of the original training data (p_{tr}) and the estimate of the nuisance-randomized distribution ($\hat{p}_{\perp\!\!\!\perp}$). For all experiments, we use $\lambda = 1$ and one or two epochs of critic model updates for every predictive model update.

2.5.1 Class-Conditional Gaussians

We generate data as follows: with $\mathcal{B}(0.5)$ as the uniform Bernoulli distribution, $q_a(\mathbf{y}, \mathbf{z}, \mathbf{x})$ is

$$\mathbf{y} \sim \mathcal{B}(0.5) \quad \mathbf{z} \sim \mathcal{N}(a(2\mathbf{y} - 1), 1) \quad \mathbf{x} = [\mathbf{x}_1 \sim \mathcal{N}(\mathbf{y} - \mathbf{z}, 9), \mathbf{x}_2 \sim \mathcal{N}(\mathbf{y} + \mathbf{z}, 0.01)]. \tag{2.7}$$

The training and test sets consist of 10000 samples from $p_{tr} = q_{0.5}$ and 2000 samples from $p_{te} = q_{-0.9}$ respectively. All models in both NuRD methods are parameterized with neural networks.

Table 2.3: Accuracy of NuRD versus ERM on Colored-MNIST. The oracle accuracy is 75%.

Method	Heldout p_{tr}	Heldout $\hat{p}_{\perp\!\!\!\perp}$	p_{te}
ERM	90 ± 0%	–	10 ± 0%
generative-NuRD	73 ± 1%	80 ± 1%	68 ± 2%
reweighting-NuRD	75 ± 0%	74 ± 0%	75 ± 0%

Results. Table 2.2 reports results. The test accuracy of predicting with the optimal linear uncorrelating representation $r^*(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$, is 62%; appendix A.2 gives the optimality proof. Both generative-NuRD and reweighting-NuRD achieve close to this accuracy.

2.5.2 Colored-MNIST

We construct a colored-MNIST dataset [3, 4] with images of 0s and 1s. In this dataset, the values in each channel for every pixel are either 0 or 1. We construct two environments and use one as the training distribution and the other as the test. In training, 90% of red images have label 0; 90% of green images have label 1. In test, the relationship is flipped: 90% of the 0s are green, and 90% of the 1s are red. In both training and test, the digit determines the label only 75% of the time, meaning that exploiting the nuisance-label relationship produces better training accuracies. The training and test data consist of 4851 and 4945 samples respectively. We run NuRD with the most intense pixel as the nuisance.

Results. See table 2.3 for the results. ERM learns to use color, evidenced by the fact that it achieves a test accuracy of only 10%. The oracle accuracy of 75% is the highest achievable by models that do not use color because the digit only predicts the label with 75% accuracy. While generative-NuRD has an average accuracy close to the oracle, reweighting-NuRD matches the oracle at 75%.

Method	p_{tr}	$\hat{p}_{\perp\!\!\!\perp}$	p_{te}
ERM	91 ± 0%	–	66 ± 2%
reweighting-NuRD	85 ± 1%	81 ± 1%	83 ± 2%

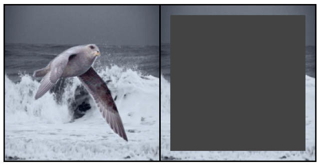


Figure 2.1: Table of results and figure showing an example of the nuisance for Waterbirds. On the left are the accuracies of NuRD and ERM on Waterbirds. Gold standard accuracy is 90% (see the results paragraph below). The figure shows an image and the corresponding border nuisance. NuRD has both images during training and only the left image at test time.

2.5.3 Learning to classify Waterbirds and Landbirds

Sagawa et al. [16] consider the task of detecting the type of bird (water or land) from images where the background is a nuisance. Unlike Sagawa et al. [16], we do not assume access to validation and test sets with independence between the background and the label. So, we split their dataset differently to create our own training and test datasets with substantially different nuisance-label relationships. The training data has 90% waterbirds on backgrounds with water and 90% landbirds on backgrounds with land. The test data has this relationship flipped. We use the original image size of $224 \times 224 \times 3$. The training and test sets consist of 3510 and 400 samples respectively. We ensure that $p(\mathbf{y} = 1) = 0.5$ in training and test data. Thus, predicting the most frequent class achieves an accuracy of 0.5. Cropping out the whole background requires manual segmentation. Instead, we use the pixels outside the central patch of 196×196 pixels as a nuisance in NuRD. This is a high-dimensional nuisance which impacts many existing methods negatively; see section 2.4. The covariates are the whole image; see fig. 2.1.

Results. Figure 2.1 reports results. We construct a gold standard model on data where waterbirds and landbirds have equal chances of appearing on backgrounds with water and land; this model achieves a test accuracy of 90%. ERM uses the background to predict the label, as evidenced by its test accuracy of 66%. Reweighting-NuRD uses the background patches to adjust for the spurious correlation to achieve an average accuracy close to the gold standard, 83%. We do not report generative-NuRD's performance as training on the generated images resulted in classifiers that predict as poorly as chance on real images. This may be due to the small training dataset.

2.5.4 Learning to label pneumonia from X-rays

In many problems such as classifying cows versus penguins in natural images, the background, which is a nuisance, predicts the label. Medical imaging datasets have a similar property, where factors like the device used to take the measurement are predictive of the label but also leave a signature on the whole image. Here, we construct a dataset by mixing two chest x-ray datasets, CheXpert and MIMIC, that have different factors that affect the whole image, with or without pneumonia. The training data has 90% pneumonia images from MIMIC and 90% healthy images from CheXpert. The test data has the flipped relationship, with 90% of the pneumonia images from CheXpert and 90% of the healthy images from MIMIC. We resize the X-ray images to 32 × 32. Healthy cases are downsampled to make sure that in the training and test sets, healthy and pneumonia cases are equally probable. Thus, predicting the most frequent class achieves an accuracy of 0.5. The training and test datasets consist of 12446 and 400 samples respectively. In chest X-rays, image segmentation cannot remove all the nuisances because nuisances like scanners alter the entire image [9, 10, 40]. However, non-lung patches, i.e. pixels outside the central patches which contain the lungs, are *a nuisance* because they do not contain physiological signals of pneumonia. We use the non-lung patches (4-pixel border) as a nuisance in NuRD. This is a high-dimensional nuisance which impacts existing methods negatively; see section 2.4. The covariates are the whole image; see fig. 2.2.

Results. Figure 2.2 reports results. Building an oracle model in this experiment requires knowledge of all factors that correlate the label with all the parts of the X-ray. Such factors also exist within each hospital but are not recorded in MIMIC and CheXpert; for example, different departments in the same hospital can have different scanners which correlate the non-lung patches of the X-ray with the label [9]. erm uses the nuisance to predict pneumonia, as evidenced by its

Method	p_{tr}	$\hat{p}_{\perp\!\!\!\perp}$	p_{te}
ERM	89 ± 0%	–	37 ± 1%
generative-NURD	70 ± 3%	90 ± 2	41 ± 2%
reweighting-NuRD	75 ± 1%	68 ± 1%	61 ± 1%

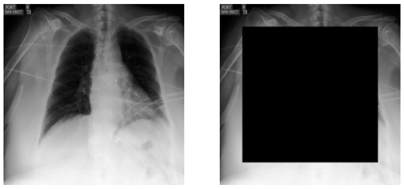


Figure 2.2: Table of results and figure showing an example of the nuisance for chest X-rays. The figure shows an example of a chest X-ray and the corresponding non-lung patches (right). NuRD has both images during training and only the left image at test time.

test accuracy of 37%. Reweighting-NuRD uses the non-lung patches to adjust for the spurious correlation and achieves an accuracy of 61%, a large improvement over ERM.

Generative-NuRD also outperforms ERM's performance on average. Unlike reweighting-NuRD which outperforms predicting without covariates, generative-NuRD performs similar to predicting without covariates on average. The few poor test accuracies may be due to two ways generative nuisance-randomization can be imperfect: 1) little reliance of $\mathbf{x}$ on $\mathbf{z}$ with $\mathbf{y}$ fixed, 2) insufficient quality of generation which leads to poor generalization from generated to real images.

2.6 Discussion

We develop an algorithm for OOD generalization in the presence of spurious correlations induced by a nuisance variable. We formalize nuisance-induced spurious correlations in a nuisance-varying family, where changing nuisance-label relationships make predictive models built from samples of one member unsuitable for other members. To identify conditional distributions that have performance guarantees on all members of the nuisance-varying family, we introduce the nuisance-randomized distribution and uncorrelating representations. We characterize one uncorrelating representation that is *simultaneously* optimal for all members. Then, we show that uncorrelating representations most informative of the label under the nuisance-randomized distribution also achieve the same optimal performance. Following this result, we propose to estimate the nuisance-randomized distribution and, under this distribution, construct the uncorrelating representation that is most informative of the label. We develop an algorithm called NuRD and show that it outperforms ERM on synthetic and real data by adjusting for nuisance-induced spurious correlations. Our experiments show that NuRD can use easy-to-acquire nuisances (like the border of an image) to do this adjustment; therefore, our work suggests that the need for expensive manual segmentation, even if it does help exclude all the nuisances, could be mitigated.

LIMITATIONS AND THE FUTURE. Given groups based on pairs of nuisance-label values, Sagawa et al. [7] suggest that subsampling equally from each group produces models more robust to spurious correlations than reweighting [16, 41]; however, subsampling is ineffective when the nuisance is high-dimensional. Instead, as sufficient statistics of the conditional $p_{tr}(\mathbf{y} \mid \mathbf{z})$ render $\mathbf{y}, \mathbf{z}$ independent, grouping based on values of sufficient statistics could be promising. The nuisance-randomization steps in generative-NuRD and reweighting-NuRD model different distributions in the training distribution: $p_{tr}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ and $p_{tr}(\mathbf{y} \mid \mathbf{z})$ respectively. Methods that combine the two approaches to produce better estimates of the nuisance-randomized distribution would be interesting. The first step in reweighting-NuRD is to estimate $p_{tr}(\mathbf{y} \mid \mathbf{z})$. As deep networks tend to produce inflated probabilities [42], one must take care to build calibrated models for $p(\mathbf{y} \mid \mathbf{z})$. Adapting either calibration-focused losses [43, 44] or ensembling [45] may produce calibrated probabilities.

In our experiments, the training data contains a single environment. Methods for invariant representation learning [4, 12, 13, 14] typically require data from multiple different environments. Nuisance-randomized data has a different nuisance-label relationship from the training data, meaning it is a different environment from the training data. Following this insight, using nuisance-randomization to produce samples from different environments using data from only a single environment would a fruitful direction. The absolute performance for both ERM which exploits spurious correlations and NuRD which does not, is too low to be of use in the clinic. Absolute performance could be improved with larger models, more data, using pretrained models, and multi-task learning over multiple lung conditions, all techniques that could be incorporated into learning procedures in general, including NuRD.

3 | Nuisances via Negativa: Adjusting for Spurious Correlations via Data Augmentation

3.1 Introduction

Relationships between the label and the covariates can change across data collected at different places and times. For example, in classifying animals, data collected in natural habitats have cows appear more often on grasslands, while penguins appear more often on backgrounds of snow; these animal-background relationships do not hold outside natural habitats [4, 46]. Some features, like an animal's shape, are predictive of the label across all settings for a task; these are *semantic features*, or *semantics* in short. Other features with varying relationships with the label, like the background, are nuisances. Even with semantics present, models trained via ERM can predict using nuisances and thus fail to generalize [5]. Models that rely only on the semantic features perform well even when the nuisance-label relationship changes, unlike models that rely on nuisances.

Building models that generalize under changing nuisance-label relationships requires additional knowledge, beyond a dataset of features and labels sampled from the training distribution. For example, many works assume knowledge of the nuisance. In the animal-background example, this would correspond to a feature that specifies the image background, which we may use when specifying our learning algorithm. [17, 18, 47, 48]; another common type of assumption is access to multiple datasets over which the nuisance-label correlation varies [4, 24, 49], and some other forms of knowledge have been explored [50, 51, 52].

Semantic Corruptions. In this paper, we explore the use of a different type of knowledge: corruptions of semantic features. Intuitively, imagine trying to predict the label from a corrupted input $T(\mathbf{x})$, where all semantic information has been removed. Any better-than-chance prediction provides us a window into the nuisances, as it must rely on them. We will then use these obtained biased models to guide methods that we identify here as biased-model-based spurious-correlation avoiding methods (B-SCAMS).

B-SCAMS. There is a class of methods in the literature that use predictions of a biased model to adjust for nuisances, and learn predictors that are free of spurious correlations. Among others, these include JTT [53], EILL [34], NuRD [48], and DFL, POE [47]. The key question arising from these works is *how can we obtain biased models?* In empirical studies, prior works on B-SCAMS either use annotations of the nuisance or an ERM-trained model over the training data as a placeholder for the biased model. The latter approach, based on an ERM-trained model, is successful if that model completely ignores semantic information. In practice, these heuristics are rather fragile. Annotations for nuisances are seldom available, and we lack a principled method to ascertain whether a model trained with ERM relies only on semantic features. Therefore, employing semantic corruptions could serve as a valuable alternative to these heuristics. We claim that semantic corruptions offer a principled and useful approach to obtaining biased models.

Semantic corruptions $T(\mathbf{x})$ must strike a delicate balance between removing semantic information and preserving nuisances. For example, if $T(\mathbf{x})$ replaces all pixels in an image with random noise, it corrupts semantics while simultaneously erasing all information about the nuisances. An ideal

$T(\mathbf{x})$ would isolate nuisances by targeting only the semantic information in the input, e.g., by in-painting the animal for the task of classifying cows and penguins. Implementing such ideal corruptions is unrealistic, as they are task-specific and may require human annotations of the semantic features; e.g., one can segment the objects in every image. Doing so for all classification problems is extremely laborious. In tasks like NLI, it is unclear even *how* to annotate semantics, as they do not correspond to simple features like subsets of words. In summary, after outlining the desired characteristics of semantic corruptions, we define corruptions that are beneficial across multiple tasks and do not require human annotation. Our contributions are as follows:

1. Show that acquiring additional knowledge beyond a labeled dataset is necessary for effectively learning robust models (theorem 2). Then, in proposition 1, we formalize sufficient conditions under which additional knowledge in the form of a semantic corruption enables B-SCAMs to learn robust models.

2. Develop multiple semantic corruptions for object recognition and natural language inference. These include patch randomization, n-gram randomization, frequency filtering, and intensity filtering. Then, we situate existing procedures, such as region-of-interest masking and premise masking, under the umbrella of semantic corruptions.

3. Empirically, we demonstrate that any semantic corruption can power any B-SCAM. The corruption-powered versions of these methods outperform ERM on out-of-distribution (OOD) generalization tasks like Waterbirds, cardiomegaly detection from chest X-rays, and NLI. Corruption-powered NuRD, DFL, and POE achieve performance similar to said methods run with extra observed nuisance variables. Corruption-powered JTT outperforms vanilla JTT.

3.2 Biased-model-based spurious-correlation avoiding methods

A spurious correlation is a relationship between the covariates $\mathbf{x}$ and the label $\mathbf{y}$ that changes across settings like time and location [5]. The features whose relationship with the label changes are called nuisances. With a vector of nuisances $\mathbf{z}$, let $p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x}), p_{te}(\mathbf{y}, \mathbf{z}, \mathbf{x})$ be the training and test distributions.

Achieving robustness to spurious correlations requires additional knowledge. In the presence of spurious correlations, the training distribution p_{tr} may not equal the test distribution p_{te}. Without further assumptions, no algorithm that only sees data from $p_{tr}(\mathbf{y}, \mathbf{x})$ can produce a predictor that works well on p_{te}. To achieve generalization when $p_{te} \neq p_{tr}$, work in the ood generalization literature assumes a relationship between the training and test distributions. We follow the work of Makar et al. [18], Puli et al. [48] and assume that only nuisance-label relationships — i.e. the conditional $\mathbf{z} \mid \mathbf{y}$ — changes between training and test. Formally, we let p_{tr}, p_{te} come from a family of distributions whose members have different nuisance-label relationships but share the same relationship between the label and the semantics $\mathbf{x}^*$:

Definition 3. *(Nuisance-varying family with semantic features* $\mathbf{x}^*$ *[18, 48])*

$$\mathcal{F} = \{p_D \ : \ p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}^*, \mathbf{x}) = p(\mathbf{y}, \mathbf{x}^*) \ p_D(\mathbf{z} \mid \mathbf{y}) \ p(\mathbf{x} \mid \mathbf{z}, \mathbf{x}^*)\} \,. \tag{3.1}$$

Many common tasks in ood generalization, including some from section 3.4, fit this definition. For example, in classifying natural images, the background type is the nuisance $\mathbf{z}$ and its relationship to the label can change across places, each corresponding to a different member of $\mathcal{F}$. The animal shape however is made of semantic features $\mathbf{x}^*$ that are related to the label in the same way across places. Like in this example, we assume that the semantic features $\mathbf{x}^*$ equal a function of the covariates $\mathbf{x}^* = e(\mathbf{x})$ almost surely under every $p_D \in \mathcal{F}$, but neither $\mathbf{x}^*$ nor $e(\cdot)$ are known. Finally, the semantics and nuisances together account for all the information that $\mathbf{x}$ has about $\mathbf{y}$, meaning $\mathbf{x} \perp\!\!\!\perp_{p_D} \mathbf{y} \mid \mathbf{x}^*, \mathbf{z}$.

Building models that are robust to a shifting nuisance-label relationship relies on additional knowledge, such as nuisance annotations, in the training data [16, 17, 18, 48, 54]. Given knowledge of $\mathbf{z}$, work like [18, 48] estimate a distribution, denoted $p_{\perp\!\!\!\perp}$, under which the label and nuisance are independent ($\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$): $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x}) = \int_{z,x^*} p(\mathbf{y}, \mathbf{x}^* = x^*) p_{tr}(\mathbf{z} = z) p(\mathbf{x} \mid \mathbf{z} = z, \mathbf{x}^* = x^*) dz dx^*$. Following [48], we call $p_{\perp\!\!\!\perp}$ the *nuisance-randomized distribution.* The model $p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x})$ achieves the lowest risk on any member of the family $\mathcal{F}$ amongst the set of risk-invariant models; see Proposition 1 [18]). However, even when $p_{tr}, p_{te} \in \mathcal{F}$ and optimal risk-invariant predictors can be built with nuisances, *it is impossible to always beat random chance when given data* $\{\mathbf{y}, \mathbf{x}\} \sim p_{tr}$*:*

Theorem 2. *For any learning algorithm, there exists a nuisance-varying family $\mathcal{F}$ where predicting with $p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x})$ achieves* 90% *accuracy on all members such that given training data* $\mathbf{y}, \mathbf{x}$ *from one member $p_{tr} \in \mathcal{F}$, the algorithm cannot achieve better accuracy than* 50% *(random chance) on some $p_{te} \in \mathcal{F}$.*

The proof is in appendix B.1 and proceeds in two steps. With $\text{ACC}_h(p)$ as the expected accuracy of a model h on distribution p, the first step of the proof defines two nuisance-varying families $\mathcal{F}_1, \mathcal{F}_2$ such that no single model can perform well on both families simultaneously; any $h(\mathbf{x})$ for which $\text{ACC}_{p_1}(h) > 50\%$ for all $p_1 \in \mathcal{F}$ will have that $\text{ACC}_{p_2}(h) < 50\%$ for some $p_2 \in \mathcal{F}_2$ and vice-versa. The second step shows that the two families $\mathcal{F}_1, \mathcal{F}_2$ have a member that has the same distribution over $\mathbf{y}, \mathbf{x}$; letting the training data come from this distribution means that any learning algorithm that returns a performant model — one that beats 50% accuracy – on one family must fail to return a performant model on the other. Next, we discuss different methods that use additional knowledge beyond $\mathbf{y}, \mathbf{x}$ to build robust predictors.

3.2.1 Biased-model-based spurious-correlation avoiding methods.

We focus on methods that correct models using knowledge of nuisances or where they might appear in the covariates [47, 48, 53]. We first establish that the common central part in these methods is a model that predicts the label using nuisances, which we call the *biased model*; due to this commonality, we call these biased-model-based spurious-correlation avoiding methods (b-scams). At a high level, a b-scam has two components. The first is a biased model that is built to predict the label by exploiting the nuisance-label relationship via extra knowledge or assumptions. The biased model is then used to guide a second model to predict the label without relying on nuisances.

We briefly summarize the different b-scams here, differentiated by the additional knowledge they use to build biased models. The differences between the methods are summarized in table 3.1. We give details for NuRD here and defer algorithmic details about the rest to appendix B.2.

Biased models from knowledge of the nuisances. The first category of b-scams from Mahabadi et al. [47], Puli et al. [48] *assumes additional knowledge in the form of nuisance annotations* $\mathbf{z}$. For example, in nli — where the goal is determining if a premise sentence entails a hypothesis — [47] compute the fraction of words shared between the hypothesis and the premise for each sample in the training data and use this as one of the nuisance features in building the biased model. The biased model in NuRD, poe, dfl is learned by predicting the label from the nuisance annotations in the training data to estimate $p_{tr}(\mathbf{y} \mid \mathbf{z})$. Using nuisance annotations, Makar et al. [18], Puli et al. [48] use the model $p_{tr}(\mathbf{y} \mid \mathbf{z})$ as the biased model to define importance weights and minimize risk w.r.t a distribution $p_{\perp\!\!\!\perp}$ obtained as

$$p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p_{tr}(\mathbf{y})p_{tr}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = \frac{p(\mathbf{y})}{p_{tr}(\mathbf{y} \mid \mathbf{z})}p_{tr}(\mathbf{z})p_{tr}(\mathbf{y} \mid \mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = \frac{p(\mathbf{y})}{p_{tr}(\mathbf{y} \mid \mathbf{z})}p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x}).$$

Table 3.1: Summary of NuRD, JTT, POE, and DFL. Each method approximates the biased model: $p_{tr}(\mathbf{y} \mid \mathbf{z})$. This table describes the different biased models, their names, how they are built.

Method	Name	What the biased model is	Assumptions/Knowledge
JTT	Identification model	$p_{tr}(\mathbf{y} \mid \mathbf{x})$ learned via ERM	ERM learns biased models.
POE/DFL	Biased model	$p_{tr}(\mathbf{y} \mid \mathbf{z})$ learned via ERM	$\mathbf{z}$ from domain-knowledge.
NuRD	Weight model	$p_{tr}(\mathbf{y} \mid \mathbf{z})$ learned via ERM	$\mathbf{z}$ from domain-knowledge.

The second step in NuRD [48] trains a model to predict $\mathbf{y}$ from a representation $r(\mathbf{x})$ on data from $p_{\perp\!\!\!\perp}$ such that $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r(\mathbf{x})$; this step is called distillation. Due to $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$, learning in $p_{\perp\!\!\!\perp}$ avoids features that depend only on the nuisance and due to $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r(\mathbf{x})$, distillation avoids features that are mixed functions of the label and the nuisance (e.g. $\mathbf{x}_1 = \mathbf{y} + \mathbf{z}$). With these insights, NuRD builds models of the form $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ that are most informative of the label. Mechanically, NuRD's distillation solves this:

$$\max_{\theta,\gamma} \mathbf{E}_{p_{\perp\!\!\!\perp}} \log p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x})) - \lambda \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{z} \mid r_\gamma(\mathbf{x})).$$

Puli et al. [48] show that such models are best in a class of predictors with lower bounds on performance. The mutual information above is zero when $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{x}$; this condition holds for semantic corruptions as we discuss in appendix B.2. Thus, we run the distillation step as importance-weighted ERM on the training data.

Mahabadi et al. [47] consider two methods to train a biased model and a base predictive model jointly to make the base model predict without relying on the biases. They propose 1) POE, where the loss is the sum of the `log` loss of the two models and 2) DFL, where the biased model is used to weight the cross-entropy loss for the base model. For both methods, Mahabadi et al. [47] build a biased model as $p_{tr}(\mathbf{y} \mid \mathbf{z})$. Intuitively, the base model focuses on classifying samples that the biased model misclassifies. The methods fine-tune a BERT model [55] and do not propagate the gradients of the biased model to update the common parameters (token embeddings).

Biased models from assumptions on erm-trained models. The second category of b-scams like LFF [56], UMIX [57], and jtt [53] require *additional knowledge that vanilla erm builds a biased model that exploits the nuisance-label relationship.* Given such a model, these works use it to reduce a second model's dependence on the nuisance. We focus on jtt [53] which aims to build models robust to group shift, where the relative mass of a fixed set of disjoint groups of the data changes between training and test times. The groups here are subsets of the data defined by a pair of values of discrete label and nuisance values. While jtt works without relying on training group annotations, i.e. without nuisances, it assumes erm's missclassifications are because of a reliance on the nuisance. jtt first builds an "identification" model via erm to isolate samples that are misclassified. Then, jtt trains a model via erm on data with the loss for the misclassified samples upweighted (by constant λ). The epochs to train the identification model and the upweighting constant are hyperparameters that require tuning using group annotations [53].

The commonality of a biased model. The central part in NuRD, dfl, poe, and jtt is a model that predicts the label using nuisances, like $p_{tr}(\mathbf{y} \mid \mathbf{z})$, which we call the biased model as in He et al. [58]. The predictive models in each b-scam are guided to not depend on nuisances used by the biased model. While b-scams reduce dependence on nuisances, they build biased models using additional nuisance annotations or require assumptions that erm-trained models predict using the nuisance. In the next section, we describe an alternative: corrupt semantic information with data augmentations to construct biased models.

3.3 OOD generalization via Semantic Corruptions

The previous section summarized how biased models can be built in b-scams using either direct knowledge of nuisances or knowledge that erm-trained models rely on the nuisances. We now introduce semantic corruptions and show how they enable building biased models. Semantic corruptions are transformations of the covariates that do not retain any knowledge of the semantics, except what may be in the nuisance $\mathbf{z}$:

Definition 4 (Semantic Corruption). *A semantic corruption is a transformation of the covariates $T(\mathbf{x}, \boldsymbol{\delta})$, where $\boldsymbol{\delta}$ is a random variable such that $\boldsymbol{\delta} \perp\!\!\!\perp (\mathbf{y}, \mathbf{z}, \mathbf{x}, \mathbf{x}^*)$, if*

$$\forall\, p_D \in \mathcal{F} \quad T(\mathbf{x}, \boldsymbol{\delta}) \perp\!\!\!\perp_{p_D} \mathbf{x}^* \mid \mathbf{z}.$$

Here, we characterize conditions under which biased models built from semantic corruptions could be used to estimate robust models. As discussed in section 3.2, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ is the optimal risk-invariant predictor, and is the target of ERM when predicting the label $\mathbf{y}$ from $\mathbf{x}$ under the nuisance-randomized distribution $p_{\perp\!\!\!\perp}$. NuRD estimates this distribution as part of the algorithm, and methods like JTT aim to approximate $p_{\perp\!\!\!\perp}$, for example, upweighting samples mis-classified by a model that relies on $\mathbf{z}$ to predict $\mathbf{y}$. We compare $p_{\perp\!\!\!\perp}$ which is obtained by breaking the nuisance-label relationship against the distribution obtained by breaking the relationship between the label and the data augmentation :

$$p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x}) = \int_z \frac{p_{tr}(\mathbf{y})}{p_{tr}(\mathbf{y} \mid \mathbf{z} = z)} p_{tr}(\mathbf{y}, \mathbf{z} = z, \mathbf{x}), \qquad p_T(\mathbf{y}, \mathbf{x}) = \int_\delta p(\boldsymbol{\delta} = \delta) \frac{p_{tr}(\mathbf{y})}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \delta))} p_{tr}(\mathbf{y}, \mathbf{x}) d\delta.$$

We show here that the L_1 distance between $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x})$ and $p_T(\mathbf{y}, \mathbf{x})$ is controlled by an L_2-distance between the biased models built from the nuisance and the data augmentations respectively:

Proposition 1. *Let $T : \mathbf{X} \times \mathbf{R}^d \to \mathbf{X}$ be a function. Assume the r.v. $p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))^{-1}$ has a bounded second moment under the distribution $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) p(\boldsymbol{\delta})$, and that $p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))$ and $p_{tr}(\mathbf{y} \mid \mathbf{z})$ satisfy*

$$\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x}) p(\boldsymbol{\delta})} p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))^{-2} \leq m^2, \qquad \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x}) p(\boldsymbol{\delta})} \left| p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z}) \right|^2 = \epsilon^2.$$

Then, the L_1 distance between $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x})$ and $p_T(\mathbf{y}, \mathbf{x})$ is bounded: $\|p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x}) - p_T(\mathbf{y}, \mathbf{x})\|_1 \leq m\epsilon$. For a *semantic corruption that also satisfies* $\mathbf{y} \perp\!\!\!\perp_{p_{tr}} \mathbf{z} \mid T(\mathbf{x}, \boldsymbol{\delta})$ *the inequalities hold with* $\epsilon = 0$.

If $\epsilon = 0$, $p_T(\mathbf{y}, \mathbf{x}) = p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x})$ which means that almost surely the conditionals match $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}) = p_T(\mathbf{y} \mid \mathbf{x})$. Then, as $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ is the optimal risk-invariant predictor, so is $p_T(\mathbf{y} \mid \mathbf{x})$. More generally, standard domain adaptation risk bounds that are controlled by the total variation distance between source and target [59, Theorem 1] bound the risk of a model under $p_{\perp\!\!\!\perp}$ using the L_1 bound $m\epsilon$ — which upper bounds the total variation — and the risk under p_T.

Without nuisance annotations, one cannot test whether estimate the L_2-distance between the two biased models $p_{tr}(\mathbf{y} \mid \mathbf{z})$ and $p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))$ in proposition 1. This distance can be large when a transformation $T(\mathbf{x}, \boldsymbol{\delta})$ retains semantic information. To avoid, we turn to a complementary source of knowledge: semantic features. Using this knowledge, we design families of data augmentations that corrupt the semantic information in $\mathbf{x}$ to construct semantic corruptions. Focusing on two popular OOD tasks, object recognition and NLI, we use **only semantic knowledge** to build corruptions that retain some aspects of the covariates. Biased models built on such corruptions will depend on any retained nuisances; more retained nuisances mean better biased models.

3.3.1 Semantic corruptions via permutations

We first build corruptions when semantics appear as global structure. We give an intuitive example for such global semantics. Consider the waterbirds dataset from Sagawa et al. [16] with waterbirds and landbirds appearing predominantly on backgrounds with water and land respectively. Semantic features like the wing shape and the presence of webbed feet are corrupted by randomly permuting small patches. See fig. 3.1(a). Formally, given subsets of the covariates $\mathbf{x}_1, \cdots \mathbf{x}_k$ extracted in an order, global semantics $e(\mathbf{x}_1, \cdots, \mathbf{x}_k)$ change with the order of extraction. Formally, with a random permutation $\pi \sim q(\pi)$ and recalling that semantics are $\mathbf{x}^* = e(\mathbf{x})$, the information about semantics is lost after permutation: $\forall p_D, \mathbf{I}_{p_D, q(\pi)}(\mathbf{x}^*; e(\mathbf{x}_{\pi(1)}, \cdots \mathbf{x}_{\pi(k)}))) = 0$.

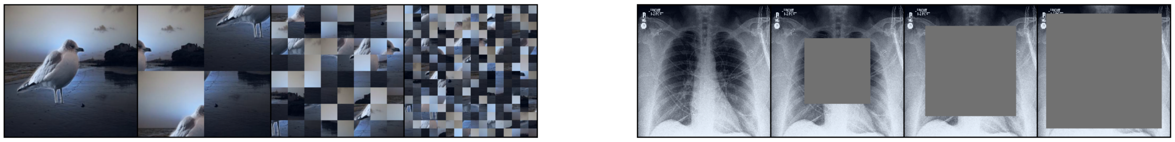
We give an example of a semantic corruption with global semantics. Consider distributions $\{p_D\}_{D\in\mathbf{R}}$ with different nuisance-label relationships. With $\mathcal{U}$ as the uniform distribution over $\{1, 2, 3\}$, and $\mathcal{N}$ as the normal distribution, $p_D(\mathbf{y}, \mathbf{z}, \mathbf{x})$ corresponds to $\mathbf{y} \sim \mathcal{U}$, $\mathbf{z} \sim \mathcal{N}(D\mathbf{y}, 1)$, and $\mathbf{y}$ selecting a configuration of $\mathbf{x}$

$$\mathbf{y} = 1 \implies \mathbf{x} = [-\mathbf{z}, \mathbf{z}, \mathbf{z}], \qquad \mathbf{y} = 2 \implies \mathbf{x} = [\mathbf{z}, -\mathbf{z}, \mathbf{z}], \qquad \mathbf{y} = 3 \implies \mathbf{x} = [\mathbf{z}, \mathbf{z}, -\mathbf{z}]$$

The index of the negated coordinate is the semantic feature $\mathbf{x}^*$ that equals $\mathbf{y}$ and computing it requires comparing coordinates: $\mathbf{y} = 1$ if $\mathbf{x}_2\mathbf{x}_3 > 0$, $\mathbf{y} = 2$ if $\mathbf{x}_1\mathbf{x}_3 > 0$, and $\mathbf{y} = 3$ otherwise. In words, the semantic feature is global. However, $\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$ is determined regardless of where the negative sign is, i.e. it is not global. A random permutation $T(\mathbf{x}, \boldsymbol{\delta})$ of the coordinates in $\mathbf{x}$ is thus a semantic corruption: as $T(\mathbf{x}, \boldsymbol{\delta})$ permutes the location of the negation, $T(\mathbf{x}, \boldsymbol{\delta}) \mid \mathbf{y}, \mathbf{z}$ is distributed identically to $T(\mathbf{x}, \boldsymbol{\delta}) \mid \mathbf{z}$. In turn, $T(\mathbf{x}, \boldsymbol{\delta}) \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$. Further, the product of the three coordinates of $T(\mathbf{x}, \boldsymbol{\delta})$ determines $\mathbf{z}$: $(\Pi_{i\in\{1,2,3\}} T(\mathbf{x}, \boldsymbol{\delta})_i)^{1/3} = -\mathbf{z}$. Thus, $T(\mathbf{x}, \boldsymbol{\delta})$ determines $\mathbf{z}$ and $\mathbf{y} \perp\!\!\!\perp \mathbf{z} \mid T(\mathbf{x}, \boldsymbol{\delta})$. These two independencies imply that $\epsilon = 0$ in proposition 1. Then, biased models from $T(\mathbf{x})$ are as good as ones from $\mathbf{z}$. Next, we give corruptions for global semantics in vision and language tasks, that retain non-global features.

Patch randomization. Object recognition tasks where the object is a shape that can satisfy the global property. For illustration, consider differentiating cows and penguins in natural images; here, shape is a global semantic feature that structures multiple patches. Permuting patches via *patch randomization (pr)*, like in fig. 3.1(a), corrupts global semantics.

N-gram randomization. Tasks like natural language inference (nli) — where the goal is determining if a premise sentence entails a hypothesis — satisfy the global-semantics property. Consider this example: the sentence "Bob speaks but Jon does not" contradicts "Jon speaks but Bob does not" but entails "Bob speaks". The meaning is inferred from a global structure on the words



(a) PR to corrupt global semantics in Waterbirds. The original is on the left, followed by PRs with sizes 112, 28, 14. At sizes > 28, shape is corrupted.

(b) ROI-MASK to corrupt semantics in chest X-rays. The original is on the left, followed by ROI-MASK of size 112, 154, 196. For 196, the heart is blocked out.

Figure 3.1: Semantic corruptions of Waterbirds via PR and chest X-rays via ROI-MASK.

and the order they appear in. Here, randomizing the order of the words corrupts the semantics: For example, one randomized order of the sentence "Jon speaks but Bob does not" is "Bob speaks but Jon does not"; the former entails "Jon speaks" but the latter contradicts it. We randomize the order by permuting different n-grams in each sentence; we call this *n-gram randomization (NR).*

3.3.2 Semantic corruptions via masking

The second corruption we build focuses on cases where certain subsets of the covariates are necessary part of semantics. Masking, by removing that subset or setting it to a constant, corrupts semantics. Formally, we corrupt the semantics by replacing subsets $\mathbf{x}_S$ with a value that is out of support: for example, in images where pixels lie in $(0, 1)$, we corrupt $\mathbf{x} = [\mathbf{x}_S, \mathbf{x}_{-S}]$ as $\mathbf{x}_{\text{corrupted}} = [0 * \mathbf{x}_S, \mathbf{x}_{-S}]$. As an illustrative example, consider a family $\mathcal{F} = \{p_D\}_{D\in R}$ with varying nuisance-label relationships. With $\mathbf{a}, \mathbf{b}$ being uniform binary random variables, $\mathbf{e}(\rho)$ as the exponential distribution with parameter ρ, and $s_+(u) = \log(1+\exp(u))$ as softplus, $p_D(\mathbf{y}, \mathbf{z}, \mathbf{x})$ describes:

$$\mathbf{y} = \mathbf{a} \oplus \mathbf{b}, \qquad \mathbf{z} \sim \mathbf{e}(s_+(D * (2\mathbf{y} - 1))), \qquad \mathbf{x} = [(2\mathbf{a} - 1)\mathbf{z}, (2\mathbf{b} - 1)\mathbf{z}]. \tag{3.2}$$

For such a family, we show that masking out coordinate $\mathbf{x}_1$ is a semantic corruption: $T(\mathbf{x}) = [0, \mathbf{x}_2]$ satisfies $T(\mathbf{x}) \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$ and $T(\mathbf{x}) \not\!\perp\!\!\!\perp \mathbf{z}$. First, due to $\mathbf{y}$ being computed as an XOR function of $\mathbf{a}, \mathbf{b}$, it holds that $\mathbf{b} \perp\!\!\!\perp \mathbf{y}$. Then, due to $\mathbf{z}$ only relying on $\mathbf{y}$ and exogenous noise, $\mathbf{b} \perp\!\!\!\perp (\mathbf{y}, \mathbf{z})$ which implies

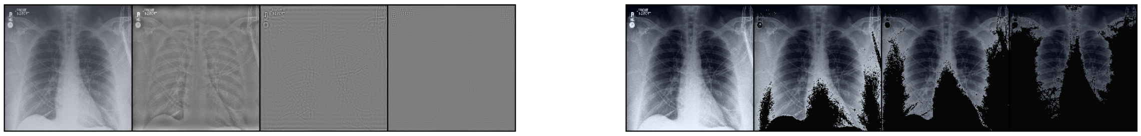
$\mathbf{b} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$. Given $\mathbf{z}$, $\mathbf{b}$ determines $\mathbf{x}_2$, so $\mathbf{b} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \implies \mathbf{x}_2 \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \implies T(\mathbf{x}) \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$. Further, $\|T(\mathbf{x})_2\| = \mathbf{z}$ which means $\mathbf{y} \perp\!\!\!\perp \mathbf{z} \mid T(\mathbf{x})$. These two independencies imply that $\epsilon = 0$ in proposition 1. Then, using $T(\mathbf{x})$ to build biased models is equivalent to building them with $\mathbf{z}$.

ROI-MASKING FOR OBJECT RECOGNITION. Semantics in images can often be localized to a region-of-interest (ROI). For example, in detecting cardiomegaly, the ROI is the chest where the heart resides. Masking out the ROI removes centrally located semantic information from the chest X-ray (fig. 3.1(b)). We call this *ROI-MASK*.

PREMISE-MASKING FOR NLI. Semantic features in NLI rely on the meanings of the premise and the hypothesis sentences: for example, the premise states the occurrence of an event ("Alice sat while Bob stood.") which can entail ("Alice sat.") or contradict ("Bob sat.") the hypothesis. The information about the setup in the premise is therefore crucial to detect entailment or contradiction. If the context given by the premise is blocked out, the hypothesis sentence can predict the label only due to nuisances. Thus, masking the premise is a semantic corruption for NLI that retains hypothesis features; we call this *PREM-MASK*.

3.3.3 SEMANTIC CORRUPTIONS VIA FREQUENCY AND INTENSITY FILTERS

PR relies on differences in relative size and ROI-MASK relies on differences in spatial position. We consider two aspects of the image that are not spatial, frequency and pixel-intensity, and give corruptions for features that depend on these aspects. Semantics can appear as signals in a particular region of the frequency spectrum, or appear at a particular luminosity in the image. For example, consider detecting cardiomegaly from chest X-rays, where the heart appears as an object formed of bright pixels with little variation in intensity across the pixels; the latter suggests that the heart features are low-frequency signals.



(a) Corruption via freq-filt. Original image to the left followed zeroing out 14, 56, 112 of the lowest frequencies. The heart features are corrupted at 56.

(b) Corruption via int-filt. Original image to the left followed by zeroing out pixels with intensities above the 80%, 60%,40%. Heart features look corrupted at 40%.

Figure 3.2: Semantic corruptions of chest X-rays via freq-filt and int-filt respectively.

This observation motivates corruptions along the axes of frequency and pixel-intensity: *freq-filt* and *int-filt*. Freq-filt zeroes-out frequencies in the discrete fourier domain, while int-filt zero-out pixels based on their intensities. See fig. 3.2 for how freq-filt and int-filt corrupt the heart region. freq-filt and int-filt require characterizing semantic features in frequency and intensity spaces; this is in contrast to roi-mask that is based on characterizing the position in pixel space that the semantics occur in.

3.3.4 Using semantic corruptions in practice

For each method in table 3.1, we use a semantic corruption $T(\mathbf{x})$ in building a model $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$. For reweighting-NuRD, we replace $p_{tr}(\mathbf{y} \mid \mathbf{z})$ with $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$, for dfl and poe, we replace the model $p_{tr}(\mathbf{y} \mid \mathbf{z})$ with $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$, and for jtt, we use $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$ as the identification model.

Choosing the corruption parameter. To corrupt with pr, nr, and roi-mask, freq-filt, one must select a size parameter and to corrupt with int-filt, one must specify an intensity threshold. For NuRD, jtt, poe and dfl, we select corruption parameters with the same validation schemes used to select other hyperparameters in each respective paper. In practice, including the b-scams run without semantic corruptions in the b-scam's validation scheme ensures a lower bound on performance. For example, for jtt, this inclusion yields a lower bound that corresponds to vanilla jtt's performance. We also report results for all corruption parameters in appendix B.3.3, showing that all semantic corruptions except INT-FILT are not sensitive to the parameters, and lead to models that outperform ERM.

3.4 Experiments

We study semantic corruptions in powering NuRD [48], JTT [53], and POE and DFL [47]. To be faithful to the original evaluations of each method, we run them on tasks from their respective papers: NuRD on waterbirds, JTT on waterbirds and NLI where the nuisance is the presence of a negation word, and POE and DFL on NLI evaluated on a challenging test dataset, HANS [60]. We run NuRD on chest X-rays but focus on detecting cardiomegaly rather than the original pneumonia [48] because pneumonia detection even with known-nuisances is not performant. See appendix B.3 for details and appendix B.3.3 for additional experiments investigating semantic corruptions.

Methods, metrics and model selection. For images, we corrupt semantics with PR, a central ROI-MASK, FREQ-FILT, and INT-FILT. To show the value of semantic corruptions relative to existing data augmentations, we also consider two baseline transformations of images. The first is random cropping (RAND-CROP) like in self-supervised learning [61, 62] where patches of random sizes are sampled, covering ≥ 0.08 fraction of the image. The second is adding gaussian noise (GAUSS-NOISE). For text, we corrupt semantics with NR and PREM-MASK. We report the average test accuracy for every method. To be able to compare to what JTT is trained for in Liu et al. [53], we report worst-group test accuracy for JTT. For each method, we compare the performance of the original method to that of the methods run with semantic corruption (including the baselines). For the corruption-powered versions, group annotations and nuisances are *unavailable* in the training data. Known-nuisance versions of POE, DFL, and NuRD use direct knowledge of one or more nuisances during training. In choosing parameters and early stopping, like Liu et al. [53]

do with vanilla JTT, corruption-powered JTT uses validation group annotations. For the other methods, we follow validation schemes from the respective papers: for NuRD we follow Puli et al. [48] and use a validation set weighted to have independent nuisance and label, and for POE/DFL, we follow Mahabadi et al. [47] and use a set of 1000 samples from the HANS training dataset.

3.4.1 Object recognition tasks

To be faithful to the original evaluations of each method, we evaluate JTT on waterbirds, and NuRD on both waterbirds and detecting cardiomegaly; both tasks have images of size 224×224×3. Both Puli et al. [48] and Liu et al. [53] use the raw waterbirds data from Sagawa et al. [16], where the task is detecting the type of bird (water or land) from images where the background is a nuisance. Unlike Liu et al. [53], Puli et al. [48] process the waterbirds to get a different setup from Sagawa et al. [16]. To stay true to the original evaluations of the methods, we recreate the setups as described in their respective papers. For both tasks, we use PR (of patch sizes $7, 14, 28, 56$), ROI-MASK (of mask sizes $112, 140, 168, 196$), FREQ-FILT (of high-pass filter sizes $196, 168, 140, 112$), and INT-FILT (of thresholds $0.1, 0.2, 0.3, 0.4$) as semantic corruptions. For GAUSS-NOISE, we use variances $0.01, 0.25, 1, 4$.

NuRD on waterbirds. For NuRD, we recreate the waterbirds experiment from Puli et al. [48] where the full waterbirds data from Sagawa et al. [16] was subsampled into training, validation, and test datasets each with label balance. However, unlike Sagawa et al. [16], the validation data comes from the same distribution as the training data. The training and validation datasets have 90% waterbirds on backgrounds with water and 90% landbirds on backgrounds with land. The test data has a flipped relationship. Known-nuisance NuRD uses an additional label denoting the background-type as the nuisance.

Table 3.2 gives results. Selecting hyperparameters using NuRD's validation approach gives sizes

14 for PR (86.9%), 196 for ROI-MASK (86.9%), 168 for FREQ-FILT (83.5%), and threshold 0.2 for INT-FILT (86.9%). Consider the gap between ERM and known-nuisance NuRD. NuRD with PR, ROI-MASK, FREQ-FILT, and INT-FILT close 99%, 99%, 82%, 99% of the gap respectively. NuRD with these semantic corruptions outperforms ERM (68.0%) and NuRD with RAND-CROP (73.7%) and GAUSS-NOISE (82.0%).

Table 3.2: Mean and standard error of test accuracy across 10 seeds of NuRD with semantic corruptions on waterbirds. *Known*-z NuRD uses a label for the type of background as the nuisance. Consider the gap between ERM and known-nuisance NuRD. NuRD with semantic corruptions PR, ROI-MASK, FREQ-FILT, and INT-FILT close 99%, 99%, 82%, 99% of the gap respectively. NuRD with semantic corruptions outperforms ERM and NuRD with RAND-CROP, GAUSS-NOISE.

Method	test acc.
Known-z NuRD	87.2 ± 1.0%
PR	86.9 ± 1.2%
ROI-MASK	86.9 ± 1.7%
FREQ-FILT	83.5 ± 1.1%
INT-FILT	86.9 ± 1.1%
RAND-CROP	73.7 ± 2.0%
GAUSS-NOISE	82.0 ± 2.6%
ERM	68.0 ± 1.9%

Additionally, in table B.4 in appendix B.3, we give the results for all corruption parameters, showing that NuRD with semantic corruptions is *insensitive to hyperparameters of the corruption* and outperforms ERM. In appendix B.3.1, we discuss how the baseline GAUSS-NOISE could close 80% of the gap between ERM and known-**z** NuRD.

Table 3.3: Test worst-group (WG) accuracies of JTT on waterbirds. JTT with semantic corruptions outperforms ERM, vanilla JTT, and JTT with baseline corruptions (RAND-CROP, GAUSS-NOISE).

Method	test WG acc.
Vanilla JTT	86.5%
PR	89.0%
ROI-MASK	88.2%
FREQ-FILT	87.2%
INT-FILT	87.0%
RAND-CROP	75.0%
GAUSS-NOISE	71.0%
ERM	72.0%

JTT ON WATERBIRDS. For JTT, we repeat the waterbirds experiment from Liu et al. [53] which uses the original data from Sagawa et al. [16]. The training data has 95% waterbirds on backgrounds with water and 95% landbirds on backgrounds with land. Both the validation and test datasets have bird label independent of the background. The groups here are subsets of the data that correspond to a pair of values of bird-type and background-type. Like vanilla JTT, we use group annotations in the validation data to compute worst-group error and early stop training when using PR and ROI-MASK. The results for vanilla JTT are from our run using the optimal hyperparameters from Liu et al. [53].

Table 3.3 shows the results. Selecting the corruption hyperparameters on the validation worst-group accuracy gives size 14 for PR (89%), size 196 for ROI-MASK (88.2%), size 112 for FREQ-FILT (87.2%), and threshold 0.4 for INT-FILT (87.0%). JTT with these semantic corruptions outperforms ERM (72.0%), vanilla JTT (86.5%), and JTT with the baseline corruptions RAND-CROP (75%) and GAUSS-NOISE (71%). Additionally, table B.7 shows that JTT with PR and ROI-MASK outperforms JTT with the baseline corruptions and ERM at every patch/border-size.

NuRD on detecting cardiomegaly In chest X-ray classification, differences between hospitals, like the scanners used to produce the X-rays, are known to correlate thoracic conditions with non-physiological aspects in the image; for example, only some scanners render the air in the lungs in white [9]. We consider the shape-based object recognition task of cardiomegaly (an irregularly sized heart) detection and, following Puli et al. [48], construct a dataset from two chest X-ray datasets: chexpert [38] and MIMIC [39]. The training and validation datasets have 90% cardiomegaly images from MIMIC and 90% healthy images from chexpert, while the test data has a flipped relationship. Known-nuisance NuRD uses hospital identity as the nuisance.

Table 3.4: Mean and standard error of test accuracy over 10 seeds of NuRD on chest X-rays. *Known*-z NuRD uses the hospital as the nuisance. Consider the gap between ERM and known-z NuRD. NuRD with PR, ROI-MASK, FREQ-FILT, and INT-FILT close 72%, 82%, 65%, 35% of the gap respectively. Except with INT-FILT, NuRD with semantic corruptions outperforms ERM and NuRD with baseline corruptions.

Method	test acc.
Known-z NuRD	81.7 ± 0.3%
PR	77.0 ± 1.2%
ROI-MASK	78.7 ± 0.3%
FREQ-FILT	76.0 ± 0.6%
INT-FILT	71.0 ± 1.0%
RAND-CROP	59.9 ± 2.1%
GAUSS-NOISE	69.0 ± 1.9%
ERM	65.3 ± 1.1%

See table 3.4 for results. Selecting the corruption parameters using NuRD's validation approach gives size 14 for PR (77%), size 196 for ROI-MASK (78.7%), size 168 for FREQ-FILT (76.0%), and threshold 0.1 for the INT-FILT (71.0%). Consider the gap between ERM and known-nuisance NuRD. NuRD with PR, ROI-MASK, FREQ-FILT, and INT-FILT close 72%, 82%, 65%, 35% of the gap respectively. NuRD with all semantic corruptions, outperforms ERM (65.3%) and NuRD with the baselines GAUSS-NOISE (69%) and RAND-CROP (59.9%). Additionally, we report results for all corruptions in table B.4 in appendix B.3 showing that NuRD with PR and ROI-MASK *are insensitive to hyperparameters* and outperform ERM.

3.4.2 Nli

For methods POE, DFL, and JTT, we use the MNLI dataset [63] to fine-tune a BERT model. The evaluations of these methods in their respective papers have different nuisances and, consequently, different test sets. In accordance, we describe the setup and the results separately. We use NR (sizes 1, 2, 3, 4) to produce nuisances for both setups.

Table 3.5: Mean and standard deviation of accuracies (over 4 seeds) on the HANS dataset. The results for POE and DFL that use known nuisances are given under *known*. POE with NR (NR) performs better than known-nuisance POE. DFL with (NR) closes 84% of the gap between ERM and known-nuisance DFL. PoE and DFL with PREM-MASK (PM) close 33% and 28% of the gap between ERM and the respective method with known **z**.

Method	HANS test acc.
POE, *known*-**z**	66.3 ± 0.6%
POE, NR	66.7 ± 1.5%
POE, PM	64.5 ± 1.9%
DFL, *known*-**z**	69.3 ± 0.2%
DFL, NR	68.4 ± 1.5%
DFL, PM	65.2 ± 0.7%
ERM	63.6 ± 1.1%

PoE AND DFL For POE and DFL, we report test accuracies on the HANS dataset [60] as in Mahabadi et al. [47]. HANS was created to test the reliance of models on three known nuisances: 1) lexical overlap, 2) subsequence match, and 3) constituent matching subtrees in the parse trees. Known-nuisance POE and DFL use exact knowledge of these nuisances.

Table 3.5 gives the mean test accuracies over 4 seeds. For both DFL and POE, selecting the size hyperparameter based on the average accuracy on a small subset of the HANS training data (1000 samples) gives $n = 3$. With this size, POE achieves 66.7%, improving over POE with known nuisances (66.3%). DFL with NR of size 3, achieves 68.4%, closing 84% of the gap between ERM and known-nuisance DFL (69.3%).

PoE and DFL with PREM-MASK (PM) close 33% and 28% of the gap between ERM and when the methods have knowledge of **z**. We expect the methods with NR do better than with PREM-MASK because the latter corrupts nuisances like lexical overlap between premise and hypothesis that HANS focuses on. Additionally, we give results for all n-gram sizes in table B.5 in appendix B.3, showing that POE and DFL beat ERM for all n-gram sizes. Further, in appendix B.3.3.1, we evaluate POE and DFL models on the ANLI [64] dataset and counterfactually-augmented data [65] in tables B.9 and B.10.

JTT For JTT, we repeat the NLI experiment from Liu et al. [53], where the presence of a negation word in the hypothesis sentence is the nuisance. The groups here are subsets of the data that correspond to a value of the label and whether or not there is a negation word in the hypothesis. Vanilla JTT uses group annotations in the validation data to tune the hyperparameters and early stop training. For each n-gram size, we run JTT with NR for two values of the number of epochs of training for the identification model: $2, 3$. Following the hyperparameter selection procedure from Liu et al. [53], for each n-gram size, we give the results for the run with the higher validation worst-group accuracy. *Vanilla* JTT is run with the optimization hyperparameters from [53].

Table 3.6: Worst-group and average test accuracies of JTT on NLI. JTT with PREMMASK (PM) and NR (NR) outperforms vanilla JTT and ERM.

	Worst-group	Avg.
Vanilla JTT	71.3%	79.1%
JTT + PM	72.1%	79.9%
JTT + NR	74.3%	79.7%
ERM	67.9%	82.4%

Table 3.6 gives the results. Selecting the size hyperparameter for NR using validation worst-group accuracy, like Liu et al. [53] do for JTT, gives $n = 1$ with test worst-group accuracy of 74.3%, better than vanilla JTT's 71.3%. Additionally, table B.8 shows that JTT using NR at *every* size or PREMMASK performs better than both vanilla JTT (71.3%) and ERM (67.9%).

3.5 Related work

Biased-model-based spurious-correlation avoiding methods (B-SCAMS) like [17, 18, 48, 58, 66] assume the nuisance is available as additional knowledge during training. Semantic corruptions offer a complementary approach to hand-crafting nuisances or obtaining auxiliary labels, by capturing nuisances that remain after corruption (e.g. non-global nuisances remain after PR). B-scams like LFF [56], UMIX [57], and jtt [53] all rely on one crucial assumption: that erm-training builds a biased model that exploits the nuisance and use it to reduce a second model's dependence on the nuisance. Each method trains the second model with weighted cross-entropy with higher weights for samples misclassified by the biased model; the methods differ in how they build the biased model and how they compute the weighted loss. The biased models learn to predict the label from the covariates. Such a model can also rely on the semantic features and upweighting its misclassified samples produces data with a different label-semantic relationship from the one in the training data. Models trained on such data are suboptimal on test data which has the same semantic relationship as the training data. Using semantic corruptions in these b-scams reduces the biased model's reliance on the semantics and makes the second model rely more on the semantics; thus, b-scams that rely on assumptions on erm-trained models being biased achieve better performance when using semantic corruptions. The experiments in section 3.4 confirm this empirically: jtt with semantic corruptions improves over vanilla jtt.

Two instances of semantic corruptions, prem-mask and roi-mask, appear in earlier work [47, 48, 58] but were designed using knowledge of where nuisances appear in the covariates. [48] used the borders of X-ray images as features that are related only to the scanner type (the nuisance), and not human physiology, to avoid spurious correlations in the detection of cardiomegaly. For nli, Mahabadi et al. [47] use knowledge that the test set was constructed from samples misclassified by a model that relies on the hypothesis alone to build a biased model using the hypothesis sentence. These are special cases of roi-mask and prem-mask from section 3.3.2 repsectively. Our work widely generalizes the observations from the papers above by formally defining and further realizing the abstraction of semantic corruptions in several instances and across applications.

Bahng et al. [67] uses cnns with small receptive fields (RFs), to capture non-global nuisances. However, their method is typically limited to very small filters because at size 3x3, deep neural networks like VGG detect global semantics like shapes. In contrast, the size choice in PR has no bearing on the choice of the model; we used default vision models. Bras et al. [68] automatically identify and remove examples with nuisances using adversarial filtering, but risk removing genuinely easy examples. Qin et al. [69] work solely with vision transformers and point out that nuisances are the only reason labels can be predicted from transformations akin to patch-randomized images. They propose to encourage transformers to have predictions and representations of the original images be dissimilar from those of patch-randomized ones. In contrast, our work applies to general flexible models and shows that semantic corruptions can be used to break the label's relationship with nuisances in the original images.

[51, 54] use additional knowledge about nuisances or environments to corrupt nuisances in the covariates, [54] corrupt nuisances in the covariates via the Mixup [70] of samples from different domains that share a label. [51] directly randomize nuisances; for example, in detecting animals in their natural habitats, they place segmented animal foregrounds onto random habitat backgrounds. Unlike these methods, we design semantic corruptions using the complementary knowledge about semantics, which can be available even without knowledge about nuisances. Clark et al. [66], Li and Vasconcelos [71] construct nuisances in the training stage using prior knowledge: for example, [66] uses the first token of the hypothesis as a nuisance for a synthetic NLI task which was created to have the first token be spuriously correlated with the label. Another example is the VQA task where the question-type is used as the nuisance. The constructed nuisances are then used to build biased (or bias-only) models, or construct per-sample weights to de-bias the loss. In contrast, we use knowledge about semantics to corrupt them; for example, the order of the words is a semantic feature that is corrupted by randomizing the order. This construction does not use knowledge of the nuisance.

Sinha et al. [72] use techniques like PR to restrict supports in self-supervised learning and generative modeling. Carlucci et al. [73] use PR images to encourage a model to recover semantic structure. In contrast, we use PR to corrupt semantics and build biased models that rely on the nuisances, which help build predictive models that avoid reliance on nuisances. We focus on corrupting semantic features using simple procedures (like permuting, masking, filtering) while papers [8, 65, 74, 75, 76, 77, 78] focus on perturbing semantic features while keeping other features the same. These transformations produce examples of different labels, and are called counterfactuals. These examples are used to counterfactually augment the training data [65]. Constructing counterfactuals can be hard. Works like [65, 74, 75, 76] rely on humans to create counterfactuals because it is difficult to automate semantic perturbation without changing nuisances. For example, consider classifying dogs versus cats. Creating a dog that looks like a specific cat is much harder than removing the cat from the image by masking out those pixels.

Methods like [8, 78] construct counterfactuals automatically, but require additional knowledge of how nuisances appear in the text. For example, Wang and Culotta [78] matches sentences that have opposite labels while sharing most words. The non-shared words would then be considered semantic. Techniques like the matching one above from [8] are unrealistic beyond the task of sentiment classification. For example, consider the label of entailment or contradiction in NLI. Data samples with entailment as the label that contain negation words are rare. This makes it hard to find a good counterfactual for data samples labeled with contradiction. Further, matching is difficult in other modalities, like images, where covariates are continuous or high-dimensional and live in spaces where metrics are unclear.

3.6 Discussion

We study the use of semantic knowledge in models robust to spurious correlations. In theorem 2, we show that additional knowledge is necessary to achieve OOD generalization even when the training and test distributions are coupled in a nuisance-varying family. Then, proposition 1 shows that a biased model built from a transformation of the covariates $T(\mathbf{x}, \boldsymbol{\delta})$ — that is

$p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})$ — can power B-SCAMs to avoid nuisances if the biased model $p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))$ is close to $p_{tr}(\mathbf{y} \mid \mathbf{z})$ in L_2 distance. There are two scenarios where this distance is large: the transformation does not corrupt semantics and it corrupts nuisances. We use knowledge of the semantics to design semantic corruptions to avoid the first scenario. *Since we work without nuisances*, to avoid the second scenario — that is to choose $T(\mathbf{x}, \boldsymbol{\delta})$ that retain nuisances — we use standard validation schemes in B-SCAMs. Using semantic corruptions, practitioners can run different kinds of B-SCAMs (NuRD, JTT, DFL, POE). Corruption-powered methods like NuRD and DFL perform close to how they would with known nuisances. For methods like JTT, the corruption-powered versions perform better than their vanilla versions that rely on ERM on the raw covariates to yield nuisances.

Limitations. The quality of any semantic corruption, and thus the quality of the results, depends on the extent to which semantics are destroyed and nuisances are retained. Pr and NR are built to corrupt global semantics, and therefore are most suitable for when the nuisances are local. Roi-MASK corrupts semantics in the ROI and PREM-MASK corrupts the semantic context in the premise; these are most suitable for when nuisances lie outside the region-of-interest (ROI) or in the hypothesis respectively. Finally, FREQ-FILT and INT-FILT corrupt semantics in particular parts of the frequency and intensity spectrum, and are most suitable for when the nuisances and semantics lie in separate parts of the spectra. Knowledge about the kind of nuisances present in a dataset can lead to better choices of semantic corruptions. Alternatively, one could use standard validation schemes to select a corruption, like we do in section 3.4.

When applied blindly, the procedures we describe may retain semantics or corrupt nuisances. Pr and NR may corrupt global nuisances and retain local semantics, ROI-MASK and PREM-MASK may corrupt nuisances that occur in the same region as the semantics, and FREQ-FILT and INT-FILT may corrupt both semantics and nuisances if they appear at similar frequencies or intensity. For example, when PR is used blindly on covariates with non-global semantics, the biased model may rely on said semantics; this in turn guides the predictive model to ignore these semantics and, thus, lose predictive performance. Alternatively, when nuisances are global, PR may corrupt them. For example in detecting cows and penguins, other nuisance animals (like dogs) may co-occur with cows more often; PR would corrupt this nuisance animal. Using PR in a B-SCAM for such tasks could lead to non-robust predictive models that rely on corrupted nuisances.

Our experiments suggest that it might be possible to guard against performance degradation due to blind usage of semantic corruptions if the corruption parameter is made a hyperparameter and selected using standard validation schemes. In both classifying waterbirds and NLI, there exist non-global semantics, like small beaks and individual words, that are not corrupted by PR and NR respectively. However, in our Waterbirds and NLI experiments, we show models built using semantic corruptions, with validated size choices, close more than 80% of the gap in test performance between ERM and the methods that use known nuisances. Now, imagine the extreme case of running NuRD, POE, DFL with a semantic corruption that destroys all information in the covariates. Biased models would predict like random chance, and the resulting predictive models would be no less robust than ERM. On the other hand, methods like JTT perform at least as well as their vanilla versions as long as the validation strategy used in vanilla JTT covers the identity function as a corruption. Future work could consider combining semantic corruptions as a way to better retain of nuisances. Given the validation strategies for B-SCAMS, a practitioner can easily validate over both single and hybrid corruptions.

SUMMARY. Semantic corruptions power B-SCAMS to build models robust to spurious correlations using knowledge about the semantic features. Additional knowledge is always required to achieve such robustness, and existing work assumes access to nuisance annotations or that ERM-trained models rely on nuisances. By developing semantic corruptions, we give an approach to use a new kind of additional knowledge, thereby enlarging the set of tasks where one can build robust models. As discussed above, our experiments show that using semantic corruptions in

B-SCAMS leads to models more robust than ERM and JTT even when the corruptions may have corrupted some nuisances. These two properties demonstrate the value of semantic corruptions as a way to build robust models.

4 | Don't Blame Dataset Shift: Shortcut Learning due to Gradients and Cross Entropy

4.1 Introduction

Shortcut learning is a phenomenon where a model learns to base its predictions on an unstable correlation, or *shortcut*, that does not hold across data distributions collected at different times and/or places [5]. A model that learns shortcuts can perform worse than random guessing in settings where the label's relationship with the shortcut feature changes [79, 80]. Such drops in performance do not occur if the model depends on features whose relationship with the label does not change across settings; these are *stable* features.

Shortcut learning is well studied in cases where models that use both shortcut and stable features achieve lower loss than models that only use the stable feature [4, 5, 80]. These works consider cases where the Bayes-optimal classifier — the training conditional distribution of the label given the covariates — depends on both stable and shortcut features. In such cases, shortcut learning occurs as the Bayes-optimal predictor is the target of standard supervised learning algorithms such as the one that minimizes the log-loss via gradient descent (GD), which we call default-

ERM.

However, in many machine learning tasks, the stable feature perfectly predicts the label, i.e. a *perfect stable feature*. For example, in task of predicting hair color from images of celebrity faces in the CelebA dataset [16], the color of the hair in the image determines the label. This task is a perception task. In such classification tasks, the label is independent of the shortcut feature given the stable feature, and the Bayes-optimal predictor under the training distribution only depends on the stable feature. Default-ERM can learn this Bayes-optimal classifier which, by depending solely on the stable feature, also generalizes outside the training distribution. But in practice, default-ERM run on finite data yields models that depend on the shortcut and thus perform worse than chance outside the training distribution [16, 53, 81]. The question is, why does default-ERM prefer models that exploit the shortcut even when a model can achieve zero loss using the stable feature alone?

To understand preferences toward shortcuts, we study default-ERM on a linear perception task with a stable feature that determines the label and a shortcut feature that does not. The perfect linear stable feature means that data is linearly separable. This separability means that default-ERM-trained linear models classify in the same way as the minimum ℓ_2-norm solution that has all margins greater than 1; the latter is commonly called max-margin classification [82]. We prove that default-ERM's implicit inductive bias toward the max-margin solution is harmful in that default-ERM-trained linear models depend more on the shortcut than the stable feature. In fact, such dependence on the shortcut occurs even in the setting with fewer parameters in the linear model than data points, i.e. without overparameterization. These observations suggest that a max-margin inductive bias is unsuitable for perception tasks.

Next, we study inductive biases more suitable for perception tasks with perfect stable features. We first observe that predicting with the perfect stable feature alone achieves uniform margins on all samples. Formally, if the stable feature $s(\mathbf{x})$ determines the label $\mathbf{y}$ via a function d, $\mathbf{y} = d \circ s(\mathbf{x})$, one can achieve any positive b as the margin on all samples simultaneously by predicting with $b \cdot d \circ s(\mathbf{x})$. We show that in the same setting without overparameterization where max-margin classification leads to shortcut learning, models that classify with uniform margins depend only on the stable feature.

Building on these observations, we identify alternative loss functions that are inductively biased toward uniform margins, which we call margin control (MARG-CTRL). We empirically demonstrate that MARG-CTRL mitigates shortcut learning on multiple vision and language tasks without the use of annotations of the shortcut feature in training. Further, MARG-CTRL performs on par or better than the more expensive two-stage shortcut-mitigating methods [53, 81]. We then introduce a more challenging setting where both training and validation shortcut annotations are unavailable, called the nuisance-free setting. In the nuisance-free setting, MARG-CTRL *always outperforms* default-ERM and the two-stage shortcut-mitigating methods. These empirical results suggest that simply incorporating inductive biases more suitable for perception tasks is sufficient to mitigate shortcuts.

4.2 Shortcut learning in perception tasks due to maximizing margins

SETUP. We use $\mathbf{y}, \mathbf{z}, \mathbf{x}$ to denote the label, the shortcut feature, and the covariates respectively. We let the training and test distributions (p_{tr}, p_{te}) be members of a family of distributions indexed by ρ, $\mathcal{F} = \{p_\rho(\mathbf{y}, \mathbf{z}, \mathbf{x})\}_\rho$, such that the shortcut-label relationship $p_\rho(\mathbf{z}, \mathbf{y})$ changes over the family. Many common tasks in the spurious correlations literature have stable features $s(\mathbf{x})$ that are perfect, meaning that the label is a deterministic function d of the stable feature: $\mathbf{y} = d \circ s(\mathbf{x})$. For example, in the Waterbirds task the bird's body determines the label and in the CelebA task, hair color determines the label [16]. As $s(\mathbf{x})$ determines the label, it holds that $\mathbf{y} \perp\!\!\!\perp_{p_\rho} (\mathbf{x}, \mathbf{z}) \mid s(\mathbf{x})$. Then, the optimal predictor on the training distribution is optimal on all distributions in the family $\mathcal{F}$, regardless of the shortcut because $p_{tr}(\mathbf{y} \mid \mathbf{x}) = p_{tr}(\mathbf{y} \mid s(\mathbf{x})) = p_{te}(\mathbf{y} \mid s(\mathbf{x})) = p_{te}(\mathbf{y} \mid \mathbf{x})$.

The most common procedure to train predictive models to approximate $p_{tr}(\mathbf{y} \mid \mathbf{x})$ is gradient-based optimization of cross-entropy (also called log-loss); we call this default-ERM. Default-ERM targets the Bayes-optimal predictor of the training distribution which, in tasks with perfect stable features, also performs optimally under the test distribution. However, despite targeting the predictor that does not depend on the shortcut, models built with default-ERM still rely on shortcut features that are often less predictive of the label and are unstable, i.e. vary across distributions [5, 80]. We study default-ERM's preference for shortcuts in a data generating process (DGP) where both the shortcut and the perfect stable feature are linear functions of the covariates.

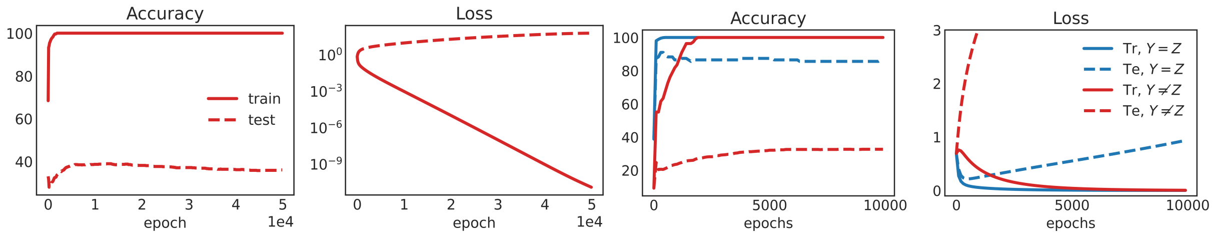
4.2.1 Shortcut learning in linear perception tasks

Let Rad be the uniform distribution over $\{1, -1\}$, $\mathcal{N}$ be the normal distribution, d be the dimension of $\mathbf{x}$, and $\rho \in (0, 1), B > 1$ be scalar constants. The DGP for $p_\rho(\mathbf{y}, \mathbf{z}, \mathbf{x})$ is:

$$\mathbf{y} \sim \text{Rad}, \quad \mathbf{z} \sim \begin{cases} p_\rho(\mathbf{z} = y \mid \mathbf{y} = y) = \rho \\ p_\rho(\mathbf{z} = -y \mid \mathbf{y} = y) = (1-\rho) \end{cases}, \quad \boldsymbol{\delta} \sim \mathcal{N}(0, \mathbf{I}^{d-2}), \quad \mathbf{x} = [B * \mathbf{z}, \mathbf{y}, \boldsymbol{\delta}]\,. \tag{4.1}$$

This DGP is set up to mirror the empirical evidence in the literature showing that shortcut features are typically learned first [16]. The first dimension of $\mathbf{x}$, i.e. $\mathbf{x}_1$, is a shortcut that is correlated with $\mathbf{y}$ according to ρ. The factor B in $\mathbf{x}_1$ scales up the gradients for parameters that interact with $\mathbf{x}_1$ in predictions. For large enough B, model dependence on the shortcut feature during default-ERM goes up faster than the stable feature [83].

The training distribution is $p_{tr} = p_{0.9}$ and the test distribution is one where the shortcut's relationship with the label is flipped $p_{te} = p_{0.1}$. Models achieve worse than random test accuracy (50%)



(a) Average accuracy and loss curves. **(b)** Accuracy/loss on shortcut and leftover groups.

Figure 4.1: Accuracy and loss curves for training a linear model with default-ERM on 1000 training samples from $p_{0.9}$, with $B = 10, d = 300$ (see eq. (4.1)), and testing on $p_{0.1}$. **(a)** The model achieves 100% train accuracy but $<$ 40% test accuracy. **(b)** The learned model achieves high test accuracy ($\approx$ 90%) on the shortcut group and low test accuracy on the leftover group ($\approx$ 30%). Models that depend more on the stable feature than on the shortcut, achieve at least 50% accuracy on both the shortcut and leftover groups. Hence the learned model exploits the shortcut to classify the shortcut group and overfits to the leftover group.

if they exploit the training shortcut relationship and the predicted class flips when the shortcut feature flips. We train with default-ERM which uses log-loss: on a data point $(\mathbf{x}, \mathbf{y})$ the log-loss is

$$\ell_{log}(\mathbf{y} f_\theta(\mathbf{x})) = \log\left[1 + \exp(-\mathbf{y} f_\theta(\mathbf{x}))\right].$$

With $d = 300$ and $B = 10$, we train a linear model on 1000 samples from the training distribution $p_{\rho=0.9}$, and evaluate on 1000 samples from $p_{\rho=0.1}$.

Observations. Figure 4.1(a) shows that when trained with default-ERM, the linear model does not do better than chance ($<$ 50%) on the test data even after 50,000 epochs. So, even in the presence of the perfect feature $\mathbf{x}_2$, the model relies on other features like the shortcut $\mathbf{x}_1$. Since the final training loss is very small, on the order of 10^{-9}, this result is not due to optimization being stuck in a local minima with high loss. **These observations indicate that, in the linear setting, gradient-based optimization with log-loss prefers models that depend more on the shortcut than the perfect stable feature.**

To better understand this preference we focus on the errors in specific groups in the data. Consider the classifier that only uses the shortcut $\mathbf{z}$ and makes the Bayes-optimal prediction w.r.t p_{tr}: $\arg\max_y p_{tr}(\mathbf{y} = y \mid \mathbf{z})$. We call instances that are classified correctly by this model the *shortcut* group, and the rest the *leftover* group. We use these terms for instances in the training set as well as the test set. In this experiment $\mathbf{y}$ is positively correlated with $\mathbf{z}$, hence the shortcut group consists of all instances with $\mathbf{y}^i = \mathbf{z}^i$ and the leftover group of those with $\mathbf{y}^i \neq \mathbf{z}^i$.

Figure 4.1(b) gives accuracy and loss curves on the shortcut and leftover groups for the first 10000 epochs. The test accuracy for the shortcut group hits 90% while the leftover group test accuracy is < 40%, meaning that the model exploits the shortcuts. Even though a model that relies solely on the shortcut misclassifies the leftover group, we see that the training loss of the learned model on this group approaches 0. The model drives down training loss in the leftover group by depending on noise, which results in larger test loss in the leftover group than the shortcut group. **Thus, fig. 4.1(b) demonstrates that the default-ERM-trained model classifies the training shortcut group by using the shortcut feature while overfitting to the training leftover group.**

Shortcut dependence like in fig. 4.1 occurs even with ℓ_2-regularization and when training neural networks; see appendix C.2.1 and appendix C.2.4 respectively. Next, we analyze the failure mode in fig. 4.1, showing that the shortcut dependence is due to default-ERM's implicit bias to learn the max-margin classifier. Next, we study the failure mode in fig. 4.1 theoretically, showing that the shortcut dependence is due to default-ERM's inductive bias toward learning the max-margin classifier.

MAX-MARGIN CLASSIFIERS DEPEND MORE ON THE THE SHORTCUT THAN THE STABLE FEATURE. We consider training a linear model $f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ where $\mathbf{w} = [\mathbf{w}_z, \mathbf{w}_y, \mathbf{w}_e]$ with default-ERM. Data from eq. (4.1) is always linearly separable due to the perfect stable feature, but many hyperplanes that separate the two classes exist. When a linear model is trained with default-ERM on linearly separable data, it achieves zero training loss and converges to the direction of a minimum ℓ_2-norm solution that achieves a margin of at least 1 on all samples [82, 84, 85]; this is called the max-margin solution. We now show that for a small enough leftover group, large enough scaling factor B and dimension d of the covariates, max-margin solutions depend more on the shortcut feature than the stable feature:

Theorem 3. *Let $\mathbf{w}^*$ be the max-margin predictor on n training samples from eq. (4.1) with a leftover group of size k. There exist constants $C_1, C_2, N_0 > 0$ such that*

$$\forall\, n > N_0, \qquad \forall \ \text{integers } k \in \left(0, \frac{n}{10}\right), \qquad \forall\, d \geq C_1 k \log(3n), \qquad \forall\, B > C_2\sqrt{\frac{d}{k}}, \tag{4.2}$$

with probability at least $1 - {}^{1}/{3n}$ over draws of the training data, it holds that $B\mathbf{w}_z^* > \mathbf{w}_y^*$.

The size of the leftover group k concentrates around $(1-\rho)n$ because each sample falls in the leftover group with probability $(1-\rho)$. Thus, for $\rho > 0.9$, that is for a strong enough shortcut, the condition in theorem 3 that $k < {}^{n}/{10}$ will hold with probability close to 1; see appendix C.1.5 for more details.

The proof is in appendix C.1. The first bit of intuition is that using the shortcut can have lower norm because of the scaling factor B. Using the shortcut only, however, misclassifies the leftover group. The next bit of intuition is that using noise from the leftover group increases margins in one group at a rate that scales with the dimension d, while the cost in the margin for the other group only grows as $\sqrt{d}$. This trade-off in margins means the leftover group can be correctly classified using noise without incorrectly classifying the shortcut group. The theorem then leverages convex duality to show that this type of classifier that uses the shortcut and noise has smaller ℓ_2-norm than any linear classifier that uses the stable feature more.

The way the margin trade-off in the proof works is by constructing a linear classifier whose weights on the noise features are a scaled sum of the product of the label and the noise vector in the leftover group: for a scalar γ, the weights $\mathbf{w}_e = \gamma \sum_{i \in S_{\text{leftover}}} \mathbf{y}_i \boldsymbol{\delta}_i$. The margin change on the jth training sample from using these weights is $\mathbf{y}_j \mathbf{w}_e^\top \boldsymbol{\delta}_j$. For samples in the shortcut group, the margin change looks like a sum of mean zero independent and identically distributed variables; the standard deviation of this sum grows as $\sqrt{d}$. For samples in the leftover group, the margin change is the sum of mean one random variables; this sum grows as d and its standard deviation grows as $\sqrt{d}$. The difference in mean relative to the standard deviation is what provides the trade-off in margins.

We now discuss three implications of the theorem.

First, theorem 3 implies that the leftover group sees worse than random accuracy (0.5). To see this, note that for samples in the leftover group the margin $\mathbf{y}(\mathbf{w}^*)^\top \mathbf{x} = \mathbf{w}_y^* - B\mathbf{w}_z^* + (\mathbf{w}_e^*)^\top \mathbf{y}\boldsymbol{\delta}$ is a Gaussian random variable centered at a negative number $\mathbf{w}_y^* - B\mathbf{w}_z^*$. Then, with Φ_e as the CDF of the zero-mean Gaussian random variable $(\mathbf{w}_e^*)^\top \boldsymbol{\delta}$, accuracy in the test leftover group is

$$p(\mathbf{y}(\mathbf{w}^*)^\top \mathbf{x} \geq 0 \mid \mathbf{y} \neq \mathbf{z}) = p[(\mathbf{w}_e^*)^\top \boldsymbol{\delta} > -(\mathbf{w}_y^* - B\mathbf{w}_z^*)] = 1 - \Phi_e(-(\mathbf{w}_y^* - B\mathbf{w}_z^*)) \leq 0.5.$$

Second, the leftover group in the training data is overfit in that the contribution of noise in prediction ($|(\mathbf{w}_e^*)^\top \boldsymbol{\delta}|$) is greater than the contribution from the stable and shortcut features. Formally, in the training leftover group, $\mathbf{w}_y^* - B\mathbf{w}_z^* < 0$. Then, due to max-margin property,

$$\mathbf{w}_y^* - B\mathbf{w}_z^* + (\mathbf{w}_e^*)^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \implies (\mathbf{w}_e^*)^\top \mathbf{y}_i \boldsymbol{\delta}_i \geq 1 - (\mathbf{w}_y^* - B\mathbf{w}_z^*) > |\mathbf{w}_y^* - B\mathbf{w}_z^*|.$$

Third, many works point to overparameterization as one of the causes behind shortcut learning [16, 86, 87], but in the setup in fig. 4.1, the linear model has fewer parameters than samples in the training data. In such cases with non-overparameterized linear models, the choice of default-ERM is typically not questioned, especially when a feature exists that linearly separates the data.

Corollary 1 formally shows shortcut learning for non-overparameterized linear models. In words, default-ERM — that is vanilla logistic regression trained with gradient-based optimization — can yield models that rely more on the shortcut feature *even without overparameterization.*

Corollary 1. *For all $n > N_0$ — where the constant N_0 is from theorem 3 — with scalar $\tau \in (0, 1)$ such that the dimension of $\mathbf{x}$ is $d = \tau n < n$, for all integers $k < n \times \min\left\{\frac{1}{10}, \frac{\tau}{C_1 \log 3n}\right\}$, a linear model trained via default-ERM yields a predictor $\mathbf{w}^*$ such that $B\mathbf{w}^*_z > \mathbf{w}^*_y$.*

If default-ERM produces models that suffer from shortcut learning even without overparameterization, its implicit inductive bias toward max-margin classification is inappropriate for perception tasks in the presence of shortcuts. Next, we study inductive biases more suited to perception tasks.

4.3 Toward inductive biases for perception tasks with shortcuts

The previous section formalized how default-ERM solutions, due to the max-margin inductive bias, rely on the shortcut and noise to minimize loss on training data even in the presence of a different zero-population-risk solution. Are there inductive biases more suitable for perception tasks?

Given a perfect stable feature $s(\mathbf{x})$ for a perception task, in that for a function d when $\mathbf{y} = d \circ s(\mathbf{x})$, one can achieve margin $b \in (0, \infty)$ uniformly on all samples by predicting with the stable $b \cdot d \circ s(\mathbf{x})$. In contrast, max-margin classifiers allow for disparate margins as long as the smallest margin crosses 1, meaning that it does not impose uniform margins. The cost of allowing disparate margins is the preference for shortcuts even without overparamterization (corollary 1). In the same setting however, any uniform-margin classifier for the linear perception task (eq. (4.1)) relies only on the stable feature:

Theorem 4. *Consider n samples of training data from* DGP *in* eq. (4.1) *with $d < n$. Consider a linear classifier $f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ such that for all samples in the training data $\mathbf{y}_i \mathbf{w}^\top \mathbf{x}_i = b$ for any $b \in (0, \infty)$. With probability 1 over draws of samples, $\mathbf{w} = [0, b, 0^{d-2}]$.*

Theorem 4 shows that uniform-margin classifiers only depend on the stable feature, standing in contrast with max-margin classifiers which can depend on the shortcut feature (theorem 3). The proof is in appendix C.1.6. **Thus, inductive biases toward uniform margins are better suited for perception tasks.** Next, we identify several ways to encourage uniform margins.

Margin control (marg-ctrl). To produce uniform margins with gradient-based optimization, we want the loss to be minimized at uniform-margin solutions and be gradient-optimizable. We identify a variety of losses that satisfy these properties, and we call them marg-ctrl losses. marg-ctrl losses have the property that per-sample loss monotonically decreases for margins until a threshold then increases for margins beyond it. In turn, minimizing loss then encourages all margins to move to the threshold.

Mechanically, when models depend more on shortcuts than the stable feature during training, margins on samples in the shortcut group will be larger than those in the leftover group; see the right panel in fig. 4.1(b) where the train loss in the shortcut group is lower than the leftover group indicating that the margins are smaller in the leftover group. This difference is margins is a consequence of the shortcut matching the label in one group and not the other, thus, encouraging the model to have similar margins across all samples pushes the model to depend less on the shortcut. In contrast, vanilla log-loss can be driven to zero in a direction with disparate margins across the groups as long as the margins on all samples go to ∞. We define marg-ctrl losses for a model f_θ with the margin on a sample $(\mathbf{x}, \mathbf{y})$ defined as $\mathbf{y} f_\theta(\mathbf{x})$.

As the first marg-ctrl loss, we develop the σ-damped log-loss: we evaluate log-loss on a mar-

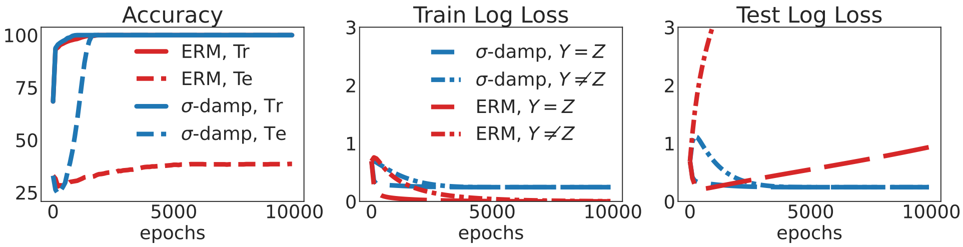


Figure 4.2: Using σ-damped log-loss yields linear models that depend on the perfect stable feature to achieve near perfect test accuracy. The middle panel shows that σ-damping maintains similar margins in the training shortcut and leftover groups unlike unconstrained log-loss, and the right panel shows σ-damp achieves better leftover test-loss.

gin multiplied by a monotonically decreasing function of the margin. In turn, the input to the loss increases with the margin till a point and then decreases. For a temperature T and sigmoid function σ, the σ-damped loss modifies the model output f_θ and plugs it into log-loss:

$$\ell_{\sigma\text{-damp}}(\mathbf{y}, f_\theta) = \ell_{\log}\left(\mathbf{y}\left(1 - \sigma\left(\frac{\mathbf{y}f_\theta}{T}\right)\right) f_\theta\right)$$

For large margin predictions $\mathbf{y}f_\theta > 0$, the term $1 - \sigma\left(\mathbf{y}f_\theta(\mathbf{x})/T\right)$ damps down the input to log-loss. The largest the input to $\ell_{\log}$ can get is $0.278T$, found by setting the derivative to zero, thus lower bounding the loss. As log-loss is a decreasing function of its input, the minimum of $\ell_{\sigma\text{-damp}}$ occurs when the margin is $0.278T$ on all samples. To demonstrate empirical advantage, we compare standard log-loss to σ-damped loss on eq. (4.1); see fig. 4.2. The left panel of figure fig. 4.2 shows that test accuracy is better for σ-damp. The middle and right panels shows the effect of controlling margins in training, where losses on shortcut and leftover groups hover at the same value.

Second, we design the σ-stitch loss, which imitates log-loss when $\mathbf{y}f_\theta(\mathbf{x}) < u$ and penalizes larger margins ($\mathbf{y}f_\theta > u$) by negating the sign of $\mathbf{y}f_\theta(\mathbf{x})$:

$$\ell_{\sigma\text{-stitch}} = \ell_{log}\left(\,\mathbf{1}[\mathbf{y}f_\theta(\mathbf{x}) \leq u]\mathbf{y}f_\theta(\mathbf{x}) \;+\; \mathbf{1}[\mathbf{y}f_\theta(\mathbf{x}) > u](2u - \mathbf{y}f_\theta(\mathbf{x}))\,\right) \tag{4.3}$$

As the third MARG-CTRL loss, we directly penalize large margins via a log-penalty:

$$\ell_{\texttt{marg-log}} = \ell_{log}(\mathbf{y}f_\theta(\mathbf{x})) + \lambda \log\left(1 + |f_\theta(\mathbf{x})|^2\right) \tag{4.4}$$

The fourth MARG-CTRL loss controls margins by penalizing $|f_\theta(\mathbf{x})|^2$:

$$\ell_{\text{SD}} = \ell_{log}(\mathbf{y}f_\theta(\mathbf{x})) + \lambda|f_\theta(\mathbf{x})|^2 \tag{4.5}$$

This last penalty was called SD by Pezeshki et al. [88], who use it as a way to decouple learning dynamics in the neural tangent kernel (NTK) regime. Instead, from the lens of MARG-CTRL, SD mitigates shortcuts in eq. (4.1) because it encourages uniform margins, even though SD was originally derived from different principles, as we discuss in section 4.5. In appendix C.2.2, we plot all MARG-CTRL losses and show that MARG-CTRL improves over default-ERM on the linear perception task; see figs. C.3 to C.5. We also run MARG-CTRL on a neural network and show that while default-ERM achieves test accuracy worse than random chance, MARG-CTRL achieves 100% test accuracy; see figs. C.7 to C.10 in appendix C.2.4.

4.4 Vision and language experiments

We evaluate MARG-CTRL on common datasets with shortcuts: Waterbirds, CelebA [16], and Civilcomments [79]. First, MARG-CTRL always improves over default-ERM. Then, we show that MARG-CTRL performs similar to or better than two-stage shortcut-mitigating methods like JTT [53] and CNC [81] in traditional evaluation settings where group annotations are available in the validation data. Finally, we introduce a more challenging setting that only provides class labels in training and validation, called the **nuisance-free setting**. In contrast to the traditional setting that always assumes validation group annotations, the nuisance-free setting does not provide group annotations in either training or in validation. In the nuisance-free setting, MARG-CTRL outperforms JTT and CNC, even though the latter are supposed to mitigate shortcuts without knowledge of the groups.

DATASETS. We use the Waterbirds and CelebA datasets from Sagawa et al. [16] and the CivilComments dataset from Koh et al. [79], Borkan et al. [89]. In Waterbirds, the task is to classify images of a waterbird or landbird, and the label is spuriously correlated with the image background consisting of land or water. There are two types of birds and two types of background, leading to a total of 4 groups defined by values of y, z. In CelebA [6, 16], the task is to classify hair color of celebrities as blond or not. The gender of the celebrity is a shortcut for hair color. There are two types of hair color and two genders in this dataset, leading to a total of 4 groups defined by values of y, z. In CivilComments-WILDS [79, 89], the task is to classify whether an online comment is toxic or non-toxic, and the label is spuriously correlated with mentions of certain demographic identities. There are 2 labels and 8 types of the shortcut features, leading to 16 groups.

METRICS, MODEL SELECTION, AND HYPERPARAMETERS. We report the worst-group test accuracy for each method. The groups are defined based on the labels and shortcut features. The more a model depends on the shortcut, the worse the worst-group error. Due to the label imbalance in all the datasets, we use variants of σ-damp, σ-stitch, MARG-LOG, and SD with class-dependent hyperparameters; see appendix C.2.6.2. For all methods, we use the standard Adam optimizer [90] and let the learning rate and weight decay hyperparameters be tuned along with the method's hyperparameters. We first report results for all methods using validation worst-group accuracy to select method and optimization hyperparameters and early stop. For both JTT and CNC, this is the evaluation setting that is used in existing work [53, 81, 83]. Finally, in the nuisance-free setting where no group annotations are available, we select hyperparameters using label-balanced average accuracy. Appendix C.2.6 gives further details about the training, hyperparameters, and experimental results.

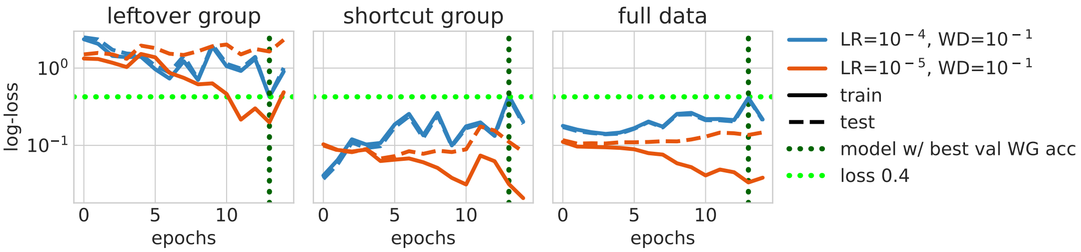


Figure 4.3: Loss curves of default-ERM on CelebA for two combinations of LR and WD. The combination with the larger learning rate (blue) achieves 72.8% test worst-group accuracy, beating the other combination by 20%. The model that achieves the best validation (and test) worst-group accuracy is the one at epoch 13 from the blue run. This model achieves similar loss in both groups and the full data model suggesting that large LR and WD controls margins from exploding (higher training loss in all panels) and avoids systematically smaller margins in the leftover group compared to the shortcut group.

4.4.1 Marg-ctrl mitigates shortcuts in the default setting

Here, we experiment in the standard setting from Liu et al. [53], Zhang et al. [81], Idrissi et al. [83] and use validation group annotations to tune hyperparameters and early-stopping.

Marg-ctrl improves over default-ERM. We compare MARG-CTRL to default-ERM on CelebA, Waterbirds, and Civilcomments. Table 4.1 shows that every MARG-CTRL method achieves higher test worst-group accuracy than default-ERM on all datasets. Default-ERM achieves a mean test worst-group accuracy of 70.8%, 72.8% and 60.1% on Waterbirds, CelebA, and Civilcomments respectively. Compared to default-ERM, MARG-CTRL methods provide a $5 - 10\%$ improvement on Waterbirds, $7 - 10\%$ improvement on CelebA, $7 - 10\%$ improvement on Civilcomments. These improvements show the value of inductive biases more suitable for perception tasks.

Large LR and WD may imitate MARG-CTRL in ERM. Default-ERM's performance varies greatly across different values of LR and WD on, for instance, CelebA: the test worst-group accuracy improves by more than 20 points over different LR and WD combinations. Why does tuning LR and WD yield such improvements? We explain this phenomenon as a consequence of instability in optimization induced by large LR and WD which prevents the model from maximizing margins and in turn can control margins. Figure 4.3 provides evidence for this explanation by comparing default-ERM's loss curves for two LR and WD combinations.

The blue loss curves in fig. 4.3 correspond to the run with the larger LR and WD combination. The model that achieves the best validation (and test) worst-group accuracy over all combinations of hyperparameters for default-ERM, including those not in the plot, is the one at epoch 13 on the blue curves. This model achieves similar train and test losses (≈ 0.4) and thus similar margins in the shortcut group, the leftover group, and the whole dataset. The red curves stand in contrast where the lower LR results in the leftover group having higher training and test losses, and therefore smaller margins, compared to the shortcut group. These observations together support the explanation that default-ERM with large LR and WD mitigates shortcuts when controlling margins like MARG-CTRL.

MARG-CTRL PERFORMS AS WELL OR BETTER THAN TWO-STAGE SHORTCUT-MITIGATING METHODS. Two-stage shortcut mitigating methods like Correct-n-Contrast (CNC) and Just Train Twice (JTT) aim to mitigate shortcuts by using a model trained with default-ERM to approximate group annotations. They rely on the assumption that a model trained via default-ERM either predicts with the shortcut feature (like background in Waterbirds) or that the model's representations separate into clusters based on the shortcut feature. The methods then approximate group annotations using this default-ERM-trained model and use them to mitigate shortcut learning in a second predictive model. JTT upweights the loss on the approximate leftover group and CNC uses a contrastive loss to enforce the model's representations to be similar across samples that have the same label but different approximate group annotations. Appendix C.2.6.1 gives details.

Table 4.1 compares MARG-CTRL to JTT and CNC on Waterbirds, Celeba, and CivilComments. On CelebA, SD, `marg-log`, and σ-stitch perform similar to CNC while all MARG-CTRL techniques outperform JTT. On Waterbirds, all MARG-CTRL methods outperform JTT and CNC. On CivilComments, σ-damp and SD perform similar to JTT and outperform CNC. CNC's performance on Wa-

	CelebA	WB	Civil
ERM	72.8 ± 9.4	70.8 ± 2.4	60.1 ± 0.4
CNC	81.1 ± 0.6	68.0 ± 1.8	68.8 ± 0.2
JTT	75.2 ± 4.6	71.7 ± 4.0	69.9 ± 0.4
marg-log	82.8 ± 1.1	78.2 ± 1.9	68.4 ± 1.8
σ-damp	79.4 ± 0.6	78.6 ± 1.1	69.6 ± 0.4
SD	81.4 ± 2.5	80.5 ± 1.4	69.9 ± 1.1
σ-stitch	81.1 ± 2.2	75.9 ± 3.4	67.8 ± 2.8

Table 4.1: Mean and standard deviation of test worst-group accuracies over two seeds for default-ERM, JTT, CNC, σ-damp, σ-stitch, SD, and marg-log. Every MARG-CTRL method outperforms default-ERM on every dataset. On Waterbirds, MARG-CTRL outperforms JTT and CNC. On CelebA, SD, marg-log, and σ-stitch beat JTT and achieve similar or better performance than CNC. On CivilComments, σ-damp and SD beat CNC and achieve similar performance to JTT.

terbirds differs from Zhang et al. [81] because their reported performance requires unique large WD choices (like WD set to 1) to build a first-stage model that relies most on the shortcut feature without overfitting to the training data.

MARG-CTRL IS FASTER THAN JTT AND CNC. MARG-CTRL takes the same time as default-ERM, taking around 1, 20 and 60 minutes per epoch for Waterbirds, CelebA, and CivilComments respectively on an RTX8000 GPU. In contrast, on average over runs, JTT takes around 6, 80, 120 minutes per epoch and CNC takes around 8, 180, 360 minutes per epoch. Thus, MARG-CTRL performs as well or better than JTT and CNC while being simpler to implement and computationally cheaper.

4.4.2 MARG-CTRL MITIGATES SHORTCUTS IN THE NUISANCE-FREE SETTING

Work like [53, 81] crucially require validation group annotations because these methods push the work of selecting models for mitigating shortcuts to validation. Determining shortcuts itself is a laborious manual process, which means group annotations will often be unavailable. Further, given a perfect stable feature that determines the label and a shortcut that does not, only models that rely on the stable feature more than the shortcut can achieve the highest validation accuracy. Thus, we introduce a more challenging setting that only provides class labels in training and

	CelebA	WB	Civil
ERM	57.5 ± 5.8	69.1 ± 2.1	60.7 ± 1.5
CNC	67.8 ± 0.6	60.0 ± 8.0	61.4 ± 1.9
JTT	53.3 ± 3.3	71.7 ± 4.0	53.4 ± 2.1
marg-log	74.2 ± 1.4	77.9 ± 0.3	66.8 ± 0.2
σ-damp	70.8 ± 0.3	74.8 ± 1.6	65.6 ± 0.2
SD	70.3 ± 0.3	78.7 ± 1.4	67.8 ± 1.3
σ-stitch	76.7 ± 0.6	74.5 ± 1.2	66.0 ± 1.0

Table 4.2: Average and standard deviation of test worst-group accuracy over two seeds of MARG-CTRL, default-ERM, JTT, and CNC in the nuisance-free setting. Hyperparameter selection and early stopping use label-balanced average accuracy. All MARG-CTRL methods outperform default-ERM, JTT, and CNC on all datasets.

validation, called the **nuisance-free setting**. In the nuisance-free setting, models are selected based on label-balanced average accuracy: the average of the accuracies over samples of each class.

Table 4.2 reports test worst-group (WG) accuracy in the nuisance-free setting. **On all the datasets, every MARG-CTRL outperforms default-ERM, JTT, and CNC.** On average, the MARG-CTRL methods close at least 61% of the gap between default-ERM in the nuisance-free setting and the best performance in table 4.1 on every dataset. **In contrast, CNC and JTT sometimes perform worse than default-ERM.**

4.5 Related work

A large body of work tackles shortcut learning under different assumptions [4, 12, 17, 24, 34, 80, 91, 92]. A different line of work focuses on learning in neural networks in idealized settings [93, 94, 95, 96, 97, 98, 99].

Shah et al. [100] study simplicity bias [101] and show that neural networks provably learn the linear function over a non-linear one, in the first epoch of training. In a similar vein, Hermann and Lampinen [102] show that neural networks can prefer a linearly-decodable feature over a non-linear but more predictive feature, and Scimeca et al. [103] make similar observations and use loss landscapes to empirically study which features are easier to learn. Simplicity bias alone only describes neural biases early in training and does not explain why more predictive stable features are not learned later. Unlike simplicity bias which focuses on linear versus non-linear features, max-margin bias is the reason default-ERM prefers one linear feature, the shortcut, over another, the stable feature, like in the synthetic experiment in section 4.2.

While Pezeshki et al. [88] allow for perfect features, they hypothesize that shortcut learning occurs because when one feature is learned first, other features are gradient-starved and are not learned as well. They focus on a special setting where feature representations for different samples have inner product equal to a small constant to show that models can depend more on the imperfect feature than the perfect feature. In this special setting, they show that penalizing the magnitudes of what we call the margin mitigates shortcuts; this method is called spectral decoupling (SD). However, as we show in appendix C.2.5, the assumption in Lemma 1 [88] is violated when using a linear model to classify in the simple linear DGP in eq. (4.1). However, SD on a linear model mitigates shortcuts in the DGP in eq. (4.1); see C.2.5. Thus, the theory in Pezeshki et al. [88] fails to not explain why SD works for eq. (4.1), but the uniform-margin property explains why all the MARG-CTRL losses, including SD, mitigate shortcuts.

Nagarajan et al. [86] consider tasks with perfect stable features and formalize geometric properties of the data that make max-margin classifiers give non-zero weight to the shortcut feature ($\mathbf{w}_z > 0$). In their set up, the linear models are overparameterized and it is unclear when $\mathbf{w}_z > 0$ leads to worse-than-random accuracy in the leftover group because they do not separate the model's dependence on the stable feature from the dependence on noise. See fig. C.11 for an example where $\mathbf{w}_z > 0$ but test accuracy is 100%. In contrast to Nagarajan et al. [86], theorem 3 gives a family of DGPs where the leftover group accuracy is worse than random, even without overparameterization. Ahuja et al. [104] also consider linear classification with default-ERM with a perfect stable feature and conclude that default-ERM learns only the stable feature because they assume no additional dimensions of noise in the covariates. We develop the necessary nuance here by including noise in the problem and showing default-ERM depends on the shortcut feature even without overparameterization.

Sagawa et al. [7] and Wald et al. [87] both consider overparameterized settings where the shortcut feature is informative of the label even after conditioning on the stable feature. In both cases, the Bayes-optimal predictor also depends on the shortcut feature, which means their settings do not allow for an explanation of shortcut dependence in examples like fig. 4.1. In contrast, we show shortcut dependence occurs even in the presence of a perfect stable feature and without overparameterization. Li et al. [105], Pezeshki et al. [106] focus on relative feature complexity and discuss the effects of large LR on which features are learned first during training, but do not allow for perfect features. Idrissi et al. [83] empirically find that tuning LR and WD gets default-ERM to perform similar to two-stage shortcut-mitigating methods like JTT [53]. We view the findings of [83] through the lens of MARG-CTRL and explain how large LR and WD approximate MARG-CTRL to mitigate shortcuts; see section 4.4.

MARG-CTRL is related to but different from methods proposed in Liu et al. [107], Cao et al. [108], Kini et al. [109]. These works normalize representations or the last linear layers and linearly transform the logits to learn models with better margins under label imbalance. Next, methods like Learning from Failure (LFF) [56], JTT [53], and CNC [81] build two-stage procedures to avoid shortcut learning without group annotations in training. They assume that default-ERM produces models that depend more on the shortcut and select hyperparamters of the two stage process using validation group annotations. In the nuisance-free setting where there are no validation group annotations, the performance of these methods can degrade below that of default-ERM. In contrast, better characterizing the source of shortcut learning in perceptual problems leads to MARG-CTRL methods that are not as reliant on validation group annotations (see nuisance-free results in Section 4.4). **Without any group annotations, encouraging uniform margins via MARG-CTRL mitigates shortcuts better than JTT and CNC.**

Soudry et al. [82] characterize the inductive bias of gradient descent to converge in direction to max-margin solutions when using exponentially tailed loses; Wang et al. [84, 85] then prove similar biases toward max-margin solutions for Adam and RMSProp. Ji et al. [110] show that for general losses that decrease in $\mathbf{y}f_\theta(\mathbf{x})$, gradient descent has an inductive bias to follow the ℓ_2-regularization path. All these inductive biases prefer shortcuts if using them leads to lower loss within an ℓ_2-norm-budget. Marg-ctrl provides a different inductive bias toward producing the same margin on all samples, which means gradient descent veers models away from imperfect shortcuts that lead to disparity in network outputs. Such inductive biases are suitable for tasks where a feature determines the label ($h(\mathbf{x}) = \mathbf{y}$).

4.6 Discussion

We study why default-ERM — gradient-based optimization of log-loss — yields models that depend on the shortcut even when the population minimum of log-loss is achieved by models that depend only on the stable feature. By studying a linear task with perfect stable features, we show that default-ERM's preference toward shortcuts sprouts from an inductive bias toward maximizing margins. Instead, inductive biases toward uniform margins improve dependence on the stable feature and can be implemented via MARG-CTRL. Marg-ctrl improves over default-ERM on a variety of perception tasks in vision and language without group annotations in training, and is competitive with more expensive two-stage shortcut-mitigating methods. In the nuisance-free setting, where even validation group annotations are unavailable, MARG-CTRL outperforms all the baselines. The performance that Marg-ctrl yields demonstrates that changing inductive biases can remove the need for expensive shortcut-mitigating methods in perception tasks.

Without overparameterization, uniform-margin classifiers are unique and learn stable features only, while max-margin classifiers can depend more on shortcuts. With overparameterization, max-margin classifiers are still unique but uniform-margin solutions are not which necessitates choosing between solutions. The experiments in section 4.4 suggest that choosing between uniform-margin classifiers with penalties like ℓ_2 improves over max-margin classifiers with ℓ_2: all experiments use overparameterized models trained with weight decay and MARG-CTRL outperforms default-ERM. Further, our experiments suggest that uniform-margin classifiers are insensitive to the WD and LR choices, unlike max-margin classifiers; appendix C.2.8 shows that MARG-CTRL achieves high performance for all LR and WD choices but ERM requires tuning.

Theorem 3 also explains how balancing may or may not improve dependence on the stable features. For example, a weighting-based approach produces the same max-margin solution as default-ERM [7, 111], but subsampling leads to a different solution that could depend less on the shortcut. For the latter however, models are more prone to overfitting on the smaller subsampled dataset. Similar observations were made in [7] but this work extends the insight to tasks with perfect stable features. Comparing ERM and MARG-CTRL on subsampled data would be fruitful.

Any exponentially tailed loss when minimized via gradient descent converges to the max-margin solution in direction [82]. Thus, theorem 3 characterizes shortcut learning for any exponentially-tailed loss. However, losses with decreasing polynomial tails — for example, $\ell(a) = \frac{1}{1+a^K}$ for some $K > 0$ — do not converge to the max-margin classifier. One future direction is to show shortcut-dependence results like theorem 3 for polynomial-tailed losses, which in turn would mean that all common classification losses with a decreasing tail impose inductive biases unsuitable for perception tasks.

In the tasks we consider with perfect stable features, Bayes-optimal predictors rely only on the stable feature. A weaker independence condition implies the same property of Bayes-optimal predictors even when $\mathbf{y}$ is not determined by $s(\mathbf{x})$: $\mathbf{y} \perp\!\!\!\perp (\mathbf{x}, \mathbf{z}) \mid s(\mathbf{x})$. For example, in the CivilComments dataset a few instances have ambiguous labels [112] meaning that there may not be a perfect stable feature. Studying uniform margins and other inductive biases under this independence would be fruitful.

Part II

Generalizing Causal Estimation

5 General Control Functions for Causal Estimation from Instrumental Variables

5.1 Introduction

Many disciplines use observational data to estimate causal effects: economics [113], sociology [114], psychology [115], epidemiology [116], and medicine [117]. Estimating causal effects with observational data requires care due to the presence of confounders that influence both treatment and outcome. Observational causal estimators deal with confounders in one of two ways. One, they assume that all confounders are observed; an assumption called *ignorability*. Two, they assume a source of external randomness that has a direct influence only on the treatment. Such a source is called an instrumental variable (IV) [118, 119]. An example is college proximity as an IV to study effects of education [120].

Two common IV-based causal effect estimation methods are the two-stage least-squares method (2SLS) [121, 122, 123] and the traditional control function method (CFN) [119, 124, 125, 126]. Both methods have a common first stage: learn a distribution over the treatment conditioned on the IV. In the second stage, 2SLS regresses the outcome on simulated treatments from the first stage, while CFN's second stage regresses the outcome on the true treatment and the error in the prediction of treatment from the first stage. The prediction error can be used to control for confounding and is thus called a *control function.* Though widely used, both 2SLS and CFN breakdown under certain conditions like, for example, when the outcome depends on multiplicative interactions of treatment and confounders. Further, CFN requires an additional assumption about the correlations between noise and outcome.

We study causal estimation with control functions. To estimate effects, control functions must satisfy ignorability. Our meta-identification result (theorem 5) shows that a control function satisfies ignorability if 1) the control function and IV together reconstruct the treatment, and 2) the confounder and control function together are jointly independent of the IV. We will refer to such control functions as *general control functions.* Effect estimation in general requires that the treatment has a chance to take any value given the control function; this is called positivity. We show positivity for general control functions holds if the IV can set treatment to any value; we call this a *strong* IV.

Any general control function uniquely determines the effect because it satisfies ignorability and positivity (given a strong IV). Causal identification requires effects to be uniquely determined by the observed data distribution. Thus, building general control functions using observed data guarantees causal identification. As reconstruction and *marginal* independence are properties of the joint distribution over observed data and control function, they can be guaranteed. Guaranteeing *joint* independence requires further assumptions as it involves the *unobserved* confounder. We show that structural assumptions on the treatment process, such as treatment being an additive function of the confounder and IV, help ensure joint independence.

To build general control functions and use them to estimate effects, we develop the general control function method (GCFN). GCFN's first stage, called variational decoupling (VDE), constructs the general control function. VDE is a type of autoencoder where the encoder constructs the control function and the decoder reconstructs treatment from control function and IV, under the constraint that the control function and IV are independent. When VDE is perfectly solved with a decoder that reflects a structural treatment process assumption, like additivity, reconstruction and joint independence are guaranteed. Thus with a strong IV, ignorability and positivity hold which implies identification, and that effect estimation does not require structural assumptions on the *outcome process* like those in 2SLS and CFN. Using VDE's general control function, GCFN's second stage estimates the causal effect. GCFN's second stage can be any method that relies on ignorability like matching/balancing methods [127, 128, 129] and doubly-robust methods [130].

We also consider a setting where a subset of the data has observed confounders that provide ignorability. We develop semi-supervised GCFN to estimate effects in this setting. Semi-supervised GCFN's first stage is an augmented VDE that forces the control function to match the confounder in the subset where it is observed. This augmented VDE helps guarantee joint independence even with a decoder that does not reflect structural treatment process assumptions.

In section 5.4, we evaluate GCFN's causal effect estimation on simulated data with the outcome, treatment, and IV observed. We demonstrate how GCFN produces correct effect estimates without additional assumptions on the true outcome process, whereas 2SLS, CFN, and DeepIV [131] fail to produce the correct estimate. Further, we show that GCFN performs on par with recently proposed methods DeepGMM [132] and DeepIV [131] on high-dimensional simulations from each respective paper. We also demonstrate that in data with a small subset having observed confounders, semi-supervised GCFN outperforms outcome regression on treatment and confounder within the subset. We also show recovery of the effect of slave export on current societal trust [2].

Related Work. Classical examples of methods that use IVs include the Wald estimator [133], two-stage least-squares method (2SLS) [121, 122, 123] and control function method (CFN) [119, 124, 125, 126]. The Wald estimator assumes constant treatment effect. 2SLS's estimation could be biased when the outcome generating process has multiplicative interactions between treatment and confounders (appendix D.1.10). Guo and Small [134] proved that under some assumptions, CFN improves upon 2SLS. Beyond these classical estimators, Wooldridge [124] discusses extensions of regression residuals for non-linear models under distributional assumptions about the noise in the treatment process. Hartford et al. [131] developed DeepIV, a deep variant of 2SLS and Singh et al. [135] kernelized the 2SLS algorithm. An alternative to 2SLS is the generalized method of moments (GMM) [136] which solves moment equations implied by the independence of the confounder and the IV. Bennett et al. [132] develop a minimax GMM and use neural networks to specify moment conditions.

Given only an IV, treatment, and outcome, causal effects are not identifiable without further assumptions [137, 138]. Newey [139] and Chetverikov and Wilhelm [140] assume additive outcome processes, where the outcome process is a sum of the causal effect and zero-mean noise; such models are also called separable. Identification in separable models relies on the *completeness* condition [141] which requires the conditional distribution of treatment given IV to sufficiently vary with the IV. Newey [139], Chetverikov and Wilhelm [140] discuss non-parametric estimators under assumptions of monotonicity of the treatment process and shape of causal effects (for eg. U-shaped). We focus on the setting where the outcome process *cannot* be represented as a sum of the causal effect and noise, often called a non-separable model [142]. Imbens and Newey [143] showed effect identification in non-separable models when the treatment has a continuous strictly monotonic cumulative distribution function (CDF) given the IV. Under this same condition, we can guarantee joint independence via a strictly monotonic reconstruction map which means identification holds.

5.1.1 Review of IVs and traditional control function theory

To define the causal effect we use causal graphs [144]. In causal graphs, each variable is represented by a node, and each causal relationship is a directed arrow from the cause to the effect. Causal graphs get transformed by interventions with the do-operator. The shared relationships between the graphs before and after the do-operation make estimation possible. The causal effect of giving a treatment $\mathbf{t} = a$ on an outcome $\mathbf{y}$ is $\mathbb{E}[\mathbf{y} \mid \mathrm{do}(\mathbf{t} = a)]$. The causal graph in Figure 5.1 describes a broad class of IV problems. The difficulty of causal estimation in this graph stems from the unobserved confounder $\mathbf{z}$. The IV $\boldsymbol{\epsilon}$ helps control for $\mathbf{z}$. Two popular IV-based methods are the two-stage least-squares method (2SLS) and control function method (CFN).

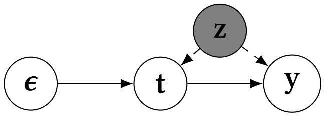


Figure 5.1: Causal graph with hidden confounder $\mathbf{z}$, outcome $\mathbf{y}$, instrument $\boldsymbol{\epsilon}$, treatment $\mathbf{t}$.

We follow the CFN setup from Guo and Small [134], where the true outcome and treatment processes have additive zero-mean noise called $\boldsymbol{\eta}_{\mathrm{y}}$ and $\boldsymbol{\eta}_{\mathrm{t}}$ that may be correlated due to $\mathbf{z}$:

$$\mathbf{y} = f(\mathbf{t}) + \boldsymbol{\eta}_{\mathrm{y}}, \quad \mathbf{t} = g(\boldsymbol{\epsilon}) + \boldsymbol{\eta}_{\mathrm{t}}. \tag{5.1}$$

To estimate the causal effect, the CFN method constructs a control function with the regression residual $\mathbf{t} - \hat{g}(\boldsymbol{\epsilon})$. Then, CFN regresses the outcome $\mathbf{y}$ on the regression residual and the treatment $\mathbf{t}$. The causal effect is the estimate of the function $f(\mathbf{t})$. For this estimate to be valid, the CFN method assumes that $\boldsymbol{\eta}_{\mathrm{t}}, \boldsymbol{\eta}_{\mathrm{y}}$ satisfy the following property for some constant ρ, (A4 in [134]):

$$\mathbb{E}[\boldsymbol{\eta}_{\mathrm{y}} \mid \boldsymbol{\eta}_{\mathrm{t}} = \eta] = \rho\eta \tag{5.2}$$

This property restricts the applicability of the CFN method by limiting how confounders influence the outcome and the treatment. Consider the following additive noise example: $\boldsymbol{\epsilon}, \mathbf{z} \sim \mathcal{N}(0, 1)$, $\mathbf{t} = \mathbf{z} + \boldsymbol{\epsilon}$, $\mathbf{y} \sim \mathcal{N}(\mathbf{t}^2 + \mathbf{z}^2, 1)$, where $\mathcal{N}$ is the standard normal. Here $\boldsymbol{\eta}_{y} = \mathbf{z}^2$ and $\boldsymbol{\eta}_{\mathrm{t}} = \mathbf{z}$ meaning that $\mathbb{E}[\boldsymbol{\eta}_{\mathrm{y}} \mid \boldsymbol{\eta}_{\mathrm{t}} = \eta] = \eta^2$, violating the assumption in eq. (5.2). Note that

$\mathbb{E}[\mathbf{z}\mathbf{t}^2] = \mathbb{E}[\mathbf{z}\mathbf{z}^2] = 0$, however $\mathbb{E}[\mathbf{t}^2\mathbf{z}^2] > 0$. This means regressing $\mathbf{y}$ on $\mathbf{t}^2$ and $\mathbf{z}$, i.e., with the correct model for $f(\mathbf{t})$, would result in an inflated coefficient of $\mathbf{t}^2$, which is an incorrect causal estimate. Equation (5.2) is required because some specified function of $\mathbf{t}$ could be correlated with an unspecified function of $\mathbf{z}$, resulting in a biased causal estimate. See appendix D.1.10 for an example where 2SLS produces biased effect estimates. The assumption in eq. (5.1) restricts the confounder's influence to be additive on both the treatment and outcome. Further, CFN assumes that the average additive influence the confounder has on the outcome to be a scaled version of the confounder's influence on the treatment (eq. (5.2)). Such assumptions may not hold in real data. For example, the effect of a medical treatment on patient lifespan is confounded by the patient's current health. This confounder influences the treatment through a human decision process, while it influences the outcome through a physiological process making it unlikely to meet CFN's assumptions.

5.2 Causal Identification with General Control Functions

With a control function that satisfies ignorability and positivity, causal estimation reduces to regression of the outcome on the treatment and the control function. We characterize such control functions:

Theorem 5. (Meta-identification result for control functions)
Let $F(\mathbf{t}, \boldsymbol{\epsilon}, \mathbf{y})$ be the true data distribution. Let control function $\hat{\mathbf{z}}$ be sampled conditionally on $\mathbf{t}, \boldsymbol{\epsilon}$. Let $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}) = q(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})F(\mathbf{t}, \boldsymbol{\epsilon})$ be the joint distribution over $\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}$. Further, let g be a deterministic function and $\boldsymbol{\delta}$ be independent noise such that $\mathbf{t} = g(\mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\delta})$ and let the implied true joint be $F'(\mathbf{t}, \mathbf{z}, \boldsymbol{\delta})$. Assume the following:

1. *(A1) $\hat{\mathbf{z}}$ satisfies the **reconstruction** property: $\exists d,\ \hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon} \sim q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}) \implies \mathbf{t} = d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$.*

2. *(A2) The IV is* ***jointly independent*** *of control function, true confounder, and noise* $\boldsymbol{\delta}$: $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}}, \boldsymbol{\delta})$.

3. *(A3)* ***Strong*** *IV. For any compact* $B \subseteq supp(\mathbf{t})$, $\exists c_B$ *s.t. a.e.* $t \in B$, $F'(\mathbf{t} = t \mid \mathbf{z}, \boldsymbol{\delta}) \geq c_B > 0$.

Then, the control function $\hat{\mathbf{z}}$ *satisfies ignorability and positivity:*

$$q(\mathbf{y} \mid \mathbf{t} = t, \hat{\mathbf{z}}) = q(\mathbf{y} \mid \text{do}(\mathbf{t} = t), \hat{\mathbf{z}}) \qquad \text{a.e. in } supp(\mathbf{t}) \quad q(\hat{\mathbf{z}}) > 0 \implies q(\mathbf{t} = t \mid \hat{\mathbf{z}}) > 0.$$

Therefore, the true causal effect is uniquely determined by $q(\hat{\mathbf{z}}, \mathbf{t}, \mathbf{y})$ *for almost every* $t \in supp(\mathbf{t})$*:*

$$\mathbb{E}_{\hat{\mathbf{z}}}[\mathbf{y} \mid \mathbf{t} = t, \hat{\mathbf{z}}] = \mathbb{E}_{\hat{\mathbf{z}}}[\mathbf{y} \mid \text{do}(\mathbf{t} = t), \hat{\mathbf{z}}] = \mathbb{E}[\mathbf{y} \mid \text{do}(\mathbf{t} = t)].$$

Theorem 5 characterizes functions of treatment and IV that satisfy reconstruction (A1) and joint independence (A2) which we call *general control functions.* Positivity of $\mathbf{t}$ w.r.t. the general control function holds under an assumption about the treatment process that the IV is strong (A3). Ignorability and positivity w.r.t. $\hat{\mathbf{z}}$ imply that the true causal effect is uniquely determined as a function of the observed data distribution $q(\hat{\mathbf{z}}, \mathbf{t}, \mathbf{y})$ [*]. If A1 and A2 are satisfied by the observed data distribution $q(\hat{\mathbf{z}}, \mathbf{t}, \mathbf{y}, \boldsymbol{\epsilon})$, the true effect is uniquely determined by the observed data distribution and thus causal identification holds. However, joint independence (A2) relies on the *unobserved* true confounder $\mathbf{z}$. So, theorem 5 is a *meta*-identification result because it does not specify how to guarantee joint independence using $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon})$. In section 5.2.1, we discuss structural assumptions on the treatment process that instantiate this meta-result and guarantee identification.

Theorem 5 holds for both discrete and continuous $\mathbf{t}$ given that the causal effect exists for all $t \in \text{supp}(\mathbf{t})$.[†] While we focus on the causal effect $\mathbb{E}[\mathbf{y} \mid \text{do}(\mathbf{t})]$, theorem 5 guarantees any property of $\mathbf{y} \mid \text{do}(\mathbf{t})$ can be estimated; for e.g. quantile treatment effects. For ease of exposition, we

[*] We also require that $\mathbb{E}_{\hat{\mathbf{z}}}\mathbb{E}[\mathbf{y} \mid \text{do}(\mathbf{t}), \hat{\mathbf{z}}]$ exists. This is guaranteed if the causal effect $\mathbb{E}[\mathbf{y} \mid \text{do}(\mathbf{t})] = \mathbb{E}_{\mathbf{z}}\mathbb{E}[\mathbf{y} \mid \mathbf{t}, \mathbf{z}]$ exists as ignorability holds w.r.t. $\hat{\mathbf{z}}$: $\mathbb{E}_{\hat{\mathbf{z}}}\mathbb{E}[\mathbf{y} \mid \text{do}(\mathbf{t}), \hat{\mathbf{z}}] = \mathbb{E}_{\hat{\mathbf{z}}}\mathbb{E}_{q(\mathbf{z} \mid \hat{\mathbf{z}})}\mathbb{E}[\mathbf{y} \mid \mathbf{t}, \mathbf{z}] = \mathbb{E}_{\mathbf{z}}\mathbb{E}[\mathbf{y} \mid \mathbf{t}, \mathbf{z}]$.

[†] Effects for certain treatments can be identified even without the strong IV assumption (A3): for any compact subset $B \subseteq \text{supp}(\mathbf{t})$ such that $\forall t \in B$, $F'(\mathbf{t} = t \mid \mathbf{z}, \boldsymbol{\delta}) \geq c_B > 0$, effects can be estimated for all $t \in B$.

restrict ourselves to treatments of the form $\mathbf{t} = g(\boldsymbol{\epsilon}, \mathbf{z})$, without noise $\boldsymbol{\delta}$. Then, theorem 5 requires only $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}})$. In appendix D.1.6, we show $\boldsymbol{\epsilon} \perp\!\!\!\perp (\hat{\mathbf{z}}, \mathbf{z}, \boldsymbol{\delta})$ is guaranteed for more general treatment processes of the form $\mathbf{t} = g(\boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta}))$. Guaranteeing joint independence requires further conditions and is the central challenge in developing two-stage IV-based estimators.

Why joint independence? A potential outcome $\mathbf{y}_{\mathbf{t}}$ is the outcome that would be observed if a unit is given treatment $\mathbf{t}$. The potential outcome $\mathbf{y}_{\mathbf{t}}$ follows the distribution of $\mathbf{y}$ under the *do* operator and only depends on the true confounder $\mathbf{z}$. For ignorability with respect to $\hat{\mathbf{z}}$, we need $\mathbf{y}_{\mathbf{t}}$ to be independent of $\mathbf{t}$, given $\hat{\mathbf{z}}$. By reconstruction, given $\hat{\mathbf{z}}$, $\mathbf{t}$ is purely a function of $\boldsymbol{\epsilon}$. This means ignorability with respect to the control function $\hat{\mathbf{z}}$ requires that the true confounder and IV be independent given the control function. Therefore, ignorability requires $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}$. Further, conditional independence $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \mathbf{z}$ implies positivity of $\mathbf{t}$ w.r.t $\hat{\mathbf{z}}$ if $\boldsymbol{\epsilon}$ is strong. Joint independence $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}})$ implies both the conditional independencies above.

The causal graph fig. 5.1 with $\mathbf{y}$ marginalized out can be represented with two sources of randomness one from the unobserved confounder $\mathbf{z}$ and one from the IV $\boldsymbol{\epsilon}$; the extra randomness in $\mathbf{t}$ denoted as $\boldsymbol{\delta}$ can be absorbed into $\mathbf{z}$. In this setup, the treatment and control function are deterministic functions of the unobserved confounder and IV. With only two sources of randomness, joint independence means the control function $\hat{\mathbf{z}}$ needs to only be a function of the true unobserved confounder $\mathbf{z}$. When $\hat{\mathbf{z}}$ is a stochastic function of the treatment and IV, joint independence holds if $\hat{\mathbf{z}}$ determines $\mathbf{z}$ while $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$.

As $\hat{\mathbf{z}}$ and $\boldsymbol{\epsilon}$ are observed, we can guarantee $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$. The marginal independence $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon}$ holds by definition of an IV. However, even both marginal independencies $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$ and $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon}$ together do not imply joint independence $\boldsymbol{\epsilon} \perp\!\!\!\perp (\hat{\mathbf{z}}, \mathbf{z})$. This means a control function $\hat{\mathbf{z}}$ that satisfies the reconstruction property and marginal independence $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$ may fail to yield ignorability. In appendix D.1.4, we build an example of a deterministic almost everywhere invertible function of two independent variables $\mathbf{c} = f(\mathbf{a}, \mathbf{b})$ such that $\mathbf{c} \perp\!\!\!\perp \mathbf{a}$ and $\mathbf{c} \perp\!\!\!\perp \mathbf{b}$ and yet, joint independence $(\mathbf{c}, \mathbf{b}) \not\!\perp\!\!\!\perp \mathbf{a}$ is violated.

As $\mathbf{z}$ is unobserved, achieving joint independence requires further assumptions. Next, we discuss how structural assumptions on the true treatment process can help guarantee joint independence.

5.2.1 Guaranteeing joint independence for identification

We show structural treatment process assumptions help guarantee joint independence by relating it to $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon})$ and thus giving identification. Joint independence can be guaranteed (via marginal independence) if the reconstruction map $d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$ (A1, theorem 5) reflects the functional structure of the treatment process. As an example, consider an additive treatment process $\mathbf{t} = \mathbf{z} + g(\boldsymbol{\epsilon})$. If the reconstruction map is $d(\hat{\mathbf{z}}, \boldsymbol{\epsilon}) = h'(\hat{\mathbf{z}}) + g'(\boldsymbol{\epsilon})$ and $\boldsymbol{\epsilon} \perp\!\!\!\perp \hat{\mathbf{z}}$, joint independence holds. To see this, note

$$h'(\hat{\mathbf{z}}) - \mathbb{E}_{\hat{\mathbf{z}}}[h'(\hat{\mathbf{z}})] = \mathbf{t} - \mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = \mathbf{z} - \mathbb{E}_{\mathbf{z}}[\mathbf{z}] \implies \exists \text{ constant } c,\ h'(\hat{\mathbf{z}}) = \mathbf{z} + c, \tag{5.3}$$

meaning $h'(\hat{\mathbf{z}})$ determines $\mathbf{z}$. By $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$, it holds that $q(\hat{\mathbf{z}}, \mathbf{z} \mid \boldsymbol{\epsilon}) = q(\hat{\mathbf{z}}, h'(\hat{\mathbf{z}}) - c \mid \boldsymbol{\epsilon}) = q(\hat{\mathbf{z}}, \mathbf{z})$. Thus, leveraging the functional structure of the treatment process helps guarantee joint independence by relating it to $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon})$, via $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$. Assuming treatment gets generated from other *known* invertible functions, such as multiplication $\mathbf{t} = h(\mathbf{z}) * g(\boldsymbol{\epsilon})$, also leads to joint independence. Imbens and Newey [143] proved effect identification when the treatment is a continuous strictly monotonic function of the confounder; these conditions helps guarantee joint independence (see appendix D.1.7). For more general treatments of the form $\mathbf{t} = g(\boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta}))$ the structural assumptions from above can only guarantee $(h(\mathbf{z}, \boldsymbol{\delta}), \hat{\mathbf{z}}) \perp\!\!\!\perp \boldsymbol{\epsilon}$; see appendix D.1.5 for general additive treatments: $\mathbf{t} = h(\mathbf{z}, \boldsymbol{\delta}) + g(\boldsymbol{\epsilon})$. However, we show in appendix D.1.6 that for such general treatment processes $(h(\mathbf{z}, \boldsymbol{\delta}), \hat{\mathbf{z}}) \perp\!\!\!\perp \boldsymbol{\epsilon} \implies (\mathbf{z}, \hat{\mathbf{z}}, \boldsymbol{\delta}) \perp\!\!\!\perp \boldsymbol{\epsilon}$ which, together with reconstruction, implies ignorability(theorem 5). In summary, under certain structural assumptions, general control functions exist ($\hat{\mathbf{z}} = \mathbf{z}$ for example) and can be built using only properties of the observed data distribution $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon})$. This guarantees identification. In section 5.3, we develop practical algorithms to build general control functions.

5.2.2 Comparison of identification with general control functions to existing work

Traditional CFN theory [134] relies on the assumption that the treatment process is additive; recall $\mathbf{t} = g(\boldsymbol{\epsilon}) + \boldsymbol{\eta}_t$ from section 5.1.1 and $\boldsymbol{\eta}_t$ is correlated with outcome noise due to $\mathbf{z}$. Beyond this additivity assumption, traditional CFN theory further assumes 1) the outcome process is additive, like in eq. (5.1), 2) the noise $\boldsymbol{\eta}_y$ in the outcome process is independent of the IV, 3) linear noise relationship between $\boldsymbol{\eta}_t, \boldsymbol{\eta}_y$, like in eq. (5.2), and 4) (relevance) the treatment effect function and IV are correlated [134]. When the treatment process is additive, joint independence can be guaranteed as a property of the distribution $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon})$, via $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$; see section 5.2.1. Then, identification with general control functions requires a strong IV. While it allows structural outcome process assumptions (like 3) can be relaxed, a strong IV needs more than the two IV properties, independence with confounder and relevance. However, domain expertise helps reason about strong IVs; for example, can college proximity influence a student's decision to go to college regardless of skill? If yes, college proximity is a strong IV. We compare against other identification conditions (like 2SLS and [143]) in appendix D.1.8.

5.3 The General Control Function Method (GCFN)

GCFN constructs a general control function and estimates effects with it. GCFN has two stages. The first stage constructs a general control function as the code of an autoencoder. The second stage builds a model from the control function and the treatment to the outcome and estimates effects.

Variational Decoupling We construct the control function $\hat{\mathbf{z}}$ as a stochastic function of the treatment $\mathbf{t}$ and the IV $\boldsymbol{\epsilon}$; with parameter θ, the estimator is $q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})$. First, to guarantee the reconstruction property (A1 in theorem 5), the control function and the IV must determine treatment, implying that with parameter ϕ, $p_\phi(\mathbf{t} \mid \hat{\mathbf{z}} = \hat{z}, \boldsymbol{\epsilon})$ should be maximized for $\hat{z} \sim q(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})$. Together, these form the parts of an autoencoder where a control function is sampled conditioned on the treatment and IV, while the treatment is reconstructed from the same control function and IV. Second, to guarantee marginal independence, we force the control function to be independent of the IV: $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$. Let the true data distribution be $F(\mathbf{t}, \boldsymbol{\epsilon})$ and $\mathbf{I}$ denote mutual information. Putting the two parts together, we define a constrained optimization to construct $\hat{\mathbf{z}}$, called variational decoupling (VDE):

$$\text{(VDE)} \quad \max_{\theta,\phi} \mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon})} \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon})} \log p_\phi(\mathbf{t} \mid \hat{\mathbf{z}}, \boldsymbol{\epsilon}) \quad s.t \quad \mathbf{I}_\theta(\hat{\mathbf{z}}; \boldsymbol{\epsilon}) = 0. \tag{5.4}$$

Recall from section 5.2.1 that with a reconstruction map $d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$ (from A1 in theorem 5) that reflects the functional structure of the treatment, marginal independence $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$ implies joint independence. To model such a map, VDE's decoder, $p_\phi(\mathbf{t} \mid \hat{\mathbf{z}} = \hat{z}, \boldsymbol{\epsilon})$ reflects the same functional structure. For example, with an additive treatment process the decoder would be parametrized as $\log p_\phi(\mathbf{t} \mid \hat{\mathbf{z}}, \boldsymbol{\epsilon}) \propto -(\mathbf{t} - h'_\phi(\hat{\mathbf{z}}) - g'_\phi(\boldsymbol{\epsilon}))/\sigma^2_\phi$; σ_ϕ allows for a point-mass distribution p_ϕ at optimum of VDE. In summary, beyond the observed treatment and IV, VDE takes a specification of the functional structure of the treatment process as input which informs the structure of the decoder.

VDE is converted to an unconstrained optimization problem by absorbing the independence constraint into the optimization via the Lagrange multipliers trick with $\lambda > 0$,

$$\max_{\theta,\phi} \mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon})} \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon})} \log p_\phi(\mathbf{t} \mid \hat{\mathbf{z}}, \boldsymbol{\epsilon}) - \lambda \mathbf{I}_\theta(\hat{\mathbf{z}}; \boldsymbol{\epsilon}). \tag{5.5}$$

Estimation of the mutual information requires $q_\theta(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon})$. Instead, we lower bound the negative mutual information by introducing an auxiliary distribution $r_\nu(\hat{\mathbf{z}})$. This yields a tractable objective:

$$\max_{\theta,\phi,\nu} \mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon})}\left[(1+\lambda)\mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon})} \log p_\phi(\mathbf{t} \mid \hat{\mathbf{z}}, \boldsymbol{\epsilon}) - \lambda \mathrm{KL}\left[q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}) \parallel r_\nu(\hat{\mathbf{z}})\right]\right]. \tag{5.6}$$

A full derivation can be found in Appendix D.1.2. The lower bound is tight when the auxiliary distribution $r_\nu(\hat{\mathbf{z}}) = q_\theta(\hat{\mathbf{z}})$. For example, when $q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})$ is categorical, optimizing eq. (5.6) with a categorical $r_\nu(\hat{\mathbf{z}})$ makes the lower bound tight. The parameters θ, ϕ, ν can be learned via stochastic optimization. VDE can be adapted to use covariates by conditioning on the covariates as needed.

Outcome Modeling. VDE provides a general control function $\hat{\mathbf{z}}$ and its marginal distribution $q_\theta(\hat{\mathbf{z}})$. If the IV is strong, $\hat{\mathbf{z}}$ satisfies ignorability and positivity and the causal effect can be estimated by regressing the outcome on the control function and the treatment. Other effect estimation methods like matching/balancing methods [127, 128, 129] and doubly-robust methods [130] can be used. This regression is GCFN's second stage, called the outcome stage. We formalize this outcome stage as a maximum-likelihood problem and learn a model with parameters β under the true data distribution $F(\mathbf{y}, \mathbf{t}, \boldsymbol{\epsilon})$ and the general control function distribution $q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})$:

$$\arg\max_{\beta} \mathbb{E}_{F(\mathbf{y},\mathbf{t},\boldsymbol{\epsilon})}\mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon})} \log p_\beta\left(\mathbf{y} \mid \hat{\mathbf{z}}, \mathbf{t}\right). \tag{5.7}$$

Semi-Supervised GCFN. The explicit optimization to learn the control function in VDE makes it simple to take advantage of datapoints where both the confounder and IV are observed by forcing the control function to predict the observed confounder. Let $\mathbf{m}$ be an missingness indicator variable that is 1 when the true confounder $\mathbf{z}$ is observed and 0 otherwise. Let the joint distribution be $F(\mathbf{t}, \boldsymbol{\epsilon}, \mathbf{m}, \mathbf{z})$ and ζ be a scaling hyperparameter parameter. Then the augmented VDE stage in semi-supervised GCFN, with $\kappa = \lambda/(1+\lambda)$, is

$$\max_{\theta,\phi,\nu} \mathop{\mathbb{E}}_{F(\mathbf{t},\boldsymbol{\epsilon},\mathbf{m},\mathbf{z})} \left[\mathop{\mathbb{E}}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon})} \log p_\phi(\mathbf{t} \mid \hat{\mathbf{z}}, \boldsymbol{\epsilon}) - \kappa \mathrm{KL}\left[q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}) \parallel r_\nu(\hat{\mathbf{z}})\right] + \zeta \mathbf{m} \log q_\theta(\hat{\mathbf{z}} = \mathbf{z} \mid \mathbf{t}, \boldsymbol{\epsilon}). \right] \quad (5.8)$$

The added term $\log q_\theta(\hat{\mathbf{z}} = \mathbf{z} \mid \mathbf{t}, \boldsymbol{\epsilon})$ encourages the control function to place all of its mass on the observed confounder value. When the control function places all of its mass on the confounder, the control function is determined by value of the confounder. Together with the fact that the confounder is independent of the IV, this implies the control function, confounder pair is jointly independent of the instrument. Therefore, given enough datapoints with the confounder and IV observed, joint independence can be guaranteed without treatment assumptions like in section 5.2.1. The second stage of semi-supervised GCFN uses the outcome regression in eq. (5.7) to estimate effects.

5.3.1 Error bounds for GCFN's estimated effects

An imperfectly estimated general control function may violate the conditional independence $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}$ which is required for ignorability. If ignorability does not hold, estimated effects are biased. First, assuming an additive treatment process, we bound the expected bias in causal effects using quantities optimized during training in VDE, specifically reconstruction error and dependence of $\hat{\mathbf{z}}$ on $\boldsymbol{\epsilon}$:

Theorem 6. *Assume an additive treatment process $\mathbf{t} = \mathbf{z} + g(\boldsymbol{\epsilon})$ where g is an L_g-Lipschitz function, and $\mathbb{E}_{F(\mathbf{z})}\mathbf{z} = 0$. Let $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = t, \mathbf{z} = z] = f(t, z)$ be an L-Lipschitz function in z for any t. Further,*

1. *let reconstruction error be non-zero but bounded $\mathbb{E}_{q(\mathbf{t},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{t} - \hat{\mathbf{z}} - g'(\boldsymbol{\epsilon}))^2 \leq \delta$. Assume that g' is also L_g-Lipschitz. Further, let $\mathbb{E}_{q(\hat{\mathbf{z}})}\hat{\mathbf{z}} = 0$, and $\mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}| < \infty$.*
2. *Assume $\boldsymbol{\epsilon} \not\perp\!\!\!\perp \hat{\mathbf{z}}$ and let the dependence be bounded:* $\max_{\hat{z}} \mathcal{W}_1\left(q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}} = \hat{z}) \parallel F(\boldsymbol{\epsilon})\right) \leq \gamma$.

With the estimated and true causal effects as $\hat{\tau}(t) = \mathbb{E}_{\hat{\mathbf{z}}} f(t, \hat{\mathbf{z}})$ and $\tau(t) = E_{\mathbf{z}} f(t, \mathbf{z})$ respectively,

$$\mathbb{E}_{F(\mathbf{t})} |\hat{\tau}(\mathbf{t}) - \tau(\mathbf{t})| \leq L \sqrt{\delta + 4\gamma L_g \mathbb{E}_{q(\hat{\mathbf{z}})} |\hat{\mathbf{z}}|}.$$

See appendix D.1.9.1 for the proof. Second, in theorem 3 in appendix D.1.9.2, we prove a general error bound for GCFN that depends on the residual confounding that $\hat{\mathbf{z}}$ does not control for, measured as the conditional mutual information $\mathbf{I}(\mathbf{z}; \mathbf{t} \mid \hat{\mathbf{z}})$. When $\mathbf{I}(\mathbf{z}; \mathbf{t} \mid \hat{\mathbf{z}}) > 0$, ignorability may not hold and estimated effects are biased. Assuming positivity and a sufficiently concentrated $\mathbf{z} \mid \hat{\mathbf{z}}$, we prove in theorem 3 that $\mathbf{I}(\mathbf{z}; \mathbf{t} \mid \hat{\mathbf{z}})$ controls average absolute error in effects. This error is tempered by the smoothness of outcome as a function of the confounder $\mathbf{z}$. This bound also accounts for errors due to poor estimation of $\mathbb{E}[\mathbf{y} \mid \mathbf{t}, \hat{\mathbf{z}}]$ in low density regions of $q(\mathbf{t}, \hat{\mathbf{z}})$ which may occur when $\hat{\mathbf{z}} \not\perp\!\!\!\perp \boldsymbol{\epsilon}$.

5.4 Experiments

We evaluate GCFN on simulated data, where the true causal effects are known and show that GCFN corrects for confounding and estimates causal effects better than CFN, 2SLS, and a 2SLS variant, DeepIV [131]. We then evaluate GCFN on high-dimensional data using simulations from DeepIV [131] and DeepGMM [132]. Then, we estimate the effect of slave export on community trust [2] and compare GCFN's estimate to the effect reported in [2].

Experimental details For GCFN, we let the control function $\hat{\mathbf{z}}$ be a categorical variable. The encoder in VDE, f_θ, is a 2-hidden-layer neural network f_θ, which parametrizes a categorical likelihood $q_\theta(\hat{\mathbf{z}} = i \mid \mathbf{t} = t, \boldsymbol{\epsilon} = \epsilon) \propto \exp\left(f_\theta(t, \epsilon, i)\right)$. The decoder is also a 2-hidden-layer network; the reconstructed likelihood of $\mathbf{t}$ is different for different experiments. In all experiments, the hidden layers in both encoder and decoder networks have 100 units and use ReLU activations. The outcome model is also a 2-hidden-layer neural network with ReLU activations. For the simulated data, the hidden layers in the outcome model have 50 hidden units. In estimating the effect of slave export, the hidden layers in the outcome model have only 10 hidden units; larger width resulted in overfitting. Unless specified otherwise, we train on 5000 samples with a batch size of 500 for optimizing both VDE and the outcome model for 100 epochs with Adam [90]. In section 5.4.1 and section 5.4.2, we evaluate effect estimates on a subset of the support of the treatment distribution where the most mass lies: 200 equally spaced treatment values in $[-1, 1]$. We defer other details to appendix D.2.

All hyperparameters for VDE, except the mutual-information coefficient $\kappa = \lambda/(1+\lambda)$, and the outcome-stage were found by evaluating the respective objectives on a held-out validation set. In our experiments, we found that setting κ between $0.1 - 0.4$ worked best. GCFN's performance was only mildly sensitive to changing κ within this range. However, one can tune κ further by choosing the one which gives the control function $\hat{\mathbf{z}}_\kappa$ that results in the largest expected outcome likelihood on a heldout set. This procedure relies on VDE and outcome objectives reaching optimum if and only if $\hat{\mathbf{z}}$ satisfies perfect reconstruction and marginal independence. See appendix D.2.1 for further details.

5.4.1 Simulations with specific decoder structure

We compare GCFN's performance against 2SLS, CFN and DeepIV and show that GCFN outperforms these methods when the functional properties of the treatment process are known. We consider two settings with continuous outcome, treatment, and confounders where the assumptions of 2SLS and CFN fail: 1) with an additive treatment process and a multiplicative outcome process and 2) with a multiplicative treatment process and an additive outcome process. For both settings, the causal effect is the same $\mathbb{E}[\mathbf{y} \mid \mathrm{do}(\mathbf{t} = t)] = t$. The control function $\hat{\mathbf{z}}$ is set to have 50 categories. We report results for the mutual information coefficient $\kappa = {}^{\lambda}/_{1+\lambda} = 0.1$. We consider 3 different strengths of confounding as captured by the parameter $\alpha \in [0.5, 1.0, 2.0]$.

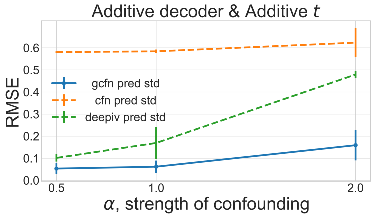


Figure 5.2: GCFN obtains better effect estimates than CFN and DeepIV when the *additive outcome process* assumption is violated.

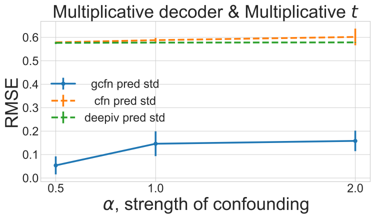


Figure 5.3: GCFN produces better effect estimates than CFN and DeepIV when the *additive treatment* process assumption is violated.

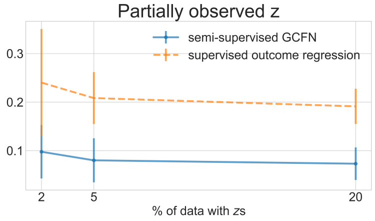


Figure 5.4: Mean RMSE of causal effects of the GCFN-predicted causal effects versus percentages of samples with z observed.

Multiplicative outcome & Additive treatment With $\mathcal{N}$ as the normal distribution, we generate $\mathbf{z}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $\mathbf{t} = (\mathbf{z} + \boldsymbol{\epsilon})/\sqrt{2}$, $\mathbf{y} \sim \mathcal{N}(\mathbf{t} + \alpha\mathbf{t}^2\mathbf{z}, 0.1)$, wher α controls confounding; larger magnitude of α means more confounding. The generation process above violates the linear noise relation assumption, $\mathbb{E}[\boldsymbol{\eta}_\mathbf{y} | \boldsymbol{\eta}_\mathbf{t}] \propto \boldsymbol{\eta}_\mathbf{t}$, that CFN requires [134]. GCFN, on the other hand, does not require this assumption. In this experiment, VDE has an additive decoder which specifies a Gaussian reconstruction likelihood: $\mathbf{t} \sim \mathcal{N}(h'_\phi(\hat{\mathbf{z}}) + g'_\phi(\boldsymbol{\epsilon}), 1)$. In Figure 5.2, we compare GCFN to CFN and DeepIV, and show that GCFN produces the best causal effect estimates. Unlike the others, GCFN can adjust for confounding when the outcome process is not additive. Averaged over all αs, GCFN outperforms the baselines with an RMSE of $\mathbf{0.09 \pm 0.06}$ compared to CFN's $\mathbf{0.58 \pm 0.01}$, 2SLS's $\mathbf{0.55 \pm 0.58}$, and DeepIV's $\mathbf{0.25 \pm 0.17}$.

Multiplicative treatment & Additive outcome. For this simulation, we generate data as follows: $\mathbf{z}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $\mathbf{t} = \mathbf{z}\boldsymbol{\epsilon}$, $\mathbf{y} \sim \mathcal{N}(\mathbf{t} + \alpha\mathbf{z}, 0.1)$. In this experiment, VDE has a *multiplicative* decoder which specifies a gaussian reconstruction likelihood with $\mathbf{t} = \mathcal{N}(h'_\phi(\hat{\mathbf{z}})g'_\phi(\boldsymbol{\epsilon}), 1)$. The 2SLS method uses a linear model $\mathbf{t} = \beta\boldsymbol{\epsilon} + \boldsymbol{\eta}_\mathbf{t}$ which will correctly estimate $\mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = 0$ in our generation process. Figure 5.3 shows that GCFN out-performs CFN and DeepIV and is robust to different strengths of confounding ($\alpha \in \{0.5, 1, 2\}$). Averaged over all αs, GCFN outperforms the baselines with an RMSE of $\mathbf{0.13 \pm 0.08}$ compared to CFN's $\mathbf{0.58 \pm 0.02}$, 2SLS's $\mathbf{0.55 \pm 0.56}$, and DeepIV's $\mathbf{0.58 \pm 0.01}$. We omit 2SLS from fig. 5.2 because it performs strictly worse than

DeepIV, its deep variant. DeepIV gives effect-estimates that are close to 0. We justify this in appendix D.1.11.

5.4.2 GCFN with confounders observed on a subset

In this experiment, we demonstrate that semi-supervised GCFN does not need outcome or treatment process assumptions if the confounder $\mathbf{z}$ is observed on a subset of the data. Let ρ be the fraction with $\mathbf{z}$ observed and $\mathcal{B}$ be the Bernoulli distribution. We generate a mask $\mathbf{m} \sim \mathcal{B}(\rho)$ and data $\boldsymbol{\epsilon}, \mathbf{z} \sim \mathcal{N}(0,1), \quad \mathbf{t} = \boldsymbol{\epsilon}\mathbf{z}, \quad \mathbf{y} \sim \mathcal{N}(\mathbf{t} + \mathbf{t}\mathbf{z}, 0.1)$. Let $\mathbf{z}' = \mathbf{z} * \mathbf{m}$. We observe $(\mathbf{y}, \mathbf{t}, \boldsymbol{\epsilon}, \mathbf{z}', \mathbf{m})$. The structurally unrestricted decoder uses a categorical reconstruction likelihood: $p_\phi(\mathbf{t} = j \mid \hat{\mathbf{z}} = z, \boldsymbol{\epsilon} = \epsilon) \propto \exp\left(g_\phi(z, \epsilon, j)\right)$. The treatment $\mathbf{t}$ is discretized into 50 bins. The intervals $[-\infty, -3.5]$ and $[3.5, \infty]$ correspond to one bin each and the interval $[-3.5, 3.5]$ is split into 48 equally-sized bins. This suffices because few samples fall outside $[-3.5, 3.5]$. For semi-supervised GCFN, VDE's objective has an additional term defined on the samples with observed $\mathbf{z}$'s (eq. (5.8)). The confounder $\mathbf{z}$ is split into bins the same way as the treatment. The additional term for the i^{th} sample is the categorical log-likelihood of the observed (t_i, ϵ_i, z_i) with respect to the encoder-specified distribution: $q(\hat{\mathbf{z}} = z_i \mid \mathbf{t} = t_i, \boldsymbol{\epsilon} = \epsilon_i) \propto \exp(f_\theta(t_i, \epsilon_i, z_i))$. We set the scaling ζ on this additional term to be 0.5. We report results for $\kappa = 0.1$. For other $\kappa \in \{0.2, 0.3\}$, results were similar or better.

We compare semi-supervised GCFN against regression with the same outcome model as the baseline, trained only on samples with the confounder observed. We estimated this "supervised" baseline in the same manner as the outcome stage of GCFN. Figure 5.4 plots the RMSE of the predicted causal effects vs. percentage of samples with observed $\mathbf{z}$'s in fig. 5.4. If the data has 2% or more samples with the confounder observed, GCFN estimates effects better than the supervised baseline.

5.4.3 GCFN on high-dimensional Covariates

In this experiment, we evaluate GCFN on a non-linear simulation given in Hartford et al. [131] to demonstrate that DeepIV improves upon 2SLS. Their generation models the effect of price ($\mathbf{t}$) on sales ($\mathbf{y}$), given customer covariates ($\mathbf{x}$, MNIST image), and time s; they use fuel price as an IV. The outcome is generated using the label of the MNIST image, which denotes customer price sensitivity. The data generation process for $\mathbf{t}$ is additive in IV and confounder. Following this, we use the same additive decoder in VDE as in section 5.4.1, but with time s as an additional input. We give further experimental details and Hartford et al. [131]'s data generating process in appendix D.2.3.

We report effect MSE on a fixed out-of-sample set (oos). We compare against Hartford et al. [131]'s reported results for two sample sizes, $10,000$ and $20,000$. DeepIV's reported results exclude a few large effect MSE outliers; we do not exclude such errors for GCFN. We report GCFN's performance over 10 seeds. Overall, GCFN performed on par or better than DeepIV. First, we report GCFN's effect MSE with $\kappa = 0.2$. For $10,000$ samples, GCFN produced effect MSEs that ranged in $[\mathbf{0.30 - 0.42}]$, better than DeepIV's reported range of around $[\mathbf{0.30 - 0.50}]$ (which is almost twice as large). For $20,000$ samples, GCFN's effect MSE range improved to $[\mathbf{0.25 - 0.40}]$ while DeepIV reported a performance of around $[\mathbf{0.25 - 0.45}]$. For both sample sizes, we note that $\kappa = 0.1, 0.3$ gave similar results. To see this, for $20,000$ samples, averaged over 10 seeds, GCFN achieved a mean effect MSE of $\mathbf{0.305}$ or better for any $\kappa \in \{0.1, 0.2, 0.3\}$, beating DeepIV's $\mathbf{0.32}$.

5.4.4 GCFN on high-dimensional IVs

In this experiment, we evaluate GCFN on data with a high-dimensional IV. Bennett et al. [132] use the following data generating process to demonstrate DeepGMM [132] improves upon existing methods: $\boldsymbol{\epsilon} \sim \mathcal{U}[-3, 3] \quad \mathbf{z} \sim \mathcal{N}(0, 1) \quad \mathbf{t} \sim \mathcal{N}(\mathbf{z} + \boldsymbol{\epsilon}, 0.1) \quad \mathbf{y} = \mathcal{N}\left(|\mathbf{t}| + \mathbf{z}, 0.1\right)$. However, the scalar $\boldsymbol{\epsilon}$ is not directly observed. Instead, $\boldsymbol{\epsilon}$ is mapped to a digit $\{0, \ldots, 9\}$ and a corresponding MNIST image $\boldsymbol{\epsilon}_M$ is given as the IV. To estimate effects well with such an IV, any method must learn to label the MNIST image. In this setting, VDE's encoder and decoder both take an embedding $\ell_\gamma(\boldsymbol{\epsilon}_M) \in \mathbb{R}^{10}$ as input. The embedding ℓ_γ is trained in VDE along with the encoder and decoder. Respecting the additive treatment process, we specify an additive decoder.

We ran GCFN with 10 different random seeds and report results for $\kappa = 0.3$, chosen based on mean test outcome MSE (0.136 ± 0.008). GCFN performs competitively with an effect MSE of $\mathbf{0.077 \pm 0.022}$ compared to DeepGMM's $\mathbf{0.07 \pm 0.02}$ and DeepIV's $\mathbf{0.11 \pm 0.00}$, both as reported in [132]. Effect MSE for $\kappa \in \{0.2, 0.4\}$ were similar and within standard error of DeepGMM's performance. See appendix D.2.4 for further experimental details and results.

5.4.5 The Effect of Slave Export on Trust

We demonstrate the recovery of the causal effect of slave export on the trust in the community [2]. Nunn and Wantchekon [2] pooled surveys and historical records to get sub-ethnicity and tribe level data from the period of slave trade. The data was used to study the long-term effects of slave-trade, measured in the 2005 Afrobarometer survey. We predict the effect of the treatment **t** =**ln(1 + slave-export/area)** on the outcome of interest, **y** =**trust in neighbors**. The dataset has 6932 samples with 59 features. After filtering out missing values, we preprocessed 46 covariates and IV to have mean 0 and maximum 1, and the treatment **t** to lie in $[0, 2]$. The authors claim that the distance to sea cannot causally affect how individuals trust each other, but it affects the chance of coming in contact with colonial slave-traders and being shipped to the Americas, making it an IV. They control for urbanization, fixed effects for sophistication, political hierarchies beyond community, integration with the rail network, contact with European explorers, and missions during colonial rule.

For this experiment, VDE's decoder g_ϕ specifies a categorical reconstruction likelihood as $p_\phi(\mathbf{t} =$

$i \mid \hat{\mathbf{z}} = z, \boldsymbol{\epsilon} = \epsilon) \propto \exp\left(g_\phi(z, \epsilon, i)\right)$. Each category of the treatment corresponds to one of 50 equally-sized bins in the interval $[0, 2]$. Nunn and Wantchekon [2] use a linear model for the outcome $\mathbf{y}$ and use the distance to sea as an IV for each community. We also use a partially linear model $\mathbf{y} = \beta\mathbf{t} + h_\theta(\hat{\mathbf{z}})$ so that the effect we recover is of comparable nature to the effect reported in the paper. The outcome network h_θ has 2 layers with 10 hidden units each and ReLUs.

Averaged over 4 mutual information coefficients κ and 5 random seeds, GCFN's estimate of β was $\mathbf{-0.21 \pm 0.04}$ compared with $\mathbf{-0.27 \pm 0.10}$, as reported by Nunn and Wantchekon [2].

5.5 Discussion and Future

In this paper, we characterize general control functions for causal estimation. General control functions allow for effect estimation without structural outcome process assumptions like 2SLS or CFN. The key challenge in building general control functions is ensuring joint independence between the IV and the control function and (unobserved) true confounder. Joint independence can be guaranteed via structural treatment process assumptions, like additivity or monotonicity. We develop the general control function method (GCFN) to build general control functions and estimate effects with them. Further, we develop semi-supervised GCFN which uses confounders observed on a subset of the data to construct general control functions without treatment process assumptions. Finally, we consider imperfect estimation of the general control function and bound average error in effects using quantities optimized in VDE.

Tradeoffs with assumptions. In causal estimation, parametric assumptions can be traded-off with assumptions of strength of IV or positivity. Consider a setting where $\boldsymbol{\epsilon}$ is binary. For every possible confounder value, only two values of the treatment are observed. Thus it is impossible to estimate a quadratic function of $\mathbf{t}$ for each fixed value of the confounder. This means $\mathbf{y} \mid \mathbf{t}$ is not identified without strong assumptions like linearity in $\mathbf{t}$. Incorporating outcome properties, like the conditional independence $\mathbf{y} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \mathbf{t}, \mathbf{z}$, into control function estimation would be a fruitful direction.

6 | Causal Estimation with Functional Confounders

6.1 Introduction

Determining the effect of interventions on outcomes using observational data lies at the core of many fields like medicine, economic policy, and genomics. For example, policy makers estimate effects to elect whether to invest in education or job training programs. In medicine, doctors use effects to design optimal treatment strategies for patients. Geneticists perform genome-wide association studies (GWAS) to relate genotypes and phenotypes. In observational data, there could exist unobserved variables that affect both the intervention and the outcome, called confounders. A necessary condition for the causal effect to be identified is that all confounders are observed; called *ignorability*. If ignorability holds, a sufficient condition for causal effect estimation is adequate variation in the intervention after conditioning on the confounders; this condition is called *positivity*.

The data apriori does not differentiate between confounders and interventions. It is the practitioners that select interventions of interest from all pre-outcome variables (variables that occur before the outcome). Then, assuming knowledge of the data generating mechanism, practitioners can label certain variables amongst the remaining pre-outcome variables as confounders. This corresponds to indexing into the set of pre-outcome variables.

In certain problems the confounders are specified as a function of the pre-outcome variables that does not simply index into the set of pre-outcome variables. For a concrete example, consider GWAS. The goal in GWAS is to estimate the influence of genetic variations on phenotypes like disease risk. In GWAS, population and family structures both result in certain genetic variations and affect phenotypes and therefore, are confounders [145]. Practitioners specify these confounders by using the genetic similarity between individuals [146, 147, 148], which is a function of the genetic variations. When the confounders are a function of the same pre-outcome variables that define the interventions, positivity is violated. Then, the class of interventions whose effects are estimable is not well-defined.

We study causal effect estimation in such settings, where a function of the pre-outcome variables provides the confounder and these same pre-outcome variables define the intervention. We call this estimation with functional confounders (EFC). In EFC, one column in the observed data is the outcome and all others are pre-outcome variables. We assume access to a function $h(\cdot)$ that takes as input the pre-outcome variables and returns the value of the confounder. Further, we assume these confounders give us ignorability. In settings like GWAS, the function h reflects the practitioner-specified function that captures the genetic variation influenced by the population structure. In traditional OBS-CI, $h(\cdot)$ reflects the selection of certain variables in the data and labelling them as confounders. In EFC, two different values of the confounder are never observed for the same setting of the pre-outcome variables. This means that positivity is violated and the effects of only certain interventions may be estimable.

We address this issue in two ways. First, we investigate a class of plausible interventions that are *functions* of the observed pre-outcome variables, called functional interventions. We develop a sufficient condition to estimate the effects of said functional interventions, called functional positivity (F-POSITIVITY). Second, we consider intervening on all pre-outcome variables, called the *full* intervention. We develop a sufficient condition to estimate the effect of the *full* intervention, called causal redundancy (c-redundancy). For an intervention, given a confounder value, c-redundancy allows us to compute a surrogate intervention such that the conditional effect of the surrogate is equal to that of the original intervention. We also show that such surrogate interventions exist only under a certain condition that we call Effect Connectivity, that is necessary for nonparametric effect estimation in efc. This condition is satisfied by default in traditional obs-ci if ignorability and positivity hold. Then, we develop an algorithm for causal estimation assuming c-redundancy, called Level-set Orthogonal Descent Estimation (lode), which estimates effects using surrogate interventions. If the surrogate is not estimated well, lode's estimates are biased. We establish bounds on this bias that capture the mitigating effect of the smoothness of the true outcome function.

Related work The problem of genome-wide association studies (gwas) is to estimate the effect of genetic variations(also called snps) on the phenotype [149]. The ancestry of the subjects acts as a confounder in gwas. In gwas practice, principle component analysis (pca) and linear mixed models (lmms) are used to compute this confounding structure [147, 148]. Lippert et al. [146] suggest estimating the confounders and effects on *separate* subsets of the snps. This separation disregards the confounding that is captured in the interaction of the two subsets of snps. Gwas is a special case of effects from multiple treatments (mte) where the confounder value is specified via optimization as a function of the pre-outcome variables [150, 151]. In all these settings, positivity is violated and not all effects are estimable. We provide an avenue for nonparametric effect-estimation of the full intervention under a new condition, c-redundancy.

Traditional observational causal inference (obs-ci) review We setup causal inference with Structural Causal Models [144] and use $do(\mathbf{t} = \mathtt{t}^*)$ to denote making an intervention. Let $\mathbf{t}$ be a vector of the interventions, $\mathbf{z}$ be the confounder, and $\mathbf{y}$ be the outcome. Let $\boldsymbol{\eta} \sim p(\boldsymbol{\eta})(\boldsymbol{\eta} \perp\!\!\!\perp (\mathbf{z}, \mathbf{t}))$ be noise. With f as the *outcome function*, we define the causal model for traditional obs-ci as

[*]:

$$\mathbf{z} \sim p(\mathbf{z}), \quad \mathbf{t} \sim p(\mathbf{t} \mid \mathbf{z}), \quad y = f(\mathbf{t}, \mathbf{z}, \boldsymbol{\eta}).$$

Let $p(y, \mathbf{z}, \mathbf{t})$ denote the joint distribution implied by this data generating process. The effects of interest under the full intervention $do(\mathbf{t} = \mathtt{t}^*)$ are the average and *conditional effect*

$$\text{(average)} \quad \tau(\mathtt{t}^*) = \mathbb{E}_{\mathbf{z},\boldsymbol{\eta}} f(\mathtt{t}^*, \mathtt{z}, \boldsymbol{\eta}) \qquad\qquad \text{(conditional)} \quad \phi(\mathtt{t}^*, \mathtt{z}) = \mathbb{E}_{\boldsymbol{\eta}}\left[f(\mathtt{t}^*, \mathtt{z}, \boldsymbol{\eta})\right]. \tag{6.1}$$

With observed confounders, two assumptions make causal estimation possible: *ignorability* and *positivity*. Ignorability means that *all* confounders $\mathbf{z}$ are observed in data. Conditioning on all the confounders, the outcome under an intervention is distributed as if conditional on the value of the intervention: $p(\mathrm{y} = y_1 \mid do(\mathbf{t} = \mathtt{t}^*), \mathbf{z} = \mathtt{z}) = p(f(\mathtt{t}^*, \mathtt{z}, \boldsymbol{\eta}) = y_1) = p(\mathrm{y} = y_1 \mid \mathbf{t} = \mathtt{t}^*, \mathbf{z} = \mathtt{z})$. This allows the expression of average effect as an expectation over the *observed* outcomes $\tau(\mathtt{t}^*) = \mathbb{E}_{\mathbf{z},\boldsymbol{\eta}}[f(\mathtt{t}^*, \mathbf{z}, \boldsymbol{\eta})] = \mathbb{E}_{\mathbf{z}}\mathbb{E}[y \mid \mathbf{z}, \mathtt{t}^*]$. The conditional expectation only exists for all $\mathtt{t}^*$ if $p(y \mid \mathbf{z}, \mathbf{t} = \mathtt{t}^*) = {}^{p(y,\mathbf{z},\mathbf{t}=\mathtt{t}^*)}/_{p(\mathbf{z})p(\mathbf{t}=\mathtt{t}^* \mid \mathbf{z})}$ exists. *Positivity* guarantees this existence

$$\text{(positivity)} \quad \forall \mathtt{t}^* \in \operatorname{supp}(\mathbf{t}) \quad p(\mathbf{z} = \mathtt{z}) > 0 \implies p(\mathbf{t} = \mathtt{t}^* \mid \mathbf{z} = \mathtt{z}) > 0. \tag{6.2}$$

6.2 Estimation with functional confounders

In traditional obs-ci, causal estimation relied on knowing the confounders. In this section, we consider settings where confounders are known via a function of the pre-outcome variables $h(\mathbf{t}) = \mathbf{z}$. We call this setting *estimation with functional confounders (efc)*. An example of this is gwas, where snps (the pre-outcome variables) are used to estimate the confounding population structure through methods like pca [148]. Assuming the confounders are a function of

[*] We focus on f that generates $\mathbf{y}$ from $\mathbf{t}, \mathbf{z}$. SCMs generally specify the function that generates $\mathbf{t}$ from $\mathbf{z}$ also.

the pre-outcome variables violates positivity in general. Positivity is violated in this setting because

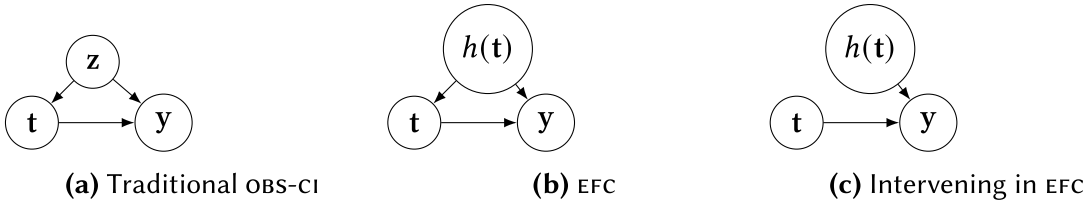
$$\forall t_1, t_2 \in \operatorname{supp}(\mathbf{t}) \;\; s.t. \;\; h(t_2) \neq h(t_1) \quad \Longrightarrow \quad p(\mathbf{z} = h(t_2) \mid \mathbf{t} = t_1) = 0 \neq p(\mathbf{z} = h(t_2)) > 0$$

In words, two different confounder values cannot occur for the same t. A positivity violation precludes nonparametric effect estimation of the full intervention $do(\mathbf{t} = \mathrm{t}^*)$.

Positivity and Regression Identifiability   Positivity can be viewed as providing identifiability. To see this, let the confounder be $\mathbf{z} = h(\mathbf{t})$ and the outcome be $y(\mathbf{t}, \mathbf{z}, \boldsymbol{\eta}) = \mathbf{z} + h(\mathbf{t})$. Now consider regressing $\mathbf{z}$ and $\mathbf{t}$ onto y. Then, functions $y = \alpha \mathbf{z} + \beta h(\mathbf{t})$ indexed by α, β, such that $\alpha + \beta = 2$, are consistent with the observed data. Thus, there exist infinitely many solutions to the conditional expectation of y on $(\mathbf{t}, \mathbf{z})$, meaning that the regression is not identifiable. Assuming positivity necessitates sufficient randomness to identify the regression and thus the causal effect. A violation of positivity means that nonparametric estimation of causal effects needs further assumptions.

6.2.1 Setup for estimation with functional confounders

In efc, the confounder is provided as a non-bijective function h of the pre-outcome variables $\mathbf{t}$. To reflect this property, we use $h(\mathbf{t})$ to denote the confounder. As an illustrative example, let $\mathcal{G}$ be the Gamma distribution and consider $\mathbf{z} \in \{-1, 1\}, p(\mathbf{z} = 1) = 0.5$ is the confounder and the intervention of interest is $\mathbf{t} = \mathbf{z} * \mathcal{G}(1, \exp(\mathbf{z}))$. Note $\operatorname{sign}(\mathbf{t}) = \mathbf{z}$ meaning that $h(\mathbf{t}) = \operatorname{sign}(\mathbf{t})$ is the confounder. Figure 6.1 shows causal graphs connecting our efc notation to that in traditional obs-ci. With noise $\boldsymbol{\eta} \sim p(\boldsymbol{\eta})(\boldsymbol{\eta} \perp\!\!\!\perp \mathbf{t})$, our causal model samples, in order, the confounder "part" of pre-outcome variables $h(\mathbf{t})$, the pre-outcome variables $\mathbf{t}$, and the outcome $\mathbf{y}$ via the *outcome*



(a) Traditional OBS-CI **(b)** EFC **(c)** Intervening in EFC

Figure 6.1: Causal Graphs for Traditional OBS-CI vs. EFC.

function f †:

$$h(\mathbf{t}) \sim p(h(\mathbf{t})) \quad \mathbf{t} \sim p(\mathbf{t} \mid h(\mathbf{t})) \quad \mathbf{y} = f(\mathbf{t}, h(\mathbf{t}), \boldsymbol{\eta})$$

Similar to traditional OBS-CI, for an intervention $\mathtt{t}^*$ the average effect, $\tau(\cdot)$, and the conditional effect, $\phi(\cdot,\cdot)$ at $h(\mathtt{t}_2^*)$, respectively, are defined as:

$$\tau(\mathtt{t}^*) = \mathop{\mathbb{E}}_{h(\mathbf{t}),\boldsymbol{\eta}}[f(\mathtt{t}^*, h(\mathbf{t}), \boldsymbol{\eta})], \qquad \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \mathop{\mathbb{E}}_{\boldsymbol{\eta}}[f(\mathtt{t}^*, h(\mathtt{t}_2^*), \boldsymbol{\eta})]. \tag{6.3}$$

As the pre-outcome variables determine the confounder, positivity is violated. Further, the *outcome function* $f(\mathbf{t}, h(\mathbf{t}), \boldsymbol{\eta})$ could recover the exact value of $h(\mathbf{t})$ from $\mathbf{t}$ instead of its second argument. Thus, two different outcome functions could lead to the same observational data distribution, posing a fundamental obstacle to causal effect estimation. This is the central challenge in EFC.

6.2.2 Causal Questions With Functional Positivity

Without positivity, we can only estimate the effects of certain functions of $\mathbf{t}$. We call such interventions, on some function $g(\mathbf{t})$, *functional interventions.* The implied causal model for the outcome for functional intervention value $g(\mathtt{t}^*)$ and confounder value $h(\mathtt{t}_2^*)$ is first $\mathbf{t} \sim p(\mathbf{t} \mid g(\mathbf{t}) =$

† We also assume no interference [152] (also called Stable Unit Treatment Value Assumption [153]) which means that an individual's outcome does not depend on others' treatment. In EFC, when $\mathbf{t}$ and $\boldsymbol{\eta}$ are sampled IID there is no interference. To see this, note $\forall i, j \ (\mathbf{t}_i, \boldsymbol{\eta}_i) \perp\!\!\!\perp (\mathbf{t}_j, \boldsymbol{\eta}_j) \implies (\mathbf{y}_i, \mathbf{t}_i) \perp\!\!\!\perp (\mathbf{y}_j, \mathbf{t}_j) \implies \mathbf{y}_i \perp\!\!\!\perp \mathbf{t}_j$.

$g(\mathtt{t}^*), h(\mathbf{t}) = h(\mathtt{t}_2^*))$ and then $\mathbf{y} = f(\mathbf{t}, h(\mathtt{t}_2^*), \boldsymbol{\eta})$ [‡]. Then, the *functional* average effect is

$$\text{(average)} \quad \tau(g(\mathtt{t}^*)) = \mathbb{E}_{h(\mathbf{t}),\boldsymbol{\eta}} \mathbb{E}_{\mathbf{t} \mid g(\mathbf{t}) = g(\mathtt{t}^*), h(\mathbf{t})} [f(\mathbf{t}, h(\mathbf{t}), \boldsymbol{\eta})].$$

An example of a functional intervention is intervening on the cumulative dosage of a drug. In contrast, traditional interventions would set each individual dose given at different points in time.

f-positivity and Functional Effect Estimation For the causal model above to be well-defined for all functional interventions $g(\mathtt{t}^*)$, the conditional $p(\mathbf{t} \mid g(\mathbf{t}) = g(\mathtt{t}^*), h(\mathbf{t}) = h(\mathtt{t}_2^*))$ must exist. To guarantee this existence, we define *functional positivity (f-positivity)* for any $g(\mathtt{t}^*)$

$$\text{(f-positivity)} \quad p(h(\mathbf{t}) = h(\mathtt{t}_2^*)) > 0 \Longrightarrow p(g(\mathbf{t}) = g(\mathtt{t}^*) \mid h(\mathbf{t}) = h(\mathtt{t}_2^*)) > 0. \tag{6.4}$$

F-positivity says that the function of the pre-outcome variables that is being intervened on needs to have sufficient randomness when the function of the pre-outcome variables that defines the confounders is fixed. Further, under f-positivity, effect estimation for functional interventions is reduced to traditional obs-ci on data $p(\mathbf{y}, g(\mathbf{t}), h(\mathbf{t}))$. With positivity and ignorability satisfied, traditional causal estimators such as propensity scores [156], matching [157], regression [158], and doubly robust methods [159] can be used to estimate the causal effect. Focusing on regression, let f_θ be a flexible function, then $\min_\theta \mathbb{E}_{y,\mathbf{t}}[(\mathbf{y} - f_\theta(h(\mathbf{t}), g(\mathbf{t})))^2]$ would estimate the conditional expectation of interest : $\mathbb{E}[\mathbf{y} \mid h(\mathbf{t}), g(\mathtt{t}^*)]$. With θ, the effect of $g(\mathtt{t}^*)$ can be estimated by averaging the estimate of the conditional expectation over the marginal distribution

[‡] Intervening on $g(\mathbf{t})$ can be interpreted as making a *soft intervention* [154, 155] of $\mathbf{t}$ to $p(\mathbf{t} \mid \mathbf{z}, g(\mathbf{t}) = g(\tilde{\mathtt{t}}))$.

$p(h(\mathbf{t}))$:

$$\tau(g(\mathtt{t}^*)) = \mathbb{E}_{\mathbf{t}}[f_\theta(h(\mathbf{t}), g(\mathtt{t}^*))]. \tag{6.5}$$

6.3 Identification of effects of the full intervention

When positivity is violated, causal effects cannot be estimated as conditional expectations over the observed data in general. We give a functional condition, called causal redundancy (c-redundancy), that allows us to estimate the effect of the full intervention $do(\mathbf{t} = \mathtt{t}^*)$, even when positivity is violated. Specifically, c-redundancy allows us to construct a *surrogate intervention* $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ whose conditional effect at $h(\mathtt{t}')$ matches the conditional effect of interest, $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$. Let $\tilde{\mathtt{t}}$ be a fixed value of the full intervention, then c-redundancy is

Assumption. *Recall the outcome $y = f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}), \eta)$. With $\nabla_{\tilde{\mathtt{t}}}$ as gradient w.r.t. to argument $\tilde{\mathtt{t}}$:*

$$\forall \tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2), \eta, \quad \nabla_{\tilde{\mathtt{t}}} f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2), \eta)^T \nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}}) = 0.$$

In words, c-redundancy is the condition that the outcome function f uses the value of the confounder from its second argument instead of computing $h(\mathbf{t})$ from the first argument[§]. To compute the conditional effect $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$, we develop Level-set Orthogonal Descent Estimation (lode). lode's key step is to construct a surrogate intervention $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ such that

$$\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}_2^*)), \qquad h(\mathtt{t}_2^*) = h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))).$$

[§] If f transforms its first argument $\tilde{\mathtt{t}}$ into $h(\tilde{\mathtt{t}})$ as one amongst many different computations, the chain rule implies $\nabla_{\tilde{\mathtt{t}}} f(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*))^\top \nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}})$ has a term $\|\nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}})\|^2$ which is non-zero in general.

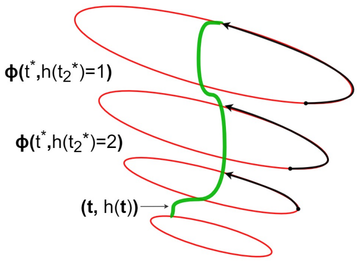


Figure 6.2: LODE's traversal.

By definition, a surrogate intervention lives in the conditional effect level-set: $\{\tilde{\mathtt{t}} : \phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*)) = \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))\}$. So LODE searches this level-set for $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$. See fig. 6.2 which plots the conditional effect level-sets with the value of $h(\mathbf{t})$ fixed (red) in $(\text{supp}(\mathbf{t}), \text{supp}(h(\mathbf{t})))$-space. Green corresponds to the observed data, $\text{supp}(\mathbf{t}, h(\mathbf{t}))$. LODE finds $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ by traversing the level-sets (**black**) to account for the confounder part mismatch $h(\mathtt{t}^*) \neq h(\mathtt{t}_2^*)$. C-REDUNDANCY ensures LODE can traverse these level-sets as it implies $\nabla_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2))\nabla_{\tilde{\mathtt{t}}}h(\tilde{\mathtt{t}}) = 0$ under the regularity conditions in theorem 6.1. Thus, under C-REDUNDANCY, surrogate interventions can be constructed by solving a gradient flow equation which guarantees identification as follows:

Theorem 6.1. *Assume C-REDUNDANCY holds. Assuming the following:*

1. *Let $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ be the limiting solution to the gradient flow equation $\frac{d\tilde{\mathtt{t}}(s)}{ds} = -\nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}(s)) - h(\mathtt{t}_2^*))^2$, initialized at $\tilde{\mathtt{t}}(0) = \mathtt{t}^*$; i.e. $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \lim_{s\to\infty} \tilde{\mathtt{t}}(s)$.*
   Further, let $h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))) = h(\mathtt{t}_2^*)$ and $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)) \in supp(\mathbf{t})$.

2. *$f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}), \eta)$ and $h(\tilde{\mathtt{t}})$ as functions of $\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}})$ are continuous and differentiable and the derivatives exist for all $\tilde{\mathtt{t}}, \eta$. Let $\nabla_{\tilde{\mathtt{t}}} f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}), \eta)$ exist and be bounded and integrable w.r.t. the probability measure corresponding to $p(\boldsymbol{\eta})$, for all values of $\tilde{\mathtt{t}}$ and $h(\tilde{\mathtt{t}})$.*

Then the conditional effect (and therefore the average effect) is identified:

$$\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \phi\left(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))\right) = \mathbb{E}\left[\mathbf{y} \mid \mathbf{t} = \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))\right] \tag{6.6}$$

In words, the key idea is that starting at $\tilde{\mathtt{t}}(0) = \mathtt{t}^*$ and following $\nabla_{\tilde{\mathtt{t}}}h(\tilde{\mathtt{t}})$ means $\tilde{\mathtt{t}}(s)$ always lies in the level-set $\{\tilde{\mathtt{t}} : \phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*)) = \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))\}$. See appendix E.1.2 for the proof. While C-REDUNDANCY is stated in terms of the gradient of the outcome function, it suffices for theorem 6.1

to assume a weaker condition about the gradient of the conditional effect: $\nabla_{\tilde{\mathrm{t}}}\mathbb{E}_{\boldsymbol{\eta}} f(\tilde{\mathrm{t}}, \tilde{\mathrm{t}}_2, \boldsymbol{\eta})^\top \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = 0$.

Surrogate Positivity In theorem 6.1, we assumed that the surrogate $\mathrm{t}'(\mathrm{t}^*, h(\mathrm{t}_2^*)) \in \operatorname{supp}(\mathbf{t})$. This condition, which we call surrogate positivity (analogous to positivity), states that for any intervention and confounder, surrogate interventions that are limiting solutions to the gradient flow equation have nonzero density conditional on the confounder value. Formally, for any intervention $\mathbf{t} = \mathrm{t}^*$

$$p(h(\mathbf{t}) = h(\mathrm{t}_2^*)) > 0 \implies p(\mathbf{t} = \mathrm{t}'(\mathrm{t}^*, h(\mathrm{t}_2^*)) \mid h(\mathbf{t}) = h(\mathrm{t}_2^*)) > 0, \tag{6.7}$$

and $\mathrm{t}'(\mathrm{t}^*, h(\mathrm{t}_2^*))$ satisfies assumption 1 in theorem 6.1. Surrogate positivity along with c-redundancy, is sufficient for full effect estimation under efc. Next, we show that the positivity assumption in traditional causal inference is a special case of surrogate positivity.

Traditional observational causal inference (obs-ci) and lode Let the confounder and intervention of interest in traditional obs-ci be $\mathbf{z}$ and $\mathbf{a}$ respectively. Assume both are scalars and ignorability and positivity hold. This setup can be embedded in efc by defining the vector of pre-outcome variables as: $\mathbf{t} = [\mathbf{a}; \mathbf{z}]$. In this setting, c-redundancy and surrogate positivity(eq. (6.7)) hold by default. Let the outcome be $\mathbf{y} = f(\mathbf{t}, h(\mathbf{t})) = f(\mathbf{a}, \mathbf{z})$, where f only depends on the first element of $\mathbf{t}$, i.e. $\mathbf{a}$[¶]. Let $e_1 = [1, 0]$ and $e_2 = [0, 1]$. In traditional obs-ci as efc, $\nabla_{\tilde{\mathrm{t}}} f(\tilde{\mathrm{t}}, h(\mathrm{t}_2^*)) \propto e_1$ and $\nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) \propto e_2$ meaning that $\nabla_{\tilde{\mathrm{t}}} f(\tilde{\mathrm{t}}, h(\mathrm{t}_2^*))^\top \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = 0$. Thus, c-redundancy holds by default. Moreover, under positivity of $\mathbf{a}$ w.r.t. $\mathbf{z}$, we also have surrogate positivity for traditional obs-ci as an efc problem. In this setting, lode computes $\mathrm{t}' = [a^*, h(\mathrm{t}_2^*)]$ by following $-\nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = [0, -1]$, which only changes the value of $h(\tilde{\mathrm{t}}_2)$, not the value of a. Thus, t^* and $\mathrm{t}'(\mathrm{t}^*, h(\mathrm{t}_2^*))$ will have the same first element and t''s second element will be $h(\mathrm{t}_2^*)$. As $\mathbf{a}$ has positivity w.r.t. $\mathbf{z}$, we

[¶] We ignore noise in the outcome for ease of exposition.

have $p(\mathbf{a} = a^*, \mathbf{z} = h(\mathtt{t}_2^*)) > 0$ which means $\mathtt{t}' \in \text{supp}(\mathbf{t})$. The estimated conditional effect is $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))] = f([a^*, z^*], h(\mathtt{t}_2^*)) = \mathbb{E}[\mathbf{y} \mid \mathbf{a} = a^*, \mathbf{z} = h(\mathtt{t}_2^*))]$, which matches the estimate in traditional OBS-CI.

IMPLEMENTATION OF LODE LODE first estimates the conditional expectation $\mathbb{E}[\mathbf{y} \mid \mathbf{t}]$; this can be done with model-based or nonparametric estimators. This is achieved by regressing $\mathbf{y}$ on $\mathbf{t}$, $\hat{f} = \arg\min_{u \in \mathcal{F}} \mathbb{E}_{\mathbf{y},\mathbf{t} \sim D}(\mathbf{y} - u(\mathbf{t}))^2$, with empirical distribution D. The surrogate intervention $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ is computed using Euler integration to solve the gradient flow equation. Euler integration in this setting is equivalent to gradient descent with a fixed step size. Other, more efficient schemes like Runge–Kutta numerical integration methods [160] could also be used. The conditional effect estimate is $\hat{f}(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))$. See algorithm 4 for a description.

6.3.1 ESTIMATION ERROR OF LODE IN PRACTICE

To compute the surrogate intervention $\mathtt{t}'$, LODE uses the gradients of $h(\cdot)$ in Euler integration. In practice, taking Euler integration steps, instead of solving the gradient flow exactly, could result in errors. Then $\mathtt{t}'$ could lie outside the level-set of the conditional effect $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \mathbb{E}_{\boldsymbol{\eta}}[f(\mathtt{t}^*, h(\mathtt{t}_2^*), \boldsymbol{\eta})]$. Further, if $h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))) \neq h(\mathtt{t}_2^*)$, LODE incurs error for conditioning on a value of the confounder that is different from $h(\mathtt{t}_2^*)$. The error due to $\mathtt{t}'$ estimation is decoupled from the error in the estimation of $\mathbb{E}[\mathbf{y} \mid \mathbf{t}]$ which adds without further amplification. We formalize this error:

Theorem 6.2. *Consider the conditional effect $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$. Let $\hat{\mathtt{t}}(\mathtt{t}^*, h(\mathtt{t}_2^*))$ be the estimate of the surrogate intervention computed by LODE, computed via Euler integration of the gradient flow $\frac{d\tilde{\mathtt{t}}(s)}{ds} = -\nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}(s)) - h(\mathtt{t}_2^*))^2$, initialized at $\tilde{\mathtt{t}}(0) = \mathtt{t}^*$. Assume the true surrogate $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ exists and is the limiting solution to the gradient flow equation.*

1. *Let the finite sample estimator of $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}]$ be $\hat{f}(\tilde{\mathtt{t}})$. Let the error for all $\tilde{\mathtt{t}}$ be bounded,*

$|\hat{f}(\tilde{\mathtt{t}}) - \mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}]| \leq c(N)$, where N is the sample size and $\lim_{N\to\infty} c(N) = 0$.

2. *Assume K Euler integrator steps were taken to find the surrogate estimate $\hat{\mathtt{t}}(\mathtt{t}^*, h(\mathtt{t}_2^*))$, each of size ℓ. Let the maximum confounder mismatch be $\max_{i\leq K}(h(\tilde{\mathtt{t}}_i) - h(\mathtt{t}_2^*))^2 = M$.*
3. *Let $L_{z,\tilde{\mathtt{t}}}$ be the Lipschitz-constant of $\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2))$ as a function of $h(\tilde{\mathtt{t}}_2)$, for fixed $\tilde{\mathtt{t}}$.*

   Let L_e be the Lipschitz-constant of $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}] = \phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}))$ as a function of $\tilde{\mathtt{t}}$.

   Assume h has a gradient with bounded norm, $\|\nabla h(\tilde{\mathtt{t}})\|_2 < L_h$.

   Assume f's Hessian has bounded eigenvalues: $\forall \tilde{\mathtt{t}}, \tilde{\mathtt{t}}_2,\ \|\nabla^2_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2))\|_2 \leq \sigma_{\mathrm{H}\phi}$.

The conditional effect estimate error, $\xi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = |\hat{f}(\hat{\mathtt{t}}) - \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))|$, is upper bounded by:

$$c(N) + \min\left(L_e\|\mathtt{t}' - \hat{\mathtt{t}}\|_2,\ 2K\ell^2\left(O(\ell) + M\sigma_{\mathrm{H}\phi}L_h^2\right) + L_{z,\hat{\mathtt{t}}}\|h(\hat{\mathtt{t}}) - h(\mathtt{t}_2^*)\|_2\right) \tag{6.8}$$

See appendix E.1.3 for the proof. Theorem 6.2 captures the trade-off between biases due to conditioning on the wrong confounder value and due to the accumulated error in solving the gradient flow equation. This accumulated error analysis may be loose in settings where the sum of many gradient steps lead to $\hat{\mathtt{t}} \approx \mathtt{t}'$, even if each step individually induces large error. In such settings, the term that depends on $\|\hat{\mathtt{t}} - \mathtt{t}'\|_2$ is a better measure of error. The maximum-mismatch M appears because Euler integrator takes steps that depend on the magnitude of the gradient which depends on the mismatch value $(h(\tilde{\mathtt{t}}_i) - h(\mathtt{t}_2^*))$. If mismatch is large for some i, the Euler step could lead to a large error for a fixed step size ℓ. We discuss the assumptions in theorems 6.1 and 6.2 in appendix E.1.1

6.3.2 Effect Connectivity and the Existence of $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$

The key element in Theorem 6.1 is the surrogate intervention $\mathtt{t}'$ such that its conditional effect given $h(\mathtt{t}')$, equals that of $\mathtt{t}^*$ and $h(\mathtt{t}_2^*)$. The orthogonality $\nabla_{\tilde{\mathtt{t}}} f^\top \nabla_{\tilde{\mathtt{t}}} h = 0$, is a functional condition that does not guarantee $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ exists in $\text{supp}(\mathbf{t})$; a necessity to compute $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \mathtt{t}']$ without additional parametric assumptions. We give a general condition called *Effect Connectivity* that guarantees the surrogate intervention exists. With conditional effect $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$, for any $\mathtt{t}^*$

$$p(h(\mathbf{t}) = h(\mathtt{t}_2^*)) > 0 \implies p(\phi(\mathbf{t}, h(\mathbf{t})) = \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) \mid h(\mathbf{t}) = h(\mathtt{t}_2^*)) > 0. \tag{6.9}$$

In words, $\mathbf{t}$ has a chance of setting the conditional effect to any possible value $\text{supp}(\phi(\mathbf{t}, h(\mathbf{t}_2)))$ given any confounder value $h(\mathtt{t}_2^*) \in \text{supp}(h(\mathbf{t}))$. An equivalent statement is that every level set of the conditional effect $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$, with $h(\mathtt{t}_2^*)$ fixed, contains an intervention for each confounder value. That is, for some $h(\mathtt{t}_2^*)$ define the level set $A_c = \{\mathtt{t}^*; f(\mathtt{t}^*, h(\mathtt{t}_2^*)) = c\}$, then $\forall h(\mathtt{t}_2^*) \in \text{supp}(h(\mathbf{t}))$, $p(\mathbf{t} \in A_c \mid h(\mathbf{t}) = h(\mathtt{t}_2^*)) > 0$.

Theorem 6.3. *Under Effect Connectivity, eq. (6.9), any surrogate intervention* $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)) \in$ *supp*$(\mathbf{t})$.

We give the proof in appendix E.1.4. Whether the intervention $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ can be found via tractable search is problem-specific. If the surrogate $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ exists $\forall \mathtt{t}^*, h(\mathtt{t}_2^*)$, then eq. (6.9) holds by definition of the surrogate. Effect Connectivity allows us to reason about values of f anywhere in $\text{supp}(\mathbf{t}) \times \text{supp}(h(\mathbf{t}))$ using only samples from $p(\mathbf{y}, \mathbf{t})$. Further, it is necessary in EFC:

Theorem 6.4. *Effect Connectivity is necessary for nonparametric effect estimation in EFC.*

We prove this in appendix E.1.5. Effect Connectivity ensures that causal models with different causal effects have different observational distributions. Then, parametric assumptions on the causal model are not necessary to estimate effects.

6.4 Experiments

We evaluate lode on simulated data first and show that lode can correct for confounding. We also investigate the error induced by imperfect estimation of the surrogate intervention in lode. Further, we run lode on a gwas dataset [161] and demonstrate that lode is able to correct for confounding and recovers genetic variations that have been reported relevant to Celiac disease [162, 163, 164, 165].

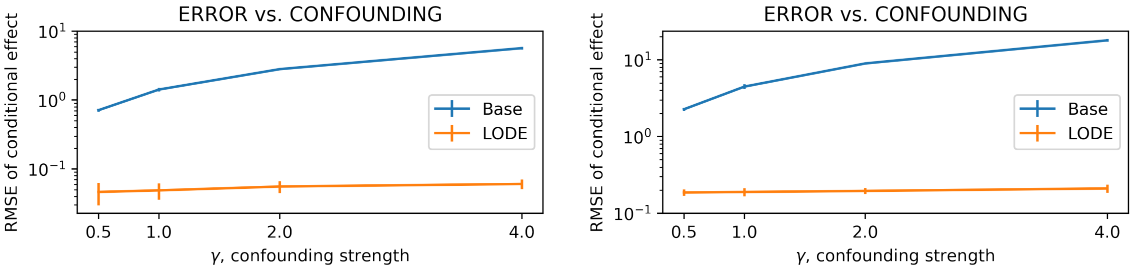
6.4.1 Simulated experiments

We investigate different properties of lode on simulated data where ground truth is available. Let the dimension of $\mathbf{t}$ (pre-outcome variables) be $T = 20$ and outcome noise be $\boldsymbol{\eta} \sim \mathcal{N}(0, 0.1)$. We consider two efc causal models, denoted by A and B with different $h(\mathbf{t})$ and $f(\mathbf{t}, h(\mathbf{t}), \boldsymbol{\eta})$:

$$(A) \quad h(\mathbf{t}) = \gamma \frac{\sum_i \mathbf{t}_i}{\sqrt{T}}, \qquad \mathbf{t} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}^{T \times T}), \quad y = \frac{\sum_i (-1)^i \mathbf{t}_i}{\sqrt{T}} + \alpha h(\mathbf{t})^2 + (1+\alpha) h(\mathbf{t}) + \boldsymbol{\eta}$$

$$(B) \quad h(\mathbf{t}) = \sum_{i : i \in 2\mathbb{Z}} \gamma \mathbf{t}_i \mathbf{t}_{i+1}, \quad \mathbf{t} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}^{T \times T}), \quad y = \frac{\sum_i (-1)^i \mathbf{t}_i^2}{\sqrt{T}} + \alpha h(\mathbf{t}) + \boldsymbol{\eta}$$

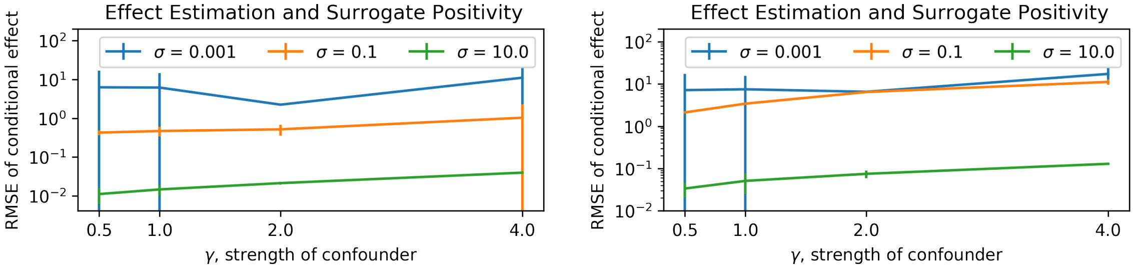
In both causal models, c-redundancy is satisfied. The constant γ controls the strength of the confounder and the constant α controls the Lipschitz constant of the outcome as a function of the confounder. We let the variance $\sigma^2 = 1$, unless specified otherwise. In the following, we train on 1000 samples and report conditional effect rmse, computed with another 1000 samples. We used a degree-2 kernel ridge regression to fit the outcome model as a function of $\mathbf{t}$. This model is correctly specified, and so the conditional $\mathbb{E}[\mathtt{y} \mid \mathbf{t} = \tilde{\mathtt{t}}]$ can be estimated well. We compare against a baseline estimate of conditional effect that is the same outcome model's estimate of $\mathbb{E}[\mathtt{y} \mid \mathbf{t} = \mathtt{t}^*]$. This baseline fails to account for confounding and produces a biased estimate of the conditional effect of $do(\mathbf{t} = \mathtt{t}^*)$, conditional on any $h(\mathtt{t}_2^*) \neq h(\mathtt{t}^*)$.



(a) Causal Model A

(b) Causal Model B

Figure 6.3: RMSE of estimated conditional effect vs. strength of confounding γ. LODE corrects for confounding and produces good effect estimates across different values of γ.



(a) Causal Model A

(b) Causal Model B

Figure 6.4: RMSE of estimated conditional effect estimate vs. the strength of confounding γ, for different levels of variance of $\mathbf{t}$, σ^2. Small σ leads to large conditional estimation error.

First, we investigate how well LODE can correct for confounding for both causal models. We let $\alpha = 1$ and obtain surrogate estimates by Euler integrating until the quantity $\mathbb{E}_{\mathtt{t}^*,h(\mathtt{t}_2^*)}(h(\tilde{\mathtt{t}}(s)) - h(\mathtt{t}_2^*))^2$ is smaller than 10^{-4} times value at initialization, where $\mathbb{E}_{\mathtt{t}^*,h(\mathtt{t}_2^*)}$ is expectation over the evaluation set. In fig. 6.3, we plot the mean and standard deviation of conditional effect RMSE averaged over 10 seeds, for different strengths of confounding. We see that LODE is able to estimate effects well across multiple strengths of confounding while the baseline suffers.

Second, we investigate LODE's estimation when surrogate positivity holds but the probability $p(\mathbf{t} \approx \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))$ is very small. This results in estimation error due to poor fitting of the outcome model in low density regions of $\text{supp}(\mathbf{t})$. We run LODE on simulated data where $\mathbf{t}$ is generated with different variances (σ^2). For small σ, the outcome model error is large when using surrogate interventions $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$, where either $h(\mathtt{t}_2^*)$ or $\mathtt{t}^*$ is large. This leads to high

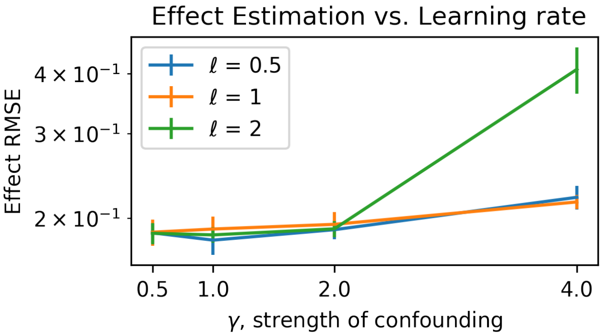
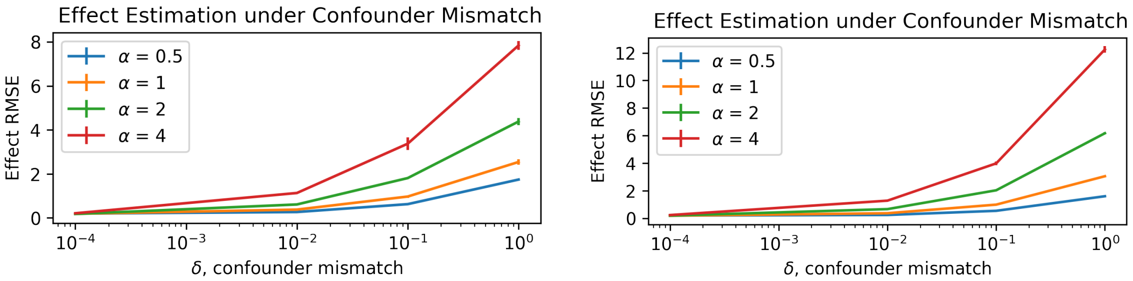


Figure 6.5: RMSE of estimated conditional effect vs. step size in Euler Integrator in causal model B. Accumulating error due to large step size in Euler integrator increases with strength of confounding.

variance effect estimation as we show in fig. 6.4 for both causal models. For various variances of $\mathbf{t}$, σ^2, we plot the mean and standard deviation of RMSE of estimated conditional effect over 10 seeds, against different γ.

Third, we investigate the bias induced due to imperfect estimation of the surrogate intervention in LODE for both causal models. We construct surrogate interventions $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ by ensuring there is confounder-value mismatch $h(\tilde{\mathtt{t}}) \neq h(\mathtt{t}_2^*)$. We do this by interrupting Euler integration when the objective $\mathbb{E}_{\mathtt{t}^*,h(\mathtt{t}_2^*)}(h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))) - h(\mathtt{t}_2^*))^2 = \delta^2 > 0$, where the $\mathbb{E}_{\mathtt{t}^*,h(\mathtt{t}_2^*)}$ is over our evaluation set upon which we estimate conditional effects. For different α, we plot in fig. 6.6 the mean and standard deviation of RMSE of estimated conditional effect over 10 seeds, against different degrees of confounder mismatch, δ. The error due to confounder mismatch is mitigated by small α, the Lipschitz-constant of the outcome as a function of $h(\mathbf{t})$. Finally, we consider how step size in Euler integration affects the quality of estimated effects. Large step sizes may result in biased surrogate estimates; this bias is captured in the accumulation error in section 6.3.1. We focus on the non-linear case in causal model B where gradient errors can accumulate(see appendix E.1.3.1). We demonstrate this error in fig. 6.5 where we plot mean and standard deviation of conditional effect RMSE against the strength of confounding, for different step sizes ℓ. We do not report results for larger step sizes ($\ell > 2$) because Euler integration diverged for many surrogate estimates.



(a) Causal Model A

(b) Causal Model B

Figure 6.6: Rmse of estimated conditional effect vs. degree of confounder mismatch δ. Error due to conditioning on a mismatched value of the confounder increases with strength of confounding but is mitigated by smoothness of the outcome function.

6.4.2 Effects in Genetics (gwas)

In this experiment, we explore the associations of genetic factors and Celiac disease. We utilize data from the Wellcome Trust Celiac disease gwas dataset [161, 162] consisting of individuals with celiac disease, called cases ($n = 3796$), and controls ($n = 8154$). We construct our dataset by filtering from the $\sim 550,000$ snps. The only preprocessing in our experiments is linkage disequilibrium pruning of adjacent snps (at 0.5 R^2) and PLINK [166] quality control. After this, $337,642$ snps remain for $11,950$ people. We imputed missing snps for each person by sampling from the marginal distribution of that snp. No further snp or person was dropped due to missingness. The objective of this experiment is to show that lode corrects for confounding and recovers snps reported in the literature [162, 163, 164, 165]. To this end, after preprocessing, we included in our data 50 snps reported in [162, 163, 164, 165] and 1000 randomly sampled from the rest.

We use outcome models and functional confounders $h()$ traditionally employed in the gwas literature. We choose a linear $h(\tilde{t}) = A^\top \tilde{t}$, where A is a matrix of the right singular vectors of a normalized Genotype matrix, that correspond to the top 10 singular values [147]. The outcome model is selected from logistic Lasso linear models with various regularization strengths, via cross validation within the training data (60% of the dataset). We defer details about the experimental setup to appendix E.2.

We then use this outcome model in LODE to compute causal effects on the whole filtered dataset. The effects are computed one SNP at a time. First, for each person $\tilde{\mathtt{t}}$, create $\tilde{\mathtt{t}}_i^1, \tilde{\mathtt{t}}_i^0$ which correspond to the ith SNP set to 1 and 0 respectively, with all other SNPs same as $\tilde{\mathtt{t}}$. Randomly sample a $h(\mathtt{t}_2^*)$ from the marginal $p(h(\mathbf{t}))$ and, using the outcome model P_θ, compute $\phi(\tilde{\mathtt{t}}, i) = \log {P_\theta(y=1|\mathtt{t}'(\tilde{\mathtt{t}}_i^1, h(\mathtt{t}_2^*)))}/{P_\theta(y=1|\mathtt{t}'(\tilde{\mathtt{t}}_i^0, h(\mathtt{t}_2^*)))}$. The average effect of SNP i is obtained by averaging across all persons: $\sum_{\tilde{\mathtt{t}}} {\phi(\tilde{\mathtt{t}}, i)}/{N}$. Any SNP that beats a specified threshold of effect is deemed relevant to Celiac disease by LODE. We use a $60 - 40\%$ train-test split, and outcome model selection is done via cross-validation within the training set. We did 5-fold cross-validation using just the training set. We use Scikit-learn [167] to fit the outcome models and for cross-validation.

Results The best outcome model was a Lasso model, trained with regularization constant 10. We select relevant SNPs by thresholding estimated effects at a magnitude > 0.1. From 1050 SNPs (1000 not reported before) LODE returned 31 SNPs, out of which 13 were previously reported as being associated with Celiac disease [162, 163, 164, 165]. In appendix E.2.2 we plot the true positive and false negative rates of identifying previously reported SNPs, as a function of the effect threshold.

SNP	Effect.	Coef.
rs13151961	0.17	0.32
rs2237236	0.17	0.00
rs1738074	−0.16	−0.23
rs11221332	−0.15	−0.24

Table 6.1: A few SNPs previously reported as relevant and recovered by LODE, with estimated effects and Lasso coefficients. LODE produces effect estimates that do not rely purely on the coefficients.

In table 6.1, we list a few SNPs that were both deemed relevant by LODE and were reported in existing literature [162, 163, 164, 165], their effects, and their Lasso coefficients. The full list is in table E.1 in appendix E.2. If LODE cannot adjust for confounding, the Lasso coefficients would dictate the effects; 0 coefficient means 0 effect. However, the two pairs of SNPs in table 6.1 show that the effects estimated by LODE do not rely solely on the

Lasso coefficients. For the first pair (rs13151961, rs2237236), the effect is the same but the coefficient of one is 0, while the other is positive. We note that rs2237236 was found to be associated with ulcerative colitis [168, 169], which is an inflammatory bowel disease that has been reported to share some common genetic basis with celiac disease [170]. For the second pair, (rs1738074, rs11221332), the magnitude of the effect is smaller for the former, but the coefficient is larger. Thus, LODE adjusts for confounding factors that the outcome model ignored.

6.5 Discussion

When positivity is violated in traditional OBS-CI, not all effects are estimable without further assumptions. In such cases, practitioners have to turn to parametric models to estimate causal effects. However, parametric models can be misspecified when used without underlying causal mechanistic knowledge. We develop a new general setting of observational causal effect estimation called estimation with functional confounders (EFC) where the confounder can be expressed as a function of the data, meaning positivity is violated. Even when positivity is violated, the effects of many functional interventions are estimable. We develop a sufficient condition called functional positivity (F-POSITIVITY) to estimate effects of functional interventions. Such effects could be of independent interest; like the effect of cumulative dosage of a drug instead of joint effects of multiple dosages at different times.

Second, we prove a necessary condition for nonparametric estimation of effects of the full intervention. We propose the C-REDUNDANCY condition, under which, the effect of the full intervention on **t** is estimable without parametric restrictions. We develop Level-set Orthogonal Descent Estimation (LODE) that computes surrogate interventions whose effects are estimable and match a conditional effect of interest. Further, we give bounds on errors (theorem 6.2) induced due to imperfect estimation of the surrogate intervention. Finally, we empirically demonstrate LODE's ability to correct for confounding in both simulated and real data.

Future. A few directions of improvement remain which we elaborate next. First, f-positivity may not hold for all functions $g(\mathbf{t})$ that we want to intervene on. Instead, one could compute a "projection" g_Π to the space of functions that satisfy f-positivity and inspect the effects defined by g_Π instead. A second direction of interest is to let $h(\mathbf{t})$ only account for a part of the confounding, meaning ignorability is violated. This bias could be mitigated under smoothness conditions of the outcome function and its interaction with the degree of violation of ignorability.

Finally, lode's search strategy is Euler integration, which is equivalent to gradient descent with a fixed step size. Optimization techniques like momentum, rescaling the gradient using an adaptive matrix, and using second order hessian information, speed up gradient descent. However, if there are many local or global minima for $(h(\tilde{\mathtt{t}}) - h(\mathtt{t}_2^*))^2$, such techniques will result in a different solution than Euler integration, which could mean that effect estimates are biased. One extension of lode would allow for search strategies that use such techniques.

Part III

An Empirical Study in Building and Transporting CHD risk Models

AI models for CHD risk assessment

The first two parts of this thesis developed methodology for two problems: out-of-distribution (OOD) generalization and causal effect estimation. These two problems have a common theme: use observed data from one distribution (training/confounded distribution) to support decisions under a different distribution (test/interventional distribution). This thesis' last part tackles a task in healthcare that fits the same theme: survival analysis for risk estimation.

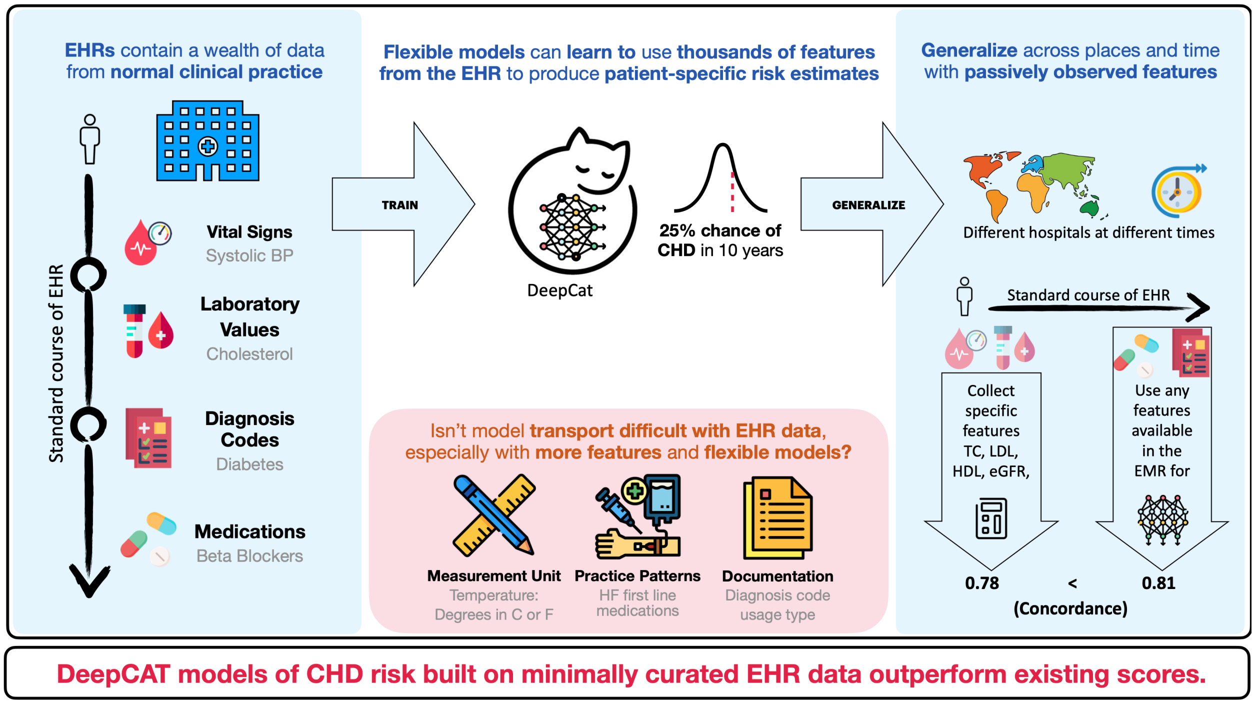
Focusing on coronary heart disease (CHD), the task is to estimate the chance that a patient develops CHD within a time horizon. Risk estimates help care providers plan preventative interventions (as opposed to curative treatment that occurs after a particular diagnosis is confirmed). Models of risk are necessarily trained on retrospective data but are deployed prospectively which means, in the same spirit as the first part of this thesis, risk models need to generalize beyond the training distribution. In healthcare, such generalizability is called *transportability*. Establishing model transportability on external populations (such as patients from a different hospital) is a crucial step before deploying said model. In the following chapter, we build transportable AI models of CHD risk from minimally curated EHR data.

7 | Performant and Transportable modeling of CHD risk from minimally curated Hospital-Scale EHRs

(*This chapter presents part of an ongoing project authored by, among others, Shreyas Bhave (Columbia DBMI), Aahlad Puli, Mark Goldstein, Mert Ketenci (Columbia DBMI), Noémie Elhadad (Columbia DBMI), Adler Perotte (Columbia DBMI), and Rajesh Ranganath*)

7.1 Introduction

Developing accurate risk assessment models for cardiovascular disease (CVD) is a foundational challenge in the primary prevention of CVD. Existing clinical guidelines from the American Heart Association (AHA), American College of Cardiology (ACC), and the World Health Organization (WHO) all recommend that 10-year CVD risk be used to guide the type and intensity of preventative treatments [171, 172]. CVD risk scores are often computed using survival models that account for the natural censoring in clinical data [173]. Popular risk scores include the Framingham Risk Score (FRS) [174, 175] and the Pooled Cohort Equations (PCE) [176]. These risk scores


EHRs contain a wealth of data from normal clinical practice
Standard course of EHR
Vital Signs
Systolic BP
Laboratory Values
Cholesterol
Diagnosis Codes
Diabetes
Medications
Beta Blockers
TRAIN
Flexible models can learn to use thousands of features from the EHR to produce patient-specific risk estimates
DeepCat
25% chance of CHD in 10 years
Isn't model transport difficult with EHR data, especially with more features and flexible models?
Measurement Unit
Temperature: Degrees in C or F
Practice Patterns
HF first line medications
Documentation
Diagnosis code usage type
GENERALIZE
Generalize across places and time with passively observed features
Different hospitals at different times
Standard course of EHR
Collect specific features TC, LDL, HDL, eGFR,
Use any features available in the EMR for
0.78
<
0.81
(Concordance)

DeepCAT models of CHD risk built on minimally curated EHR data outperform existing scores.

Figure 7.1: Flexible survival models built from minimally curated EHR data perform better than existing score and transport well.

perform around a c-index of 0.70 [176, 177]), and have established utility by verifying whether a model *transports* to new settings [178, 179, 180].

The widespread adoption and use of electronic health records (EHRs) [181] has spurred the study of alternative approaches to computing risk scores [182, 183, 184, 185] that ingest large amounts of the EHR. For survival analysis specifically, training on hospital-wide EHRs has shown promising results for predicting lung cancer survival [186], Chronic Kidney Disease [187] and CVD [182, 188].

In pilot studies, the use of artificial intelligence (AI) for survival analysis on EHRs yield models for CVD that improve over existing risk scores in the population on which they were trained [189, 190]. However, there has been limited investigation into the transportability of these feature-rich and flexible survival models (Figure 7.1). Several drivers are responsible. First, larger feature sets require more effort to harmonize data for validation. Second, the advent of EHRs and easy-to-use

AI tools has made it easier for different places to have models trained on their own data, which leads to questioning the need to study transportability.

Considering that training occurs on historical data with the intended use being on future populations, all model deployment *requires transporting* a model. This transport implies that performance within-institution on historical data is not the definite criteria for a predictive model. For example, a model that is performant on historical data may degrade in the future due to changes in unmarked patient demographics, such as a shift in race/ethnicity [191], or changes in data quality or collection [192]. Further, online testing within a hospital can also be too slow for long-term chronic diseases like Coronary Heart Disease (CHD) because labels may only appear years after predictions were made. Finally, just as in the sciences (for example, experimental physics), one needs to rule out bugs in the experimental setup, such as label leakage and reproducibility issues, which have been shown to give an unrealistic picture of model performance in healthcare [193, 194]. Validating externally is at the core of any argument that a model is applicable beyond the retrospective data used to train it. **This paper performs an external validation of flexible survival models for CHD by fitting dozens of survival models across two large health systems.**

The standard approach to assess transportability is to compare a metric like concordance at an external institution against a baseline or reference [195]. However, when concerns arise about the external performance, the standard approach does not separate the role of model transport on external performance from that of the nature of the external population. As an example, the external hospital could have a larger fraction of a population on which it is harder to make predictions. Low performance metrics in this hospital do not stem from transporting a model, but rather reflect that it is harder to predict on the underlying data. Thus, when external performance raises concerns, it may be helpful to compute further quantitative metrics that adjust for the varying hardness across populations [196]. As one such quantitative metric, we propose the *T-gap* which computes the difference in performance between the transported model and an externally trained model, with the latter providing an empirical characterization of the hardness of the external population.

Empirically, we build dozens of survival models — which we call DeepCAT— for coronary heart disease using different choices of feature sets and evaluate their transportability. DeepCAT models achieve high absolute concordance and low absolute Brier score; that is, DeepCAT models transport well (Tables 7.1 and 7.2) in the populations we study.

As a whole this paper makes several contributions

- AI models of CHD risk built on minimally curated EHR data perform better internally and externally than the FRS and PCE scores, as well as the recently introduced PREVENT score [181, 197].
- Models including a broad set of diagnoses perform best. This shows the value of using features beyond the small curated sets common in previous risk scores for building transportable models.
- Using T-gaps, we find that variation in external performance across demographic subgroups is driven by the differences between the data within the subgroups rather than disproportionate model transport across subgroups.

7.2 Results

Comparing Cohorts. We consider two outcomes. The first outcome, referred to as *soft CHD*, follows from what was used in the framingham score [174, 175]: which accounts for angina pectoris (ICD 9 prefix: 413), acute myocardial infarction (ICD 9 prefix: 410) and coronary insufficiency (ICD 9 prefix: 411), and hard CHD which accounts for acute myocardial infarction (ICD 9 prefix: 410) and The second outcome, referred to as *hard CHD*, follows from the CHD definition in

PREVENT score [181, 197]: acute myocardial infarction (ICD 9 prefix: 410 and ICD 10CM prefix I21) and subsequent myocardial infarction (ICD 10CM prefix: I22).

We use data derived from EHRs in-use at two large, academic institutions: Columbia University Irving Medical Center (CUIMC) and New York University Langone Health (NYULH). The demographic characteristics of the cohorts are shown in Table F.7. CUIMC and NYULH have similar cohort size, with over 1 million patients and over 20 million encounters, and the prevalence for both SoftCHD and Hard CHD is similar in both systems: $\approx$ 5% and $\approx$ 2.5% respectively. Both cohorts have significantly more female than male patients; sex differences are diminished among CHD cases.

Comparing DeepCat to Existing Scores. We introduce a survival model, named DeepCat, to study the transportability of flexible, feature-rich survival models for both soft and hard CHD. DeepCat uses the SetTransformer++ [198] which can encode features like labs, vitals, drugs, conditions, smoking status, age, and demographics into a survival distribution for time-to-event (Section 7.5.3). DeepCat makes no assumptions on the shape of the survival distribution. We train candidate DeepCat models for soft CHD at each institution with different feature combinations (conditions and drugs, conditions only etc.). One feature set is chosen for each cohort based on the soft CHD concordance on the external cohort. We do this to replicate the model building study in two locations. We then train DeepCat models for hard CHD on each cohort with their respective feature sets. The DeepCat models are then evaluated internally and externally against established scores. For performances reported in the text, confidence intervals are in the appendix (table F.6).

Table 7.1 reports concordance of DeepCat models for both soft and hard CHD outcomes. In summary, DeepCat outperforms the existing scores, FRS and PCE, and the newly designed score PREVENT [181, 197], in every situation. For soft CHD, DeepCat models built at CUIMC and NYULH attain at least 0.84 concordance internally and 0.81 externally. FRS and PCE attain $\leq$

Table 7.1: Comparing established risk scores to DeepCat models at CUIMC (test N=66,319) and NYULH (test N=87,151). DeepCat trained at either NYULH or CUIMC outperforms established scores at both sites.

	SoftCHD		HardCHD	
	CUIMC	NYULH	CUIMC	NYULH
FRS [174, 175]	.73 (.724, .742)	.74 (.735, .749)	.79 (.778, .802)	.75 (.734, .754)
PCE [176]	.70 (.688, .709)	.73 (.726, .741)	.75 (.739, .765)	.74 (.732, .757)
PREVENT [181, 197]	.75 (.74, .758)	.78 (.768, .782)	.81 (.796, .818)	.79 (.780, 0.799)
DeepCat NYULH	**.81** (.802, .817)	**.84** (.831, .842)	**.85** (.837, .858)	**.85** (.839, .857)
DeepCat CUIMC	**.85** (.849, .861)	**.81** (.806, .822)	**.88** (.869, .888)	**.82** (.813, .829)

0.74 at both institutions, which is in line with prior EHR studies of these models [199, 200, 201]. DeepCat models trained at either institution outperform PREVENT both internally and externally (DeepCat ≥ .81, PREVENT ≤ .78). For hard CHD, DeepCat models built at CUIMC and NYULH attain at least 0.85 concordance internally and 0.82 externally. At CUIMC, all existing scores achieve < 0.81 concordance, while the DeepCat models trained at either institution achieve better than 0.85 concordance. At NYULH, all existing scores achieve < 0.79 concordance, while the DeepCat models trained at either institution achieve better than 0.82 concordance.

Discriminative performance similar to that in table 7.1 holds for a variant of concordance that emphasizes discriminating small and large event times, when adjusting for censoring and restricting to a subpopulation with time-to-event of at least 1 year; see tables F.1 to F.4 in appendix F.1.2.

To evaluate calibration, we computed the Brier score at 5 years. Table 7.2 shows that DeepCAT models for soft and hard CHD outcome achieve good brier score both internally and externally (<0.05 [202]). We show calibration curves in appendix F.1.1.

Exploring Subpopulation Performance. Risk scores should be evaluated across subpopulations to understand how performance varies across groups that appear with lower frequency. We first study the difference in external performance (concordance) between subpopulations. The left panels (A) of the subplots of Figure 7.2 show histograms of these differences between all pairs of subpopulations for both soft and hard CHD outcomes. The differences in external performance can be as large as 0.20 of concordance. This difference could indicate that the model transports worse on some subgroups relative the others. However, the two panels in the middle (B-C) of Figure 7.2 show that differences in performance between subgroups correlate less with differences in T-gap than with differences in the performance of the externally trained models. In turn, the differences in external performance between subgroups would not be resolved by replacing an external model with an externally trained model. That is, variable transport across groups does not drive the differences in external performance.

To verify that differences arise for reasons other than transport, the right panels (D) of Figure 7.2 plot the external performance and the performance of the externally trained model in on the various subpopulations, ordered by external performance; the two performances increase in concert with one other. These plots suggest that the differences in external performance between subpopulations primarily sprout from differences in the data distribution of sub-population rather due to model transport.

Focusing on performance internally, for both soft and hard CHD outcomes DeepCat models perform better on Asian and White populations than Black populations at both institutions. For both outcomes, the performance in Asian populations at CUIMC is particularly higher than the other racial or ethnic categories, with the largest gap being between Asian Females and Hispanic/Latino Males: CUIMC Asian Females $\geq$ 0.96 versus CUIMC Hispanic/Latino males $<$ 0.81. In both institutions, models also perform better on females compared to males. For soft CHD, NYULH females 0.85 versus NYULH males 0.82 and CUIMC females 0.86 versus CUIMC males

Table 7.2: Performance and Transportability of DeepCat models measured by the IPCW-Brier score at 5 years (lower is better). Both internally and externally, DeepCat models achieve good brier scores (< 0.05 is considered excellent [202]).

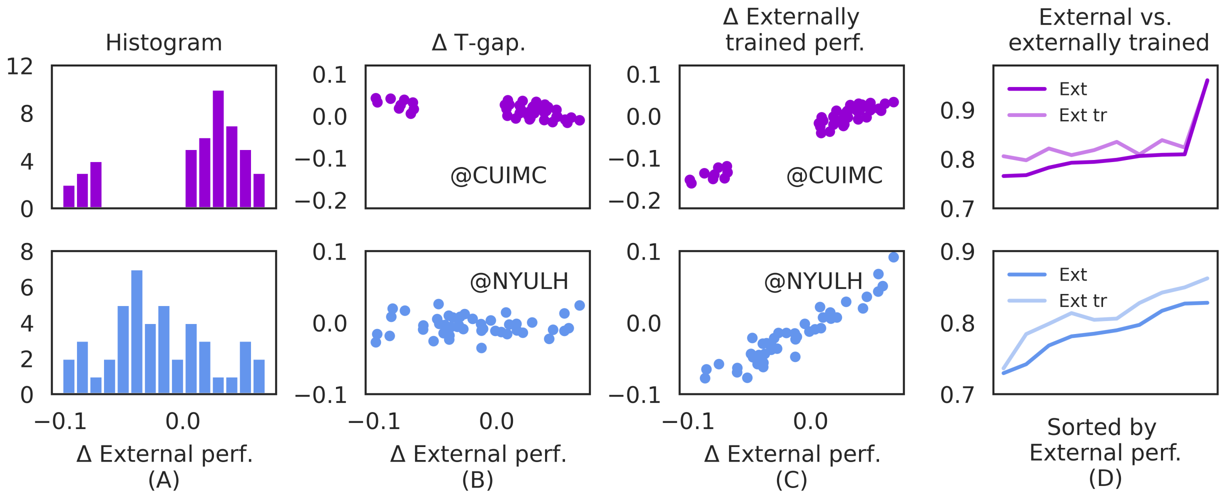
		Internal	External
	Model	Performance	Performance
Soft CHD.	NYULH	.05 (.039,.055)	.04 (.038,.042)
	CUIMC	.04 (.036,.039)	.06 (.045,.063)
Hard CHD	NYULH	.03 (.028, .032)	.02 (.015,.017)
	CUIMC	.02 (.014, .017)	.034 (.032,.036)

0.82. For hard CHD, NYULH females 0.86 versus NYULH males 0.83 and CUIMC females 0.89 versus CUIMC males 0.84.

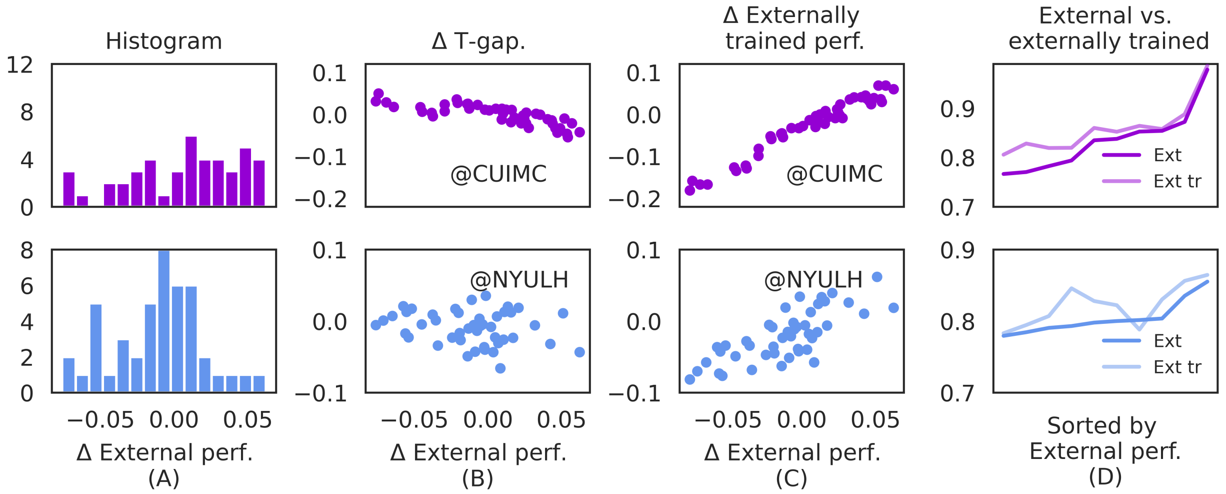
Which features are needed for CHD prediction? We performed the feature set selection study (based on validation soft CHD performance) at two locations, which resulted in two models, DeepCat-NYULH, and DeepCat-CUIMC. While DeepCat-NYULH uses age, conditions, and demographics, DeepCat-CUIMC additionally uses smoking status, lab measurements, and drug brands. Given that different feature sets were chosen in different places, a natural question is *which features are actually important for CHD?*. We further ablated to see if this feature difference important.

For soft CHD, when using the smaller set of NYULH features, the CUIMC model maintains .85 internal performance (concordance) and external performance is .80. Similarly for hard CHD, when using the smaller set of NYULH features, the CUIMC model maintains .87 internal performance (concordance) and external performance is .82. Therefore, on both outcomes, when trained on a smaller set of features, the CUIMC model retains its performance gain of several points over the existing scores FRS, PCE, and PREVENT.

DeepCat models featuring conditions observe whether any of 669 possible codes are present in



(a) Subpopulation performance differences for*soft CHD* models trained at NYULH (top) and CUIMC (bottom) and evaluated externally.



(b) Subpopulation performance differences for*hard CHD* models trained at NYULH (top) and CUIMC (bottom) and evaluated externally.

Figure 7.2: For both outcomes, external performance differences correlate more with internal performance differences than with T-gaps. Left panels (a.(A), b.(A)) show the differences in external performance across pairs of subpopulations. The two plots in the middle (a.(B-C), b.(B-C)) show that external performance differences are more correlated with internal performance differences than with differences in T-gaps. The right panels (a.(D), b.(D)) orders subpopulations based on increasing external performance and shows the external performance and the performance of the externally trained model; the two performances tend to increase in concert with one other.

a patient's medical history. Models with no demographic features may sometimes be required; on predicting softCHD subject to the constraint of no demographic features, DeepCat-CUIMC achieves .85 concordance internally and .81 externally while DeepCat-NYULH achieves .83 internally and .80 externally, both nearly retaining their performance. On predicting hardCHD, subject to the constraint of no demographic features, DeepCat-CUIMC achieves .87 concordance internally and .82 externally while DeepCat-NYULH achieves .84 internally and .85 externally, both nearly retaining their performance.

Which Features Contribute to Performance and Transportability? We next explore which features contribute to improved performance within and across institutions. Focusing on soft CHD, Figure 7.3 plots the change in concordance when a feature group gets added to any other subset of the remaining features. For both internal and external performance, including conditions provides the largest improvement in concordance. The median improvement both internally and externally for including conditions is 0.06 versus at most 0.01 improvement for any other inclusion. Finally, looking at the change in performance when including measurements in Figure 7.3 shows that including more features does not always lead to better generalization, with most models decreasing in external performance when measurements are included.

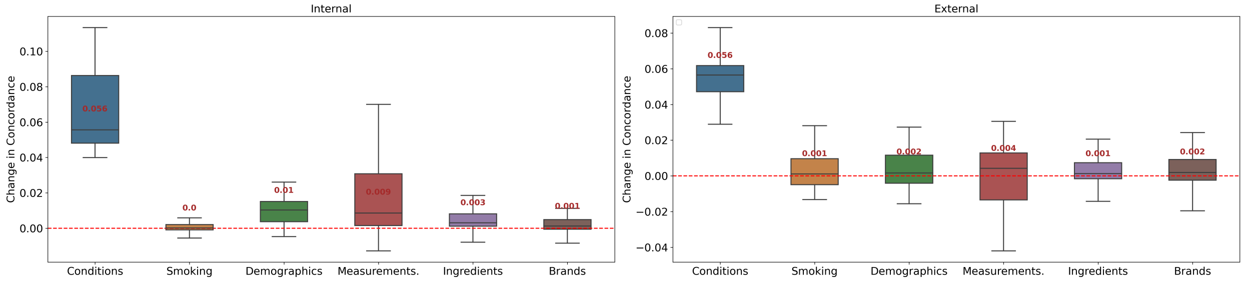


Figure 7.3: Condition features contribute most to model improvements at both institutions. For each feature grouping, each box-and-whisper plot displays concordance differences between many models built with and without the feature grouping, for all combinations of the remaining features and across two sites. Models built in either institution improve in concordance at both institutions most notably when condition features are included; median improvement 6 concordance points versus less than 1 point for all other features. Notably, for external performance, most models decrease in concordance when measurement features are added.

Best Internally or Best Externally. Selecting models on best internal performance may skip those with better external performance. On soft CHD, NYULH models improve from 0.782 concordance to 0.809 at CUIMC when selecting externally-best instead of internally best models. These externally best models also retain near optimal performance internally: while the external performance gain is 0.03, the internal performance loss is only 0.003, an order of magnitude smaller. For CUIMC models, the effect is less pronounced.

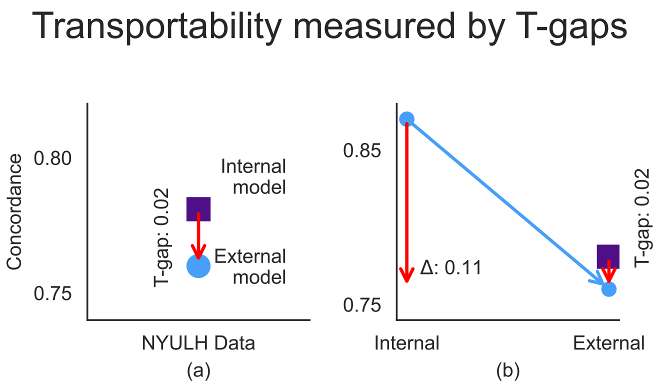
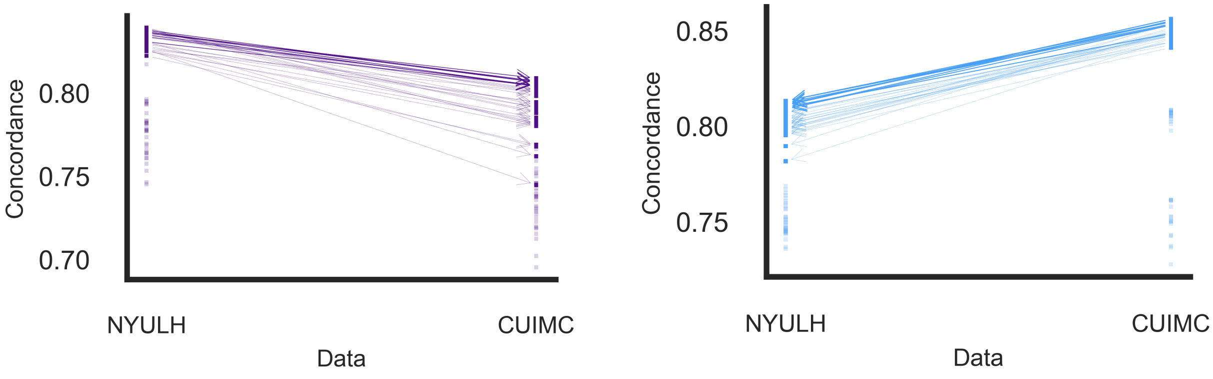


Figure 7.4: understanding tgap

External Performance Differences versus T-gaps. T-gaps contextualize performance of a model transported to an external site by comparing against a model trained at the external site. The left panel of fig. 7.4 shows visual intuition on how the T-gap is computed. However, computing T-gaps would not be useful if they always matched the difference between internal and external performance (denoted Δ). The right panel of fig. 7.4 shows that models with large performance differences across institutions may have small T-gaps. A model that has a large absolute difference in 5-year IPCW-weighted concordance of 0.11 between internal and external performance. But, the T-gap is 0.02 meaning that the model drops very little performance relative to the externally-trained model (NYULH) and indicates that model transport does not drive the large difference between internal and external performance.



(a) NYULH models

(b) CUIMC models

Figure 7.5: The bold squares have internal performance above 0.82. Arrows connect a model's external performance to its internal performance. Bold arrows show that the top externally-performing models have near-top internal performance while near-top internally performing models have variable external performance ranging from good to bad.

7.3 Discussion

DeepCat Models are Predictive of soft and hard CHD and Transportable Across Populations. We developed DeepCat, a flexible and feature-rich survival model for soft and hard coronary heart disease (CHD). We demonstrate that DeepCat's performance, both internal and external, surpasses established scores such as FRS, PCE, and PREVENT at two large metropolitan hospitals, NYULH and CUIMC. DeepCat, trained at either institution on minimally-curated EHR features, achieves concordance greater than 0.81 on soft CHD and 0.82 on hard CHD at both institutions. The models maintain this good discrimination when considering patients with longer times to events (over one year) and when evaluating predictions over a five-year horizon, which aligns with clinically relevant timeframes [171]; see appendix F.1.2. We observe that strong external performance does not come at expense of internal performance; on soft CHD, models that are good externally are good internally (Figure 7.5). Our findings also confirm that DeepCat models transport equitably across diverse clinical subpopulations, discussed below.

Flexible Feature-Rich Models. In machine learning, flexible models are typically built by incorporating a wide range of potentially relevant features and optimizing over a large function class, enabling the model to access useful relationships for prediction where they exist [203]. In clinical settings, however, modeling is often approached differently, using carefully chosen feature sets (for example, just features known to be mechanistically relevant [197]) to minimize potential dependence on non-transportable associations, or *shortcuts* [5]. Moreover, a common motif in clinical settings is that feature sets should be kept small to avoid requiring patients to undergo numerous measurements at the time of risk assessment and to avoid over-screening [181]. This line of reasoning involves a trade-off, however, as excluding relevant features beforehand can introduce statistical bias.

Further, choosing between large feature sets and over-screening may also be a false dichotomy: if we do need to expand the set of features to decrease bias, must it always require additional screenings? Recent work in clinical tasks — such as predicting lung cancer survival [186], Chronic Kidney Disease [187], CVD [182, 188], cardiogenic shock [204] and others [182, 183, 184, 185] — represents a shift in this thinking, hinging on the idea that there is significant predictive value in whichever data *happens* to be available about a patient at the time of risk assessment, even if not each possible feature is always observed. With this data, a computer can decide whether to use a given observation. Such models can be built with methods that handle missingness [190] and can then be validated on external populations to see if they generalize to new patient populations [178, 179, 180]. In this work, we observe that such an approach—using all available features in the EHR and validating for best external performance—leads to good models for CHD.

Demographics Features. Our findings demonstrated the utility of demographic features such as race, ethnicity, and gender, with both DeepCat models using them when selecting models for optimal external performance. On the other hand, recent work on cardiovascular risk modeling [181, 197] has discounting the use of these features a priori. While it is true that features like race are social constructs and not directly linked to disease physiology, they can serve as a proxy for missing information that is physiologically relevant. For example, South Asian males are at increased risk of heart disease, even after adjusting for traditional risk factors [205]. A model with genetic data could in principle discern this risk, but in its absence, demographic features can help identify such high-risk groups. Recognizing that certain settings may require the exclusion of race, we also investigated the performance of models without demographics on both soft and hard CHD outcomes and found that they still performed better than existing scores, albeit with a marginal decrease in performance relative to DeepCat models with demographics ($\leq$ 1 point decrease in concordance). The current results suggest, at least when predicting CHD with flexible models, that it may be advantageous not to rule out demographics a priori.

Condition Features Promote Transportability Condition features (diagnosis codes) significantly improve both internal and external performance of DeepCat models across all feature combinations, with a median improvement in concordance of 6 points for including conditions compared to less than 1 point for other features. Such diagnosis codes are often associated with two concerns: that usage varies across practices [206] and that codes may appear in EHRs months after diagnosis, raising potential concerns about label leakage [193]. However, the strong external performance of DeepCat models suggests that variations in coding practice may be sufficiently mitigated by our standardization of data in the OMOP format. Further, strong performance on long times-to-event data suggests that performance gains due to label leaks are unlikely.

In relationship to other scores, performance improves progressively when adding conditions, from FRS and PCE (2 conditions) to PREVENT (4 conditions) to DeepCat (669 conditions) (performance in Table 7.1 and frequent features listed in Table F.13). Two more cardiovascular risk scores that are newer than FRS, but study different outcomes (composite CHD and stroke), are ASSIGN [207] and QRISK3 [182]; these scores also include new features for diagnoses like arthritis, chronic kidney disease, migraines, Lupus, and mental illness. Looking forward, it may be advantageous to use a large number of conditions from the outset and enhance validation practices, rather than gradually expanding small feature sets over time.

Measurements may harm external performance. Unlike conditions, we may expect labs and vitals (e.g., cholesterol levels) to support transportable risk models due to sufficient standardization into common measurement units. However, we find that incorporating labs and vitals in models can negatively impact transportability, with most models showing decreased external performance when these measurements are included (Figure 7.3). We hypothesize that conditions (e.g. a diagnosis of high cholesterol) excel over measurements (one instantaneous cholesterol test) because diagnoses can function as a summary statistic that a lab value has been abnormal over a duration of time rather in any one instant. We additionally hypothesize that variations in measurement missingness patterns across institutions may be the cause. Missingness can vary based on institutional practices or patient-specific factors (e.g., labs missing at random for a subpopulation in one location but not in another). Addressing this issue requires investigate assumptions about missingness (e.g., monotone missingness patterns [208]) to ensure risk estimation under the complete data distribution. Generative models of complete feature distributions, which could account for missingness patterns, could be beneficial for addressing missingness despite the inherent challenges of generative modeling [209, 210].

Subpopulation Transportability Prior studies have indicated that CHD risk models exhibit varying absolute performance across clinically relevant subpopulations [211, 212]. Our examination of subpopulations confirms this external performance disparity (Figure 7.2 left panels). To better understand the source of variability, we introduced and analyzed T-gaps, which measure the extent to which external performance can match the performance of an externally-trained model (*can an NYULH-trained model do as well at CUIMC as a CUIMC-trained model?*). In the middle left panel (B) in Figure 7.2, T-gap disparities are not correlated with external performance meaning that fact that an NYULH model matches a CUIMC model on one subpopulation but not another is not predictive of which subpopulation the NYULH model predicts better on. Further investigation shows that external performance are correlated with performance of externally-trained models, meaning that it is the inherent differences in the subpopulation data at either site that must at least partially drive external performance variation across subpopulations (fig. 7.2 right panels).

To better understand the source of these performance discrepancies, we investigated internal performance. DeepCat models perform better in Asian and White populations compared to Black populations, consistent with challenges observed in existing risk models such as FRS and PCE [181]. Additionally, our findings reveal that models perform worse in male populations than in female populations, despite the higher incidence of CHD in males [213]. This highlights the necessity for gender-specific adjustments in risk models to address these discrepancies effectively. More broadly, the disparity in model performance across subpopulations, especially in higher incidence groups, underscores the importance of deeply understanding EHR data nuances across groups. Such understanding can come from considering unaccounted social determinants of health or biases in data collection, for example why some subpopulations report their demographics more completely than others. Such steps are crucial for mitigating performance variability across subpopulations and enhancing overall model accuracy and transportability.

OMOP and Large Observational Health Data Networks. Standardizing data across institutions using the OMOP common data [*] model plays a crucial role in this work. In this study, data from two institutions were harmonized such that measurements (labs and vitals, LOINC codes), diagnoses (SNOMED or ICD-9/10), and medications (RxNorm) shared a common meaning across the institutions. The process of standardizing data into the OMOP format was relatively straightforward. CUIMC already maintained an OMOP view of their data, and the authors at NYULH were able to create a similar view using a simple set of queries. This standardization is not unique to our study; the OMOP format has been employed in thousands of studies within the OHDSI

[*] https://www.ohdsi.org/data-standardization/

network [†], demonstrating its utility and broad applicability. The success and transportability of flexible, feature-rich models highlight the potential for deploying these models across various hospital systems, provided that health records are available in standardized formats. This study lays the groundwork for extending our approach to multi-site cases, leveraging the capabilities of large observational health data networks to evaluate and enhance model generalizability on a larger scale.

Causality: Toward Safe Prospective Deployment Assessing transportability is a crucial step toward the prospective deployment of predictive models, but it is not sufficient for ensuring their safe deployment. Various dataset shifts can occur that affect prospective model performance, with treatments being a significant example. Treatment decisions based on the model may induce their own shift. For instance, patients in the training data for current models may have been classified as high risk by older risk scores (e.g., FRS) and subsequently treated by providers. This treatment could have delayed their CHD onset, making them appear healthier in the data than they would be without intervention. Consequently, predictive models may incorrectly classify these patients and similar ones as low-risk. Instead, models that can account for the effect of interventions (like high dose statins) on a reduction in CHD risk are more suitable for assigning treatments to patients.

7.4 Limitations

Phenotyping based on rule-based codes is a challenging problem. For example, the true diagnosis of CHD may be prior to the one derived (left-censoring). As one check, under the assumption that left-censoring often comes from patients that enter the system just after diagnosis, we omitted data with time-to-CHD less than one year, and observed negligible effects on model performance; this would indicate that left-censoring is not a major concern in this data.

[†] https://dash.ohdsi.org/research

When building or evaluating risk scores, censored data is unavoidable. In line with previous risk scoring studies, we build survival models, since standard classification may only estimate risk in the uncensored population and cause disparities for infrequent patients. We followed current best practices by additionally estimating censoring models and using them in re-weighted estimators of metrics such as Brier score and IPCW-concordance, intended to estimate model performance on the complete data distribution [214, 215, 216].

Leveraging the OMOP format allowed us to replicate model training across institutions seamlessly. There may however be discrepancies; for example, conditions could be recorded with varying levels of granularity between the two institutions without being mapped to the same features. In this work, we evaluated models across institutions and found that, on the contrary, condition features do help build models that transport, and models built with conditions were less susceptible to transportability failures due to other features (i.e. e.g. data discrepancies in non condition features such as drugs were less detrimental when conditions were included).

Both institutions in this study are based in New York City, meaning there is potential for overlapping patients and care patterns, yielding optimistic estimates of how well CHD models transport. With that said, these findings may still generalize; New York City is among the most diverse cities, and features many sub-populations and sub-communities, each with distinct income levels and health backgrounds. The two hospitals studied are large, diverse systems featuring four large centers and dozens of small geographically separate practices.

Finally, while two institutions is a start, we believe that such studies should be extended to the multi-site setting, since this is more in line with the aim of building transportable models; whether a model is to be used in many institutions or just one, the population the model must serve, going beyond the training data, is not just one alternative population, but rather a constantly evolving heterogeneous one.

7.5 Methods

7.5.1 Data

Standardized Data using OMOP. To replicate cohort creation across CUIMC and NYULH, we use a standardized version of the data using the Observational Medical Outcomes Partnership (OMOP) common data model [217]. We obtain data for labs, vitals, drugs, conditions, smoking status and demographics including gender, ethnicity, race, and age. We standardize conditions using the SNOMED-CT terminology, labs using the LOINC terminology, and drugs using the RxNorm vocabulary.

Phenotype. The first outcome, referred to as *soft CHD*, follows from what was used in the framingham score [174, 175] and in prior work [190, 200]: which accounts for angina pectoris (ICD 9 prefix: 413), acute myocardial infarction (ICD 9 prefix: 410) and coronary insufficiency (ICD 9 prefix: 411). The second outcome, referred to as *hard CHD*, follows from the CHD definition from [181, 197]: acute myocardial infarction (ICD 9 prefix: 410 and ICD 10CM prefix I21) and subsequent myocardial infarction (ICD 10CM prefix: I22). We map these ICD codes and their children to SNOMEDs whose descendants are then found by using the OMOP *Concept* table to map SNOMEDs to *Concept IDs*, and then using the OMOP *Concept Relationship* table to retrieve all Concept IDs that "*Map To*" the parent concepts. In total, we retrieve 162 distinct SNOMED codes for soft CHD and 206 for hard CHD.

During phenotyping, the time of the CHD event or the censoring time is extracted for each patient. The number of months between each interaction time (when features are measured) and the CHD or censoring time becomes the time-to-event label. An additional indicator records whether the time is a CHD time or a censoring time. These labels are used for survival modeling. We follow the same process of cohort and dataset construction for both soft and hard CHD outcomes.

So we say CHD when explaining the process of construction.

Cohort Definition. We define a broad adult cohort including data from inpatient, outpatient and emergency room visits. We set the following inclusion criteria: (1) patients must be above 18 and (2) to ensure data quality, patients are required to have at least 5 distinct months of data recorded (not necessarily consecutive).

Among included patients, we record the first time of any CHD diagnosis according to the phenotype. For patients with no CHD code, we record the last observed data point in the EHR as their censoring time. We give complete cohort details in Appendix F.2. We report soft CHD patients statistics in Table F.7 and data statistics in Table F.8.

Feature Extraction. We start with a list of relevant lab measurements (Table F.11) and vital signs (Table F.12) present at both institutions, and with all possible conditions (Table F.13) and medications (Table F.14). Among features with nonzero occurrence at both institutions, if at least one hospital has frequency (in % of patients) above a specified threshold (1% for diagnosis codes, 0.5% for medications ordered, 1% for labs/vitals), we include the feature in the final set. For medications, we compare counts at different levels of the RxNorm hierarchy and finally included two levels: ingredients and brands. The final feature set consists of 452 drug ingredients, 2507 drug brands, 49 lab measurements, 5 vital signs, 669 diagnoses, age, race, gender, ethnicity, and smoking status.

Dataset construction. For each patient, we extract data from their entire health record until the time of the first CHD event or censoring time. We aggregate data at the month level; each distinct (patient, month) interaction becomes a single data point. For labs and vital signs, we take the mean value across all observations within the month. For conditions, each month's feature is the aggregate set of all past patient conditions. For medications and diagnoses, we use the set of codes present within a month. We manually map smoking status, race, gender and ethnicity to specific categories (Table F.9 and Table F.10).

To handle missing labs and vitals, we introduce missingness indicators for whether measurements are observed or imputed in that month [190], and include the indicators as features to CHD models. For the imputed value of the missing feature, we use last-one-carry-forward imputation which uses past information as the current information for a lab or vital when it is not measured but previously observed. For any values that remain missing (never measured) we use the population median.

To mitigate the influence of outliers due to potential data entry issues in the measurements, we clip them to standard clinically observable (i.e. possible) ranges [218, 219, 220, 221] (Table F.5).

7.5.2 Models

Model Architecture. The model first creates *embeddings* (numerical representations of categorical features) for drugs, conditions, demographics, and smoking. We use the permutation-invariant transformer, SetTransformer++ [198], to model the relationship between time-to-CHD and the sets of conditions and drugs at each month. Then, the embeddings and remaining features are combined and transformed through a neural network to parameterize a time-to-event categorical distribution [190, 222, 223, 224], where each category represents a small time interval (e.g. several months). This choice makes no assumptions on the shape of the survival distribution because any shape can be approximated by a categorical with enough bins, and the number of bins can be increased when more data becomes available [190, 224].

Model Training. We model the data with survival analysis, which is the standard approach for time-to-event data in the presence of censoring [173]. We assume *right-censoring*, meaning the true time-to-event is greater than the observed time for censored patients. With x_i as features, t_i

as the possibly observed time-to-event, and c_i as the possibly observed censoring time, we define $u_i = \min(t_i, c_i)$ as the observed time and δ_i as an indicator that is 1 if $u_i = t_i$ and 0 if $u_i = c_i$. The observed survival data is $\{x_i, u_i, \delta_i\}_{i=1}^{N}$.

The statistical object of interest is $p(t \mid x)$, the conditional probability mass function of the time-to-event given the input features. We estimate $p(t \mid x)$ using maximum likelihood estimation. Under conditionally independent and noninformative censoring, the likelihood for model p_θ is [225]:

$$L(\theta) = \prod_{i=1}^{N} p_\theta(t_i = u_i \mid x_i)^{\delta_i} S_\theta(u_i \mid x_i)^{(1-\delta_i)}, \tag{7.1}$$

where $S_\theta(t \mid x_i) = 1 - p_\theta(t_i \leq t \mid x_i)$ is acquired by summing the probability mass model p_θ. We use a split of 90% training, 5% evaluation and 5% test data across patients such that the same patient does not appear in multiple splits. Early stopping is used by assessing performance on the evaluation set to prevent overfitting. See Appendix Appendix F.4 for hyperparameter details. We describe the Framingham, Pooled Cohort equations, and PREVENT scores in appendix F.3. The train, evaluation, and test sets for soft and hard CHD contain the same patients. To compute test metrics and evaluation metrics for choosing feature sets, we use a subset of the data formed by randomly dropping all but one visit per patient.

7.5.3 Evaluation

Evaluation Approach. To assess the role of features in transportability we train dozens of models for soft CHD, using 63 different feature grouping combinations, at each hospital. We evaluate all models on the evaluation and test sets of both institutions. We choose the best feature sets (with and without demographics) based the external evaluation concordance, obtaining four feature sets and corresponding soft CHD models. We then train hard CHD models with these four feature sets. These steps produce the four soft and four hard CHD models whose test metrics are in reported in section 7.2. We compute performance under the various metrics in key subpopulations (e.g. Asian females and Black males) to assess whether within-institution performance and transportability differs across subpopulations.

Metrics. We evaluate several discrimination and calibration metrics. For discrimination, we compute concordance using the expected value of $p_\theta(t \mid x_i)$ as a predicted time, and checking the ordering with respect to the true times. This ordering is known for all pairs (i, j) where u_i is uncensored and $u_j > u_i$. The concordance is then the proportion of such correctly ordered comparable pairs:

$$C = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{1}(\mathbb{E}_{p_\theta}[t_i \mid x_i] < \mathbb{E}_{p_\theta}[t_j \mid x_j])\mathbf{1}(u_i < u_j)\delta_i}{\sum_{i=1}^{N} \sum_{j=1}^{N} 1(u_i < u_j)\delta_i} \tag{7.2}$$

We also compute concordance with expected value predictions replaced by modeled Cumulative Distribution Function (CDF) values $F_\theta(t \mid x_i) := p_\theta(t_i \leq t \mid x_i)$ at prediction horizons of 1 year, 5 years, and 10 years, and a weighted version to account for censoring. We chose these horizons because they are key choices over which risk is often estimated for cardiovascular disease [226].

For calibration, we compute inverse-weighted Brier score and binomial log likelihood at 1 year, 5 year and 10 years. These metrics check the overall fit of CDF F_θ, which assesses calibration. To estimate these metrics under censoring, we employ the reweighting technique [214, 216]. which reweights by the probabilities of the censoring time being larger than some time τ: $G(\tau, x_i) := p(c_i > \tau \mid x_i)$. For example, the weighted Brier Score is:

$$BS_{wt}(\tau) = \frac{1}{N} \sum_{i=1}^{N} \frac{(1 - F_\theta(\tau \mid x_i))^2 \delta_i \mathbf{1}(u_i < \tau)}{G(u_i, x_i)} + \frac{F_\theta(\tau \mid x_i)^2 \mathbf{1}(u_i > \tau)}{G(\tau, x_i)}. \tag{7.3}$$

When $1 - G$ equals the censoring CDF, BS_{wt} equals the Brier Score under no censoring. Approximating G requires training time-to-censoring models; this mirrors the training of time-to-event models, but where now the target time is the censoring time, which is censored by observed CHD times. The censoring model uses all features. We use NYULH's censoring model to evaluate all models on NYULH's data, and vice versa for CUIMC. Further details and other metrics are reported in Appendix F.4.

Notation for External Evaluation Definitions We have two institutions **A** and **B** where models may be fit on datasets $\mathcal{D}_A$ and $\mathcal{D}_B$. A conditional model is fit at each institution with parameters θ and ϕ, respectively: $M_A = p^A_\theta(t \mid x)$ and $M_B = p^B_\phi(t \mid x)$. We may define a function $f(M, \mathcal{D})$ which corresponds to a performance assessment of a fit model M evaluated on some data source $\mathcal{D}$. Examples of this function could be a negative log likelihood computation, concordance or other performance metrics.

The Existing Approach to External Validation In the usual setup for clinical external validation, the difference in performance of the same model at two institution is assessed. We define this quantity as $\Delta_{M_A} = f(M_A, \mathcal{D}_A) - f(M_A, \mathcal{D}_B)$. If f is concordance, this amounts to assessing the difference in concordance of a single model trained at institution A across data from both institutions, as shown in fig. 7.6. A limitation of using this evaluation lies in that it assesses the transportability of a model comparing performances on two different populations without any consideration to difficulty of predicting in either population.

The Transport Gap We introduce the T-validation gap (T-gap), which measures the difference between external performance and the known best-achievable performance on the external data, using the same inputs, which we can think of as an empirical upper bound on performance at the external institution. This quantity $\Delta^{tgap}_{\mathcal{D}_A} = f(M_A, \mathcal{D}_A) - f(M_B, \mathcal{D}_A)$ compares two models on a fixed dataset, and is taken as a performance metric for model M_B. This process is shown in Figure 7.6 on the right panel. The T-gap assesses transportability by comparing performances on the same population, thereby correcting for the difficulty of prediction in different populations.

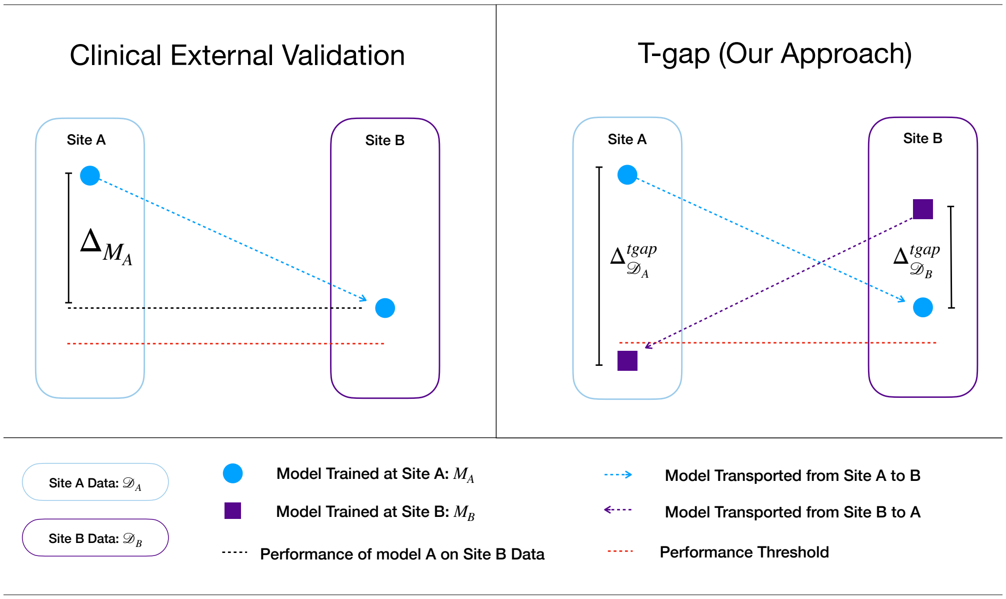

Clinical External Validation
Site A
Site B
Δ_{M_A}
T-gap (Our Approach)
Site A
Site B
$\Delta^{tgap}_{\mathscr{D}_A}$
$\Delta^{tgap}_{\mathscr{D}_B}$
Site A Data: $\mathscr{D}_A$
Site B Data: $\mathscr{D}_B$
Model Trained at Site A: M_A
Model Trained at Site B: M_B
Performance of model A on Site B Data
Model Transported from Site A to B
Model Transported from Site B to A
Performance Threshold

Figure 7.6: Cross-Institutional Validation Enables More Specific Conclusions About Transportability. In T-validation, a CUIMC model (blue) is compared against the NYULH model (purple) at Site B, rather than to the CUIMC model's performance at Site A.

8 | Discussion

The thesis presents a set of techniques and empirical results that enable better healthcare decisions. In part one, we develop an understanding of why models fail to predict well when the distribution shifts between training and testing and propose solutions for building models that generalize OOD with different levels of knowledge about the task. Part two makes it easier to estimate effects on confounded data with IVs and extend causal inference to be able to use confounders specified as functions of variables that we want to intervene on. In part three, we built CHD risk models from minimally curated EHR data that outperformed existing risk scores at the hospital where they were trained and when transported externally. Such models are easy to run on patient information collected in the natural course of a hospital visit to plan preventative treatment. We outline future directions next.

8.1 Future

Flexible and Adaptive OOD generalization. Existing OOD generalization algorithms work for specific kinds of distribution shifts, which may not capture important real-world settings. For example, no single OOD algorithm substantially outperforms empirical risk minimization on all the problems in the WILDS benchmark [227]. Then, it is essential to develop algorithms for OOD generalization that are flexible enough to work for a wide variety of shifts. What if the environment at test time continuously shifts? The model then needs the capability to adaptively improve prediction with a few labeled or unlabeled examples at test time; test-time adaptation algorithms exist [228] but are designed for fixed test distributions. There's a reason to at least conceptually divide the problem into the parts of inferring new shifts and adaptation; the inference and adaptation steps can rely on information beyond the covariates. For example, in classifying chest X-rays or MRIs, hospital information is available during training and for adaptation, *often in clinical process notes*. Despite the conceptual division, a single model could handle multiple kinds of side information by leveraging recent advances in large multi-modal modeling. An important direction here is to adapt to shifts by leveraging various kinds of side information via large multi-modal models with the inference and adaptation steps folded into a single training objective.

Safely leveraging rich EHRs with large AI models. Large EHR systems typically track patients for decades. Modeling from such a rich view of patient trajectories presents an opportunity to improve health outcomes. However, EHRs are not designed for research, and confounding variables that support treatment decisions are often hidden in process or discharge notes. However, as the providers typically know more than what is recorded in the vitals or lab values, collider variables may also be recorded in the notes. Conditioning on such colliders creates a bias in the inferred effects [193] or removes overlap between the treated and control groups. Leveraging clinical notes without introducing additional bias may need new techniques. It would be interesting to leverage techniques from chapter 2 that isolate stable correlations via independence objectives to distill large language model (LLM) representations of clinical notes into the right variables to condition on.

Implicit biases in adapting large models. A foundational issue to building models from large datasets in a trustworthy way is building and adapting large general models for specific clinical tasks. Such adaptation either passes in labeled pairs as part of the prompt to exploit in-context learning [229] or train on targeted data obtained by model-based filtering of a larger dataset [230, 231]. What if the data passed in the prompt contains shortcuts? One direction here is to extend the study from chapter 4 to investigate implicit biases in-context learning when the prompt contains shortcuts. Alternatively, what if the model used to filter data relies on spurious correlations? Turning the independence constraints in chapter 2 to test for spuriousness would be fruitful here. A different direction would be to train the model simultaneously on the filtered data and on augmentations like NR from chapter 3 while weighting the loss of the former samples with the inverse-probability from the latter samples.

A | Appendices for chapter 2

A.1 Further details about NuRD and proofs

A.1.1 Details about NuRD

The algorithm boxes for reweighting-NuRD and generative-NuRD are given in algorithms 1 and 2.

Estimating and using the weights in reweighting-NuRD. In learning $p_{tr}(\mathbf{y} \mid \mathbf{z})$ for a high-dimensional $\mathbf{z}$, flexible models like deep neural networks can have zero training loss when the model memorizes the training data. For a discrete $\mathbf{y}$, such a model would output $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z}) = 1$ for every sample in the training data. Then, the model's weight estimates on the training data are $p(\mathbf{y})/\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z}) = p(\mathbf{y})$. Weighting the training data with such estimates fails to break the nuisance-label relationship because $p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x})\frac{p(\mathbf{y})}{\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})} = p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x})p(\mathbf{y}) \propto p_{tr}(\mathbf{z} \mid \mathbf{y})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$. To avoid such poor weight estimation, we employ a cross-fitting procedure: split the data into K disjoint folds, and the weights for each fold are produced by a model trained and validated on the rest of the folds. See algorithm 1 for details. In estimating loss for each batch under $\hat{p}_{\perp\!\!\!\perp}$ during training, one can either weight the per-sample loss or produce the batches themselves via weighted sampling from the data with replacement.

Density-ratio trick in Distillation. The density-ratio trick for estimating mutual information [11] involves Monte Carlo estimating the mutual information using a binary classifier. Let $\ell = 1$ be the pseudolabel for samples from $p_{\perp\!\!\!\perp}(\mathbf{y}, r_\gamma(\mathbf{x}), \mathbf{z})$, and $\ell = 0$ for samples from $p_{\perp\!\!\!\perp}(\mathbf{y}, r_\gamma(\mathbf{x}))p_{\perp\!\!\!\perp}(\mathbf{z})$. Then,

$$\mathbf{I}_{\hat{p}_{\perp\!\!\!\perp}}([r_\gamma(\mathbf{x}), \mathbf{y}]; \mathbf{z}) = \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})} \log \frac{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, r_\gamma(\mathbf{x}), \mathbf{z})}{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, r_\gamma(\mathbf{x}))\hat{p}_{\perp\!\!\!\perp}(\mathbf{z})} = \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})} \log \frac{p(\ell = 1 \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))}{1 - p(\ell = 1 \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))}.$$

With parameters ϕ, we estimate the conditional probability with a *critic model*, denoted p_ϕ.

Accounting for shifts in the marginal label distribution. NuRD relies on the assumption in eq. (2.1) that distributions in the nuisance-varying family $\mathcal{F}$ have the same marginal $p(\mathbf{y})$. What happens if p_{te} comes from a nuisance-varying family with a different marginal? Formally, with $p_{tr} \in \mathcal{F}$, let p_{te} belong to a nuisance-varying family $\mathcal{F}' = \{{}^{p_{te}(\mathbf{y})}/_{p_{tr}(\mathbf{y})} p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p_{te}(\mathbf{y})p_D(\mathbf{z} \mid \mathbf{y})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})\}$ where $p_D \in \mathcal{F}$. Given knowledge of the marginal distribution $p_{te}(\mathbf{y})$, note that the weighted training distribution $p_{tr}' = {}^{p_{te}(\mathbf{y})}/_{p_{tr}(\mathbf{y})} p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x})$ lives in $\mathcal{F}'$. Running NuRD on p_{tr}' produces predictive models that generalize to p_{te}. To see this, note

$$p'_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p'_{tr}(\mathbf{y})p'_{tr}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$$

is a nuisance-randomized distribution in $\mathcal{F}'$. With $\mathcal{R}(p'_{\perp\!\!\!\perp})$ as the uncorrelating set of representations defined with respect to $p'_{\perp\!\!\!\perp}$, i.e. $r'(\mathbf{x}) \in \mathcal{R}(p'_{\perp\!\!\!\perp}) \implies \mathbf{y} \perp\!\!\!\perp_{p'_{\perp\!\!\!\perp}} \mathbf{z} \mid r'$, lemma 1 and theorem 1 hold. It follows that running NuRD on samples from ${}^{p_{te}(\mathbf{y})}/_{p_{tr}(\mathbf{y})} p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x})$ produces an estimate of $p'_{\perp\!\!\!\perp}(\mathbf{y} \mid r'(\mathbf{x}))$ $(r'(\mathbf{x}) \in \mathcal{R}(p'_{\perp\!\!\!\perp}))$ with the maximal performance on every $p_{te} \in \mathcal{F}'$ if a maximally blocking $r^*(\mathbf{x}) \in \mathcal{R}(p'_{\perp\!\!\!\perp})$.

A.1.2 Extended related work

Domain generalization methods aim to build models with the goal of generalizing to unseen test data different from the training data [3]. Recent work uses multiple *sufficiently different* environments to generalize to unseen test data that lies in the support of the given environments or subgroups [4, 12, 13, 19, 20, 21, 22, 23, 24]. Chang et al. [232] develop a multi-environment objective to interpret neural network predictions that are robust to spurious correlations. Similarly, domain-invariant learning and related methods build representations that are independent of the domain [25, 26, 27, 28, 29, 233].

Due to its focus on nuisances, NuRD works with data from a single environment. As in section 2.4, to split the data into multiple environments, one can split the data into groups based on the value of the nuisance. Then, domain-invariant methods build representations that are independent of the nuisance and under nuisance-induced spurious correlations these representations may ignore semantic features because they are correlated with the nuisance. Domain adaptation [234, 235, 236] methods assume access to unlabelled test data which NuRD does not require. We do not assume access to the test data because nuisance-label relationships can change over time or geography which, in turn, changes the the target distribution.

Taking a distributional robustness [31] approach, Sagawa et al. [16] applied group-DRO to training data where the relative size of certain groups in the training data results in spurious correlations. Given these groups, group-DRO optimizes the worst error across distributions formed by weighted combinations of the groups. With high dimensional $\mathbf{z}$ as in our experiments, defining groups based on the value of the nuisance typically results in groups with at most one sample; with such groups, group-DRO will encourage memorizing the training data. Other work aims to minimize worst subgroup error with a finite number of fixed but unknown subgroups [32, 33]; as subgroups are unknown, they only find an approximate minimizer of the worst subgroup error in general even with infinite data.

Algorithm 1: Reweighting-NuRD

```
Input: Training data $D$, specification of the weight model $p_\alpha(\mathbf{y} \mid \mathbf{z})$ which estimates $p_{tr}(\mathbf{y} \mid \mathbf{z})$, representation model $r_\gamma(\mathbf{x})$, predictive model $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ and critic model
    $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$; regularization coefficient $\lambda$, number of iterations for the weight model $N_w$, for the predictive model and representation $N_p$, and the number of critic model
    steps $N_c$. Number of folds $K$.
Result: Return estimate of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ for $r_\gamma \in \mathcal{R}(p_{\perp\!\!\!\perp})$ with maximal information with $\mathbf{y}$.
Nuisance Randomization step;
Estimate the marginal distribution over the label $\hat{p}(\mathbf{y})$;
Split data into $K$ equal disjoint folds, $D = \{F_i\}_{i \leq K}$, for cross-fitting;
for each fold $F_i$, $i \leq K$ do // (cross-fitting)
    Initialize $p_\alpha(\mathbf{y} \mid \mathbf{z})$;
    for $N_w$ iterations do
        Sample training batch from the rest of the folds $(F_{-i}) : B \sim F_{-i}$;
        Compute likelihood $\sum_{(\mathbf{y}_i, \mathbf{z}_i) \in B} \log p_\alpha(\mathbf{y}_i \mid \mathbf{z}_i)$;
        Update $\alpha$ to maximize this likelihood (via Adam for example);
    end
    Produce weights $w_i = \hat{p}(\mathbf{y}_i)/p_\alpha(\mathbf{y}_i \mid \mathbf{z}_i)$ for each $(\mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i) \in F_i$;
end
Distillation step;
Initialize $r_\gamma, p_\theta, p_\phi$;
for $N_p$ iterations do
    for $N_c$ iterations do
        Sample training batch $B \sim D$ and sample independent copies of $\mathbf{z}$ marginally: $\tilde{\mathbf{z}}_i \sim D$;
        Construct batch $\tilde{B} = \{\mathbf{y}_i, \tilde{\mathbf{z}}_i, \mathbf{x}_i\}$;
        Compute likelihood
            $$\sum_{(\mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i) \in B} w_i \log p_\phi(\ell = 1 \mid \mathbf{y}_i, \mathbf{z}_i, r_\gamma(\mathbf{x}_i)) + \sum_{(\mathbf{y}_i, \tilde{\mathbf{z}}_i, \mathbf{x}_i) \in \tilde{B}} w_i \log p_\phi(\ell = 0 \mid \mathbf{y}_i, \tilde{\mathbf{z}}_i, r_\gamma(\mathbf{x}_i)).$$
        Update $\phi$ to maximize likelihood (via Adam for example);
    end
    Sample training batch $B \sim D$;
    Compute distillation objective using the density-ratio trick
        $$\frac{1}{|B|} \sum_{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i) \in B} w_i \left[ \log p_\theta(\mathbf{y}_i \mid r_\gamma(\mathbf{x}_i)) - \lambda \log \frac{p_\phi(\ell = 1 \mid \mathbf{y}_i, \mathbf{z}_i, r_\gamma(\mathbf{x}_i))}{1 - p_\phi(\ell = 1 \mid \mathbf{y}_i, \mathbf{z}_i, r_\gamma(\mathbf{x}_i))} \right].$$
    Update $\theta, \gamma$ to maximize objective (via Adam for example).
end
Return $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$.
```

In contrast, NuRD builds predictive models with performance guarantees across all test distributions (that factorize as eq. (2.1)) using knowledge of the nuisance. Given the nuisance, existence of a finite number of subgroups maps to an additional discreteness assumption on the nuisance variable; NuRD works with general high-dimensional nuisances. Wang and Culotta [237] focus on sentiment analysis of reviews and build a dataset where the nuisance label relationship is destroyed by swapping words known to be associated with sentiment of the review, with their antonyms. This is equivalent to using domain-specific knowledge to sample from $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ in generative NuRD. NuRD requires no domain-specific knowledge about the generative model $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$.

Algorithm 2: Generative-NuRD

```
Input: Training data $D$, specification of the generative model $p_\beta(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ that estimates $p_{tr}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$, representation model $r_\gamma(\mathbf{x})$, predictive model $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$, and critic model
    $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$; regularization coefficient $\lambda$, number of iterations for the weight model $N_w$, number of iterations for the predictive model and representation $N_p$, number
    of critic steps $N_c$.
Result: Return estimate of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ for $r_\gamma \in \mathcal{R}(p_{\perp\!\!\!\perp})$ with maximal information with $\mathbf{y}$.
Nuisance Randomization step;
for $N_w$ iterations do
    Sample training batch $B \sim D$;
    Compute likelihood $\sum_{(\mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i) \in B} \log p_\beta(\mathbf{x}_i \mid \mathbf{z}_i, \mathbf{y}_i)$ (or some generative objective);
    Update $\beta$ to maximize objective above;
end
Estimate the marginal distribution over the label $\hat{p}(\mathbf{y})$;
Sample independent label and nuisance $\mathbf{y}_i \sim D, \mathbf{z}_j \sim D$, and then sample $\tilde{\mathbf{x}} \sim p_\beta(\mathbf{x} \mid \mathbf{y}_i, \mathbf{z}_j)$;
Construct dataset $\hat{D}$ using triples $\{\mathbf{y}_k = \mathbf{y}_i, \mathbf{z}_k = \mathbf{z}_j, \mathbf{x}_k = \hat{\mathbf{x}}\}$;
Distillation step;
Initialize $r_\gamma, p_\theta, p_\phi$;
for $N_p$ iterations do
    for $N_c$ iterations do
        Sample training batch $B \sim D$ and sample independent copies of $\mathbf{z}$ marginally: $\tilde{\mathbf{z}_i} \sim D$;
        Construct batch $\tilde{B} = \{\mathbf{y}_i, \tilde{\mathbf{z}_i}, \mathbf{x}_i\}$ using $B$;
        Compute likelihood
            $$\sum_{(\mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i) \in B} w_i \log p_\phi(\ell = 1 \mid \mathbf{y}_i, \mathbf{z}_i, r_\gamma(\mathbf{x}_i)) + \sum_{(\mathbf{y}_i, \tilde{\mathbf{z}_i}, \mathbf{x}_i) \in \tilde{B}} w_i \log p_\phi(\ell = 0 \mid \mathbf{y}_i, \tilde{\mathbf{z}_i}, r_\gamma(\mathbf{x}_i)).$$
        Update $\phi$ to maximize likelihood (via Adam for example);
    end
    Sample batch from generated training data $B \sim \tilde{D}$;
    Compute distillation objective using the density-ratio trick
        $$\frac{1}{|B|} \sum_{(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k) \in B} \left[ \log p_\theta(\mathbf{y}_k \mid r_\gamma(\mathbf{x}_k)) - \lambda \log \frac{p_\phi(\ell = 1 \mid \mathbf{y}_k, \mathbf{z}_k, r_\gamma(\mathbf{x}_k))}{1 - p_\phi(\ell = 1 \mid \mathbf{y}_k, \mathbf{z}_k, r_\gamma(\mathbf{x}_k))} \right];$$
    Update $\theta, \gamma$ to maximize objective (via Adam for example).
end
Return $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$.
```

A.1.3 Key lemmas for uncorrelating representations

In this lemma, we derive the performance of the nuisance-randomized conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ for any $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$ and show that it is at least as good as predicting without covariates on any $p_{te} \in \mathcal{F}$.

Lemma 1. *Let $\mathcal{F}$ be a nuisance-varying family (eq. (2.1)) and $p_{\perp\!\!\!\perp} = p(\mathbf{y})p_{tr}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ for some $p_{tr} \in \mathcal{F}$. Assume $\forall p_D \in \mathcal{F}$, $p_D(\mathbf{z} \mid \mathbf{y})$ is bounded. If $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$, then $\forall p_{te} \in \mathcal{F}$, the performance of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ is*

$$\texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) = \texttt{Perf}_{p_{te}}(p(\mathbf{y})) + \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} KL\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \,\|\, \mathbb{E}_{p(\mathbf{y})}p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right]. \qquad (A.1)$$

As the KL-divergence is non-negative, $\texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) \geq \texttt{Perf}_{p_{te}}(p(\mathbf{y}))$.

Proof. (of lemma 1) Note that the identity $\mathbb{E}_{p(\mathbf{x})} g \circ f(\mathbf{x}) = \mathbb{E}_{p(f(\mathbf{x}))} g \circ f(\mathbf{x})$ implies that

$$\mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} = \mathbb{E}_{p_{te}(\mathbf{y},r(\mathbf{x}))} \log \frac{p_{te}(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))}.$$

As $p_{\perp\!\!\!\perp}(\mathbf{z} \mid \mathbf{y}) = p_{tr}(\mathbf{z}) > 0$ on $\mathbf{z} \in S_{\mathcal{F}}$ and $\mathbf{y}$ s.t. $p(\mathbf{y}) > 0$ is bounded, lemma 3 implies that $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \Leftrightarrow p_{te}(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0$. This fact implies the following **KL** terms and expectations of log-ratios are all well-defined:

$$\begin{aligned}
-\texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) &= \mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \,\|\, p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] \\
&= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y} \mid \mathbf{x})p_{te}(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))p_{te}(\mathbf{y})} \\
&= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y} \mid \mathbf{x})}{p_{te}(\mathbf{y})} + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \\
&= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y} \mid \mathbf{x})}{p_{te}(\mathbf{y})} + \mathbb{E}_{p_{te}(\mathbf{y},r(\mathbf{x}))} \log \frac{p(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \,\|\, p_{\perp\!\!\!\perp}(\mathbf{y})\right] + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))} \log \frac{p(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))}
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\mathbb{E}_{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y},\mathbf{z})} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\mathbb{E}_{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y},\mathbf{z})} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z})} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\mathbb{E}_{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y},\mathbf{z})} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{z})p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y}, r(\mathbf{x}) \mid \mathbf{z})} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{y},\mathbf{z})} \log \frac{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z})}{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] - \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\mathbb{E}_{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y},\mathbf{z})} \log \frac{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z})} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] - \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\text{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z})\right]
\end{aligned}
$$

Here, $p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) = p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})$ as $p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ by definition of the nuisance-varying family. The proof follows by noting that the gap in performance of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ and $p(\mathbf{y})$ equals an expected **KL** term:

$$
\begin{aligned}
-\texttt{Perf}_{p_{te}}(p(\mathbf{y})) + \texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) &= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\text{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z})\right] \\
&= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})}\text{KL}\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})}p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right].
\end{aligned}
\tag{A.2}
$$

Rearranging these terms completes the proof. □

Lemma 2 shows that uncorrelating sets are the same for any nuisance-randomized distribution and that the conditional distribution of the label given an uncorrelating representations is the same for all nuisance-randomized distributions.

Lemma 2. *Let $\mathcal{F}$ be a nuisance-varying family with $p(\mathbf{y})$ and $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ and nuisance space $S_{\mathcal{F}}$. Consider distributions $p_{\perp\!\!\!\perp,1}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y})p_{\perp\!\!\!\perp,1}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ and $p_{\perp\!\!\!\perp,2}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y})p_{\perp\!\!\!\perp,2}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ such that $p_{\perp\!\!\!\perp,1}(\mathbf{z}) > 0, p_{\perp\!\!\!\perp,2}(\mathbf{z}) > 0$ for $\mathbf{z} \in S_{\mathcal{F}}$, and $p_{\perp\!\!\!\perp,1}(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \iff p_{\perp\!\!\!\perp,2}(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0$. Then, the* *uncorrelating sets are equal* $\mathcal{R}(p_{\perp\!\!\!\perp,1}) = \mathcal{R}(p_{\perp\!\!\!\perp,2})$ *and for any* $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp,1})$*,*

$$p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) = p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x})).$$

Proof. By the assumption that $p_{\perp\!\!\!\perp,1}(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \Leftrightarrow p_{\perp\!\!\!\perp,2}(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0$, there exist some $\mathbf{z}$ such that $p_{\perp\!\!\!\perp,1}(\mathbf{z} \mid r(\mathbf{x})) > 0$ and $p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x})) > 0$. With such $\mathbf{z}$, for any $r \in \mathcal{R}(p_{\perp\!\!\!\perp,1})$,

$$\begin{aligned}
p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) &= p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) \\
&= p(\mathbf{y}) \frac{p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{p_{\perp\!\!\!\perp,1}(r(\mathbf{x}) \mid \mathbf{z})} \\
&= p(\mathbf{y}) \frac{p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{\mathbb{E}_{p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid \mathbf{z})}[p_{\perp\!\!\!\perp,1}(r(\mathbf{x}) \mid \mathbf{z}, \mathbf{y})]} \\
&= p(\mathbf{y}) \frac{p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{\mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{z}, \mathbf{y})} \\
&= p(\mathbf{y}) \frac{p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{\mathbb{E}_{p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid \mathbf{z})} p(r(\mathbf{x}) \mid \mathbf{z}, \mathbf{y})} \\
&= p(\mathbf{y}) \frac{p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{p_{\perp\!\!\!\perp,2}(r(\mathbf{x}) \mid \mathbf{z})} \\
&= p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z})
\end{aligned}$$

Taking expectation on both sides with respect to $p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x}))$,

$$\mathbb{E}_{p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x}))} p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) = \mathbb{E}_{p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x}))} p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) = p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x})). \tag{A.3}$$

Note that $\mathbb{E}_{p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x}))} p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) = p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x}))$, which implies

$$p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) = p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) = p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) = p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x})),$$

completing one part of the proof, $p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) = p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x}))$.

Further, we showed $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp,1}} \mathbf{z} \mid r(\mathbf{x}) \implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp,2}} \mathbf{z} \mid r(\mathbf{x})$ which means $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp,2})$. As the above proof holds with $p_{\perp\!\!\!\perp,1}, p_{\perp\!\!\!\perp,2}$ swapped with each other, $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp,1}) \Longleftrightarrow r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp,2})$.

□

The next lemma shows that every member of the nuisance-varying family is positive over the same set of $\mathbf{y}, \mathbf{z}, \mathbf{x}$ and is used in proposition 1 and lemma 1.

Lemma 3. *Let the nuisance-varying family $\mathcal{F}$ be defined with $p(\mathbf{y}), p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ and nuisance space $S_{\mathcal{F}}$. Let distributions $p_D = p(\mathbf{y})p_D(\mathbf{z} \mid \mathbf{y})p(\mathbf{x} \mid \mathbf{z}, \mathbf{y})$ and $p'_D = p(\mathbf{y})p'_D(\mathbf{z} \mid \mathbf{y})p(\mathbf{x} \mid \mathbf{z}, \mathbf{y})$ be such that $p_D(\mathbf{z} \mid \mathbf{y}), p'_D(\mathbf{z} \mid \mathbf{y}) > 0$ for all $\mathbf{y}$ such that $p(\mathbf{y}) > 0$ and $\mathbf{z} \in S_{\mathcal{F}}$. Further assume $p_D(\mathbf{z} \mid \mathbf{y}), p'_D(\mathbf{z} \mid \mathbf{y})$ are bounded. Then, $p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \Leftrightarrow p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0$.*

Proof. For any $\mathbf{z} \in S_{\mathcal{F}}$ and any $\mathbf{y}$ such that $p(\mathbf{y}) > 0$, $p_D(\mathbf{z} \mid \mathbf{y}) > 0$ and $\frac{p'_D(\mathbf{z} \mid \mathbf{y})}{p_D(\mathbf{z} \mid \mathbf{y})} > 0$,

$$p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})p'_D(\mathbf{z} \mid \mathbf{y})p(\mathbf{y}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})p_D(\mathbf{z} \mid \mathbf{y})p(\mathbf{y})\frac{p'_D(\mathbf{z} \mid \mathbf{y})}{p_D(\mathbf{z} \mid \mathbf{y})} = p_D(\mathbf{y}, \mathbf{z}, \mathbf{x})\frac{p'_D(\mathbf{z} \mid \mathbf{y})}{p_D(\mathbf{z} \mid \mathbf{y})}. \tag{A.4}$$

Thus, for all $\mathbf{z}$ in the nuisance space $S_{\mathcal{F}}$ and any $\mathbf{y}$ such that $p(\mathbf{y}) > 0$,

$$p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \Longleftrightarrow p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0.$$

As $\mathbf{z}$ only takes values in the nuisance space $S_{\mathcal{F}}$, when $p(\mathbf{y}) = 0$,

$$p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = 0.$$

Together, the two statements above imply

$$p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \Leftrightarrow p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0.$$

□

A.1.4 Optimal uncorrelating representations

Theorem 1. *Let $r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})$ be **maximally blocking**:*

$$\forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r^*(\mathbf{x}).$$

Then,

1. *(Simultaneous optimality)* $\forall p_{te} \in \mathcal{F}, \forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \mathtt{Perf}_{p_{te}}(r^*(\mathbf{x})) \geq \mathtt{Perf}_{p_{te}}(r(\mathbf{x}))$.

2. *(Information maximality)* $\forall r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp}), \quad \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) \geq \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r(\mathbf{x}))$.

3. *(Information maximality implies simultaneous optimality)* $\forall r' \in \mathcal{R}(p_{\perp\!\!\!\perp})$,

$$\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r'(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) \quad \Longrightarrow \quad \forall p_{te} \in \mathcal{F}, \quad \mathtt{Perf}_{p_{te}}(r^*(\mathbf{x})) = \mathtt{Perf}_{p_{te}}(r'(\mathbf{x})).$$

Proof. (proof for theorem 1)

We first prove that for any pair $r, r_2 \in \mathcal{R}(p_{\perp\!\!\!\perp})$ such that r_2 blocks r, $r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid \mathbf{z}, r_2(\mathbf{x})$, r_2 dominates the performance of r on every $p_{te} \in \mathcal{F}$. The simultaneously optimality of the maximally blocking representation will follow. For readability, let $\ell(r_2) = \mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))\right]$. We will show that

$$\mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] \geq \ell(r_2).$$

We will use the following identity which follows from the fact that $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ does not change between distributions in the data generating process eq. (2.1):

$$\begin{aligned} p_D(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) &= \frac{p_D(r_2(\mathbf{x}), r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{p_D(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})} \\ &= \frac{p(r_2(\mathbf{x}), r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})} \end{aligned}$$

$$= p(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})).$$

Next, we will show that

$$\begin{aligned}
&\mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] \\
&\qquad = \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))}\mathrm{KL}\left[p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})\right].
\end{aligned}$$

The steps are similar to lemma 1's proof

$$\begin{aligned}
\mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] &= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y} \mid \mathbf{x})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))} + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z})} \qquad \{\text{as } r, r_2 \in \mathcal{R}(p_{\perp\!\!\!\perp})\} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}), r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z})} \qquad \{\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r_2(\mathbf{x})\} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y}, r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))}\mathbb{E}_{p_{te}(r_2(\mathbf{x}) \mid \mathbf{y},\mathbf{z},r(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})} \\
&= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))}\mathrm{KL}\left[p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})\right]
\end{aligned}$$

Noting that **KL** is non-negative and that `Perf` is negative-**KL** proves the theorem:

$$\mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))\right] \leq \mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right]. \qquad \text{(A.5)}$$

It follows that for a maximally blocking r^*

$$\forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}) \quad \mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))\right] \leq \mathbb{E}_{p_{te}(\mathbf{x})}\mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right].$$

As performance is negative **KL**, the proof follows that r^* dominates r in performance. This concludes the first part of the proof.

For the second part, we prove information maximality of a maximally blocking $r^*(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$. The proof above shows that the model $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))$ performs at least as well as $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ for any $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$ on any $p_{te} \in \mathcal{F}$. We characterize the gap in performance between $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))$ and $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ for any $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$ as the following conditional mutual information term:

$$\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},r(\mathbf{x}))}\mathrm{KL}\left[p_{\perp\!\!\!\perp}(r^*(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(r^*(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})\right] = \mathbf{I}_{p_{\perp\!\!\!\perp}}(r^*(\mathbf{x}); \mathbf{y} \mid \mathbf{z}, r(\mathbf{x})).$$

The entropy decomposition of conditional mutual information (with $\mathbf{H}_q(\cdot)$ as the entropy under a distribution q) gives two mutual information terms.

$$\begin{aligned}
\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},r(\mathbf{x}))}\mathrm{KL}\left[p_{\perp\!\!\!\perp}(r^*(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(r^*(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})\right] &= \mathbf{I}_{p_{\perp\!\!\!\perp}}(r^*(\mathbf{x}); \mathbf{y} \mid \mathbf{z}, r(\mathbf{x})), \\
&= \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid \mathbf{z}, r(\mathbf{x})) - \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid \mathbf{z}, r(\mathbf{x}), r^*(\mathbf{x})) \\
&= \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid \mathbf{z}, r(\mathbf{x})) - \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid \mathbf{z}, r^*(\mathbf{x})) \qquad \{\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r^*(\mathbf{x})\} \\
&= \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid r(\mathbf{x})) - \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid r^*(\mathbf{x})) \qquad \{r, r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})\} \\
&= \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid r(\mathbf{x})) - \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y}) + \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y}) - \mathbf{H}_{p_{\perp\!\!\!\perp}}(\mathbf{y} \mid r^*(\mathbf{x})) \\
&= \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) - \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}, r(\mathbf{x})).
\end{aligned}$$

This difference is non-negative for any $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$ which proves the second part of the theorem:

$$\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) - \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}, r(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(r^*(\mathbf{x}); \mathbf{y} \mid \mathbf{z}, r(\mathbf{x})) \geq 0.$$

For the third part, note that any representation r' which satisfies $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}, r'(\mathbf{x}))$ (information-equivalence) also satisfies

$$\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x}) \mid \mathbf{z}, r'(\mathbf{x})) = 0 \implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r^*(\mathbf{x}) \mid \mathbf{z}, r'(\mathbf{x}).$$

Under this condition, eq. (A.5) implies

$$\mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))\right] \geq \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r'(\mathbf{x}))\right].$$

However, as $r' \in \mathcal{R}(p_{\perp\!\!\!\perp})$ and that $r^*(\mathbf{x})$ is maximally blocking, which (by the proof above) implies

$$\mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))\right] \leq \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r'(\mathbf{x}))\right].$$

The only way both these conditions hold is if

$$\mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))\right] = \mathbb{E}_{p_{te}(\mathbf{x})}\text{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r'(\mathbf{x}))\right].$$

This completes the proof that for any $r' \in \mathcal{R}(p_{\perp\!\!\!\perp})$ that is information-equivalent to $r^*(\mathbf{x})$ under $p_{\perp\!\!\!\perp}$, the model $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r'(\mathbf{x}))$ has the same performance as $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r^*(\mathbf{x}))$ for every $p_{te} \in \mathcal{F}$, and consequently, r' is also optimal.

□

A.1.5 Minimax optimality

Proposition 1. *Consider a nuisance-varying family $\mathcal{F}$ (eq. (2.1)) such that for some $p_{tr} \in \mathcal{F}$ there exists a distribution $p_{\perp\!\!\!\perp} \in \mathcal{F}$ such that $p_{\perp\!\!\!\perp} = p(\mathbf{y})p_{tr}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \in \mathcal{F}$. Let $\mathcal{F}$ satisfy*

$$\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z} \implies \exists p'_D \in \mathcal{F} \, s.t. \, \left[\mathbb{E}_{p'_D(\mathbf{x})} KL\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y})\right] > 0. \tag{A.6}$$

If $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{x}$, then $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ is minimax optimal :

$$p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}) = \underset{p_D(\mathbf{y} \mid \mathbf{x}); p_D \in \mathcal{F}}{\arg\min} \max_{p'_D \in \mathcal{F}} \mathbb{E}_{p'_D(\mathbf{x})} KL\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right].$$

Proof. (of proposition 1) By lemma 3, as $p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0 \Leftrightarrow p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) > 0$, performance is well defined for any $p_D(\mathbf{y} \mid \mathbf{x})$ on any $p'_D \in \mathcal{F}$. First, lemma 1 with $p_{te} = p'_D$ and $r(\mathbf{x}) = \mathbf{x}$ gives

$$\begin{aligned}
&\mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y}) - \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] \\
&\qquad = \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] - \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] \\
&\qquad = \mathbb{E}_{p'_D(\mathbf{y}, \mathbf{z})} \mathrm{KL}\left[p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})\right]. \\
&\qquad = \mathbb{E}_{p'_D(\mathbf{y}, \mathbf{z})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \parallel p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{z})\right] \geq 0.
\end{aligned} \tag{A.7}$$

Thus, unlike any $p_D \in \mathcal{F}$ such that $\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z}$,

$$\max_{p'_D \in \mathcal{F}} \left[\mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y})\right] \leq 0. \tag{A.8}$$

For any p_D such that $\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z}$, let p'_D be such that $\mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y}) > 0$. As eq. (A.8) implies $\mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y}) \leq 0$, it follows that $\forall p_D$ such that

$\mathbf{y} \perp\!\!\!\perp_{p_D} \mathbf{z}$,

$$\max_{p'_D \in \mathcal{F}} \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right] > \max_{p'_D \in \mathcal{F}} \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right].$$

By lemma 3, any $p_D \in \mathcal{F}$ is positive over the same set of $\mathbf{y}, \mathbf{z}, \mathbf{x}$ and if $\mathbf{y} \perp\!\!\!\perp_{p_D} \mathbf{z}$, then $p_D(\mathbf{y} \mid \mathbf{x}) = p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ (see lemma 2 for proof with instantiation $r(\mathbf{x}) = \mathbf{x}$). This means

$$p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}) = \arg\min_{p_D \in \mathcal{F}} \max_{p'_D \in \mathcal{F}} \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right].$$

□

See proposition 3 for an example nuisance-varying family where the information criterion in eq. (A.6) holds.

A.1.6 Distillation details and a local optima example for eq. (2.5)

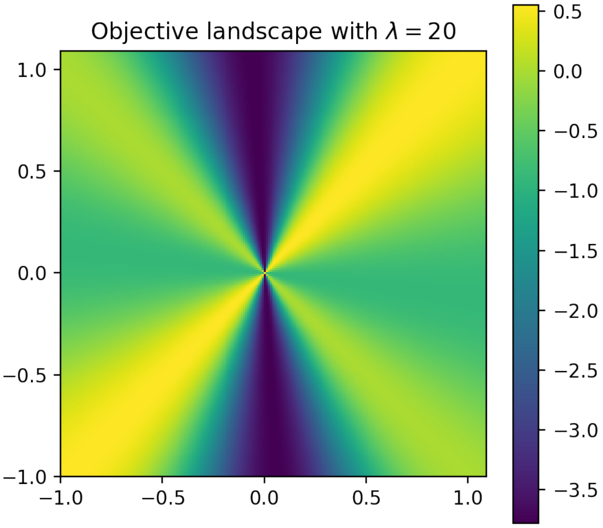


Figure A.1: Landscape of the objective in eq. (2.5) for the example in eq. (2.2) for linear representations $r_{u,v}(\mathbf{x}) = u\mathbf{x}_1 + v\mathbf{x}_2$. Local maxima correspond to representations $r_{-u,u}$ and global maxima to representations $r_{u,u}$.

The objective in eq. (2.5) can have local optima when the representation is a function of the nuisance and the exogenous noise in the generation of the covariates given the nuisance and the label. Formally, let the exogenous noise $\boldsymbol{\epsilon}$ satisfy $(\boldsymbol{\epsilon}, \mathbf{z}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y}$. Then,

$$(\boldsymbol{\epsilon}, \mathbf{z}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \implies (f(\boldsymbol{\epsilon}, \mathbf{z}), \mathbf{z}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \implies \mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid f(\mathbf{z}, \boldsymbol{\epsilon}).$$

Such a representation $r(\mathbf{x}) = f(\boldsymbol{\epsilon}, \mathbf{z})$ is both in the uncorrelating set and independent of the label $\mathbf{y}$ under the nuisance-randomized distribution $p_{\perp\!\!\!\perp}$ meaning it does not predict the label.

Local optima example for conditional information regularization eq. (2.5). Figure A.1 plots the value of the objective in eq. (2.5) computed analytically for $\lambda = 20$, over the class of linear representations indexed by $u, v \in \mathbf{R}$, $r_{u,v}(\mathbf{x}) = u\mathbf{x}_1 + v\mathbf{x}_2$, under the data generating process in eq. (2.2). Representations of the kind $r_{-u,u}(\mathbf{x}) = u(\mathbf{x}_2 - \mathbf{x}_1)$ are functions of $\mathbf{z}$ and some noise independent of the label and, as fig. A.1 shows, are local maxima on the landscape of the maximization objective in eq. (2.5). Global maxima correspond to representations $r_{u,u}$.

Performance characterization for jointly independent representations

Lemma 4. *Let $\mathcal{F}$ be a nuisance varying family. For any jointly independent representation r, i.e. $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$,*

$$\forall p_{te} \in \mathcal{F} \qquad \texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) = C_{p_{te}} + \mathbf{I}_{p_{\perp\!\!\!\perp}}(r(\mathbf{x}); \mathbf{y}),$$

where $C_{p_{te}}$ is a p_{te}-dependent constant that does not vary with $r(\mathbf{x})$.

NuRD maximizes the information term $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r_\gamma(\mathbf{x}))$ and, therefore, maximizes performance on every member of $\mathcal{F}$ simultaneously. It follows that within the set of jointly independent representations, NuRD, at optimality, produces a representation that is simultaneously optimal on every $p_{te} \in \mathcal{F}$.

Proof. Lemma 1 says that for any uncorrelating representation $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$ and $\forall p_{te} \in \mathcal{F}$,

$$\texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) = \texttt{Perf}_{p_{te}}(p(\mathbf{y})) + \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right].$$

However, as the joint independence $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ implies both the uncorrelating property and $r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{y}$, the second term in the RHS above can be expressed as $\mathbf{I}_{p_{\perp\!\!\!\perp}}(r(\mathbf{x}); \mathbf{y})$:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right] \\
&\quad = \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y})} p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right] \\
&\quad = \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}) \parallel \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y})} p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y})\right] \\
&\quad = \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}) \parallel p_{\perp\!\!\!\perp}(r(\mathbf{x}))\right] \\
&\quad = \mathop{\mathbb{E}}_{p_{te}(\mathbf{y})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}) \parallel p_{\perp\!\!\!\perp}(r(\mathbf{x}))\right] \\
&\quad = \mathop{\mathbb{E}}_{p_{\perp\!\!\!\perp}(\mathbf{y})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}) \parallel p_{\perp\!\!\!\perp}(r(\mathbf{x}))\right] \\
&\quad = \mathbf{I}_{p_{\perp\!\!\!\perp}}(r(\mathbf{x}); \mathbf{y}).
\end{aligned}$$

Noting $C_{p_{te}} = \texttt{Perf}_{p_{te}}(p(\mathbf{y}))$ does not vary with $r(\mathbf{x})$ completes the proof. □

Performance gaps between jointly independent representations and uncorrelating representations. The joint independence $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ implies the uncorrelating property but uncorrelating representations only satisfy this joint independence when they are independent of the nuisance. Thus, representations that satisfy joint independence form a subset of uncorrelating representations. This begs a question: is there a loss in performance by restricting NuRD to representations that satisfy said joint independence? In appendix A.1.7.1, we use the theory of minimal sufficient statistics [37] to show that there exists a nuisance-varying family where the best uncorrelating representation dominates every representation that satisfies joint independence on every member distribution, and is strictly better in at least one.

A.1.7 Counterfactual invariance vs. the uncorrelating property

We A) show that counterfactually invariant representations are a subset of uncorrelating representations by reducing counterfactual invariance to the joint independence $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ and B) give an example nuisance-varying family $\mathcal{F}$ where the best uncorrelating representation strictly dominates every jointly independent representation in performance on every test distribution $p_{te} \in \mathcal{F}$: at least as good on all $p_{te} \in \mathcal{F}$ and strictly better on at least one.

We show A by proving counterfactually invariant representations satisfy joint independence $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ which implies the uncorrelating property, but not vice versa. Counterfactual invariance implies that for all $p_D \in \mathcal{F}$, the conditional independence $r(\mathbf{x}) \perp\!\!\!\perp_{p_D} \mathbf{z} \mid \mathbf{y}$ holds by theorem 3.2 in [17]. As $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$, it follows that $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$; this joint independence implies the uncorrelating property, $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid r(\mathbf{x})$. But, uncorrelating representations only satisfy the said joint independence when they are independent of the nuisance.

We show B in appendix A.1.7.1 by constructing a nuisance-varying family where the optimal performance is achieved by an uncorrelating representation that is *dependent* on the nuisance.

A.1.7.1 Joint independence vs. the uncorrelating property

Here, we discuss the performance gap between representations that are uncorrelating ($\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid r(\mathbf{x})$) and those that satisfy the joint independence $(\mathbf{y}, r(\mathbf{x})) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$. We construct a data generating process where optimal performance on every member of $\mathcal{F}$ is achieved only by uncorrelating representations that do not satisfy joint independence.

Theorem 2. *Define a nuisance-varying family* $\mathcal{F} = \{p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y}) p_D(\mathbf{z} \mid \mathbf{y}) p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})\}$. *Let* $\mathcal{R}_J = \{r(\mathbf{x}); [r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}\}$ *and* $\mathcal{R}_C = \{r(\mathbf{x}); \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid r(\mathbf{x})\}$ *be the set of representations that, under the nuisance-randomized distribution, satisfy joint independence and conditional independence* *respectively. Then there exists a nuisance-varying family $\mathcal{F}$ such that*

$$\forall p_{te} \in \mathcal{F} \qquad \max_{r \in \mathcal{R}_J} \texttt{Perf}_{p_{te}}(r(\mathbf{x})) \leq \max_{r \in \mathcal{R}_C} \texttt{Perf}_{p_{te}}(r(\mathbf{x})), \tag{A.9}$$

and $\exists p_{te} \in \mathcal{F}$ for which the inequality is strict

$$\max_{r \in \mathcal{R}_J} \texttt{Perf}_{p_{te}}(r(\mathbf{x})) < \max_{r \in \mathcal{R}_C} \texttt{Perf}_{p_{te}}(r(\mathbf{x})), \tag{A.10}$$

Proof. In this proof we will build a nuisance-varying family $\mathcal{F}$ such that $p_{\perp\!\!\!\perp} \in \mathcal{F}$ and $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{x}$. This makes $\mathbf{x}$ a maximally blocking uncorrelating representation because $r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid \mathbf{x}, \mathbf{z}$. Thus it has optimal performance on every $p_{te} \in \mathcal{F}$ within the class of uncorrelating representations. We let $\mathbf{y}$ be binary, $p_{\perp\!\!\!\perp} \in \mathcal{F}$. The structure of the rest of the proof is as follows:

1. The representation $f(\mathbf{x}) = p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x})$ is optimal in that it performs exactly as well as $\mathbf{x}$ on every member of the family $\mathcal{F}$.

2. Any representation $T(\mathbf{x})$ that matches the performance of $\mathbf{x}$ on every $p_{te} \in \mathcal{F}$ satisfies $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x} \mid T(\mathbf{x})$.

3. All functions $T(\mathbf{x})$ such that $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x} \mid T(\mathbf{x})$ determine $f(\mathbf{x})$. This is shown in lemma 5.

4. We construct a family where $f(\mathbf{x}) \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ which, by the point above, means that every optimal representation $T(\mathbf{x})$ is dependent on $\mathbf{z}$: $T(\mathbf{x}) \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$. But every representation $r \in \mathcal{R}_J$ satisfies $r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ and, therefore, is strictly worse in performance than $f(\mathbf{x})$ on $p_{\perp\!\!\!\perp}$, meaning that they perform also strictly worse than $\mathbf{x}$ (because $\texttt{Perf}_{p_{te}}(f(\mathbf{x})) = \texttt{Perf}_{p_{te}}(\mathbf{x})$). Noting $\mathbf{x} \in \mathcal{R}_C$ completes the proof.

For 1, let $f(\mathbf{x}) = p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x})$. However, we show here that $p_{\perp\!\!\!\perp}(\mathbf{y} \mid f(\mathbf{x})) = p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$.

$$p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid f(\mathbf{x})) = \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x} \mid f(\mathbf{x}))} p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x}, f(\mathbf{x}))$$

$$
\begin{aligned}
&= \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x} \mid f(\mathbf{x}))} p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x}) \\
&= \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x} \mid f(\mathbf{x}))} f(\mathbf{x}) \\
&= f(\mathbf{x}) \\
&= p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x}) \quad (= p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}, f(\mathbf{x})))
\end{aligned}
$$

This means $f(\mathbf{x})$ performs exactly as well as $\mathbf{x}$ on every $p_{te} \in \mathcal{F}$ and $\mathbf{x} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid f(\mathbf{x})$.

For 2, recall $\texttt{Perf}_{p_{te}}(r(\mathbf{x})) = -\mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right]$ and note that 1 implies

$$
\texttt{Perf}_{p_{\perp\!\!\!\perp}}(\mathbf{x}) = \texttt{Perf}_{p_{\perp\!\!\!\perp}}(f(\mathbf{x})) = 0.
$$

Let $T(\mathbf{x})$ be any function that performs as well as $f(\mathbf{x})$ on every $p_{te} \in \mathcal{F}$. As $p_{\perp\!\!\!\perp} \in \mathcal{F}$,

$$
\begin{aligned}
0 = \texttt{Perf}_{p_{\perp\!\!\!\perp}}(f(\mathbf{x})) = \texttt{Perf}_{p_{\perp\!\!\!\perp}}(T(\mathbf{x})) & \\
&= -\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid T(\mathbf{x})\right] \\
&= -\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}, T(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid T(\mathbf{x})\right] \\
&= -\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{x} \mid T(\mathbf{x})) \\
&\implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x} \mid T(\mathbf{x}).
\end{aligned}
$$

We leave 3 to lemma 5 and show 4 here.

The example data generating process. We give a data generating process where $f(\mathbf{x}) = p(\mathbf{y} = 1 \mid \mathbf{x})$ is dependent on $\mathbf{z} : f(\mathbf{x}) \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$. We assume $p_{\perp\!\!\!\perp} \in \mathcal{F}$. With a binary $\mathbf{y}$ and a normal $\mathbf{z}$, let $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y})p(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ be generated as follows: with $\rho : \{0,1\} \times \{0,1\} \longrightarrow (0,1)$, let

$$
p(\mathbf{y} = 1) = 0.5, \quad \mathbf{z} \sim \mathcal{N}(0,1), \quad p(\mathbf{b} = 1 \mid \mathbf{y} = y, \mathbf{z} = z) = \rho(y, \mathbf{1}[z \geq 0]), \quad \mathbf{x} = [\mathbf{b}, \mathbf{1}[\mathbf{z} \geq 0]].
$$

We will drop the subscript in $\perp\!\!\!\perp_p$ for readability next. Throughout the next part, we use a key property of independence: $[a, b] \perp\!\!\!\perp c \iff b \perp\!\!\!\perp c \mid a, a \perp\!\!\!\perp c$.

As $\mathbf{z}$ is a standard normal random variable $\mathbf{1}[\mathbf{z} \geq 0] \perp\!\!\!\perp |\mathbf{z}| \mid \mathbf{y}$, meaning we can write $(\mathbf{y}, \mathbf{1}[z \geq 0]) \perp\!\!\!\perp |\mathbf{z}|$ because $\mathbf{z}$ is generated independently of $\mathbf{y}$. Thus, as the distribution of $\mathbf{b}$ only depends on $\mathbf{1}[z \geq 0]$ and $\mathbf{y}$ due to the data generating process, it holds that $(\mathbf{b}, \mathbf{y}, \mathbf{1}[\mathbf{z} \geq 0]) \perp\!\!\!\perp |\mathbf{z}|$. Then

$$(\mathbf{b}, \mathbf{y}, \mathbf{1}[\mathbf{z} \geq 0]) \perp\!\!\!\perp |\mathbf{z}| \implies \mathbf{y} \perp\!\!\!\perp |\mathbf{z}| \mid \mathbf{b}, \mathbf{1}[\mathbf{z} \geq 0] \implies \mathbf{y} \perp\!\!\!\perp \mathbf{z} \mid \mathbf{b}, \mathbf{1}[\mathbf{z} \geq 0] \implies \mathbf{y} \perp\!\!\!\perp \mathbf{z} \mid \mathbf{x}$$

As $\mathbf{x}$ only depends on $\mathbf{1}[\mathbf{z} \geq 0]$ and $\mathbf{b}$, for readability, we define $\mathbf{a} = \mathbf{1}[\mathbf{z} \geq 0]$. Then $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{a}, \mathbf{b}) = p_{\perp\!\!\!\perp}(\mathbf{y}) p_{\perp\!\!\!\perp}(\mathbf{a}) p(\mathbf{b} \mid \mathbf{y}, \mathbf{a})$, where $p(\mathbf{b} = 1 \mid \mathbf{y} = y, \mathbf{a} = a) = \rho(y, a)$ and $\mathbf{x} = [\mathbf{b}, \mathbf{a}]$.

We overload the notation for f: expanding $\mathsf{x} = [b, a]$, we let $f(\mathsf{x}) = f(b, a) = p(\mathbf{y} = 1 \mid \mathbf{x} = [b, a])$. We write $f(b, a)$ for different values of b here,

$$\begin{aligned}
f(1, a) &= p(\mathbf{y} = 1 \mid \mathbf{x} = [1, a]) = p(\mathbf{y} = 1 \mid \mathbf{b} = 1, \mathbf{a} = a) \\
&= \frac{p(\mathbf{b} = 1, \mathbf{y} = 1 \mid \mathbf{a} = a)}{p(\mathbf{b} = 1 \mid \mathbf{a} = a)} \\
&= \frac{p(\mathbf{y} = 1) p(\mathbf{b} = 1 \mid \mathbf{y} = 1, \mathbf{a} = a)}{\sum_{y \in \{0,1\}} p(\mathbf{y} = y) p(\mathbf{b} = 1 \mid \mathbf{y} = y, \mathbf{a} = a)} \\
&= \frac{0.5 p(\mathbf{b} = 1 \mid \mathbf{y} = 1, \mathbf{a} = a)}{0.5 \sum_{y \in \{0,1\}} \left(p(\mathbf{b} = 1 \mid \mathbf{y} = y, \mathbf{a} = a)\right)} \\
&= \frac{\rho(1, a)}{\rho(0, a) + \rho(1, a)}. \\
f(0, a) &= p(\mathbf{y} = 1 \mid \mathbf{x} = [0, a]) = p(\mathbf{y} = 1 \mid \mathbf{b} = 0, \mathbf{a} = a) \\
&= \frac{p(\mathbf{b} = 0, \mathbf{y} = 1 \mid \mathbf{a} = a)}{p(\mathbf{b} = 0 \mid \mathbf{a} = a)} \\
&= \frac{p(\mathbf{y} = 1) p(\mathbf{b} = 0 \mid \mathbf{y} = 1, \mathbf{a} = a)}{\sum_{y \in \{0,1\}} p(\mathbf{y} = y) p(\mathbf{b} = 0 \mid \mathbf{y} = y, \mathbf{a} = a)} \\
&= \frac{0.5(1 - p(\mathbf{b} = 1 \mid \mathbf{y} = 1, \mathbf{a} = a))}{0.5 \left(\sum_{y \in \{0,1\}} 1 - p(\mathbf{b} = 1 \mid \mathbf{y} = y, \mathbf{a} = a)\right)}
\end{aligned}$$

$$= \frac{1 - \rho(1, a)}{2 - \rho(0, a) - \rho(1, a)}.$$

We let $\rho(y, 1) = 0.5$ for $y \in \{0, 1\}$, $\rho(0, 0) = 0.1$, and $\rho(1, 0) = 0.9$. Then, with $a = 1$,

$$f(1, a) = \frac{\rho(1, 1)}{\rho(0, 1) + \rho(1, 1)} = \frac{0.5}{0.5 + 0.5} = 0.5, \tag{A.11}$$

$$f(0, a) = \frac{1 - \rho(1, 1)}{2 - \rho(0, 1) - \rho(1, a)} = \frac{1 - 0.5}{2 - 0.5 - 0.5} = 0.5, \tag{A.12}$$

and with $a = 0$,

$$f(1, a) = \frac{\rho(1, 0)}{\rho(0, 0) + \rho(1, 0)} = \frac{0.9}{0.1 + 0.9} = 0.9, \tag{A.13}$$

$$f(0, a) = \frac{1 - \rho(1, 0)}{2 - \rho(0, 0) - \rho(1, 0)} = \frac{1 - 0.9}{2 - 0.1 - 0.9} = 0.1. \tag{A.14}$$

Thus, the distribution $f(\mathbf{x}) \mid \mathbf{a} = a$ changes with a meaning that $f(\mathbf{x}) \not\perp\!\!\!\perp \mathbf{a}$ which implies $f(\mathbf{x}) \not\perp\!\!\!\perp \mathbf{z}$ as $\mathbf{a} = \mathbf{1}[\mathbf{z} \geq a]$ is a function of $\mathbf{z}$.

Note that $f(\mathbf{x}) \not\perp\!\!\!\perp \mathbf{z}$, then $f(\mathbf{x}) \notin \mathcal{R}_J$ as $f(\mathbf{x}) \perp\!\!\!\perp \mathbf{z}$ is an implication of joint independence. Any function $T(\mathbf{x})$ that achieves the same performance as $p_{\perp\!\!\!\perp}(\mathbf{y} \mid f(\mathbf{x}))$ (by 3 and lemma 5), determines $f(\mathbf{x})$. It follows that, $T(\mathbf{x}) \notin \mathcal{R}_J$ because

$$f(\mathbf{x}) \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \implies T(\mathbf{x}) \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}.$$

So every $r \in \mathcal{R}_J$ must perform worse than $f(\mathbf{x})$, and consequently $\mathbf{x}$, on $p_{\perp\!\!\!\perp}$. Finally, the independence $\mathbf{x} \not\perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$ implies $\mathbf{x} \notin \mathcal{R}_J$ but $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{x}$ and so $\mathbf{x} \in \mathcal{R}_C$. In this example we constructed, $\mathbf{x}$ is the maximally blocking uncorrelating representation which means that it is optimal in $\mathcal{R}_C$ on every $p_{te} \in \mathcal{F}$. As $\mathcal{R}_J$ is a subset, any $r \in \mathcal{R}_J$ can at best match the performance of $\mathbf{x}$ and we already showed that every $r \in \mathcal{R}_J$ is worse than $f(\mathbf{x})$ and consequently $\mathbf{x}$ on $p_{\perp\!\!\!\perp}$. This completes the proof.

□

Lemma 5. *Consider a joint distribution $p(\mathbf{y}, \mathbf{x})$ with binary $\mathbf{y}$. Assume that $p(\mathbf{x} \mid \mathbf{y} = y)$ has the same support for $y \in \{0, 1\}$. Then, for any function $T(\mathbf{x})$ such that $\mathbf{y} \perp\!\!\!\perp \mathbf{x} \mid T(\mathbf{x})$, the function $f(\mathbf{x}) = p(\mathbf{y} = 1 \mid \mathbf{x})$ is $T(\mathbf{x})$-measurable ($T(\mathbf{x})$ determines $f(\mathbf{x})$).*

Proof. We use the notion of sufficient statistics from estimation theory, which are defined for a family of distributions, to define the set of functions $T(\mathbf{x})$ for the joint distribution $p(\mathbf{y}, \mathbf{x})$.

Sufficient statistics in estimation theory. Consider a family of distributions $\mathcal{P} = \{p_\theta(\mathbf{x}); \theta \in \Omega\}$. Assume θ is discrete and that Ω is finite. A function $T(\mathbf{x})$ is a sufficient statistic of a family of distributions $\mathcal{P}$ if the conditional distribution $p_\theta(\mathbf{x} \mid T(\mathbf{x}) = t)$ does not vary with θ for (almost) any value of t. A minimal sufficient statistic is a sufficient statistic $M(x)$ such that for any sufficient statistic $T(X)$ $T(x) = T(x') \implies M(x) = M(x')$. Any bijective transform of $M(\mathbf{x})$ is also a minimal sufficient statistic.

The rest of the proof will follow from relying on theorem 6.12 from [37] which constructs a minimal sufficient statistic for a finite family of distributions $\mathcal{P} = \{p_i; i \in \{0, K-1\}\}$ as

$$M(\mathbf{x}) = \left\{\frac{p_1(\mathbf{x})}{p_0(\mathbf{x})}, \frac{p_2(\mathbf{x})}{p_0(\mathbf{x})}, \cdots, \frac{p_{K-1}(\mathbf{x})}{p_0(\mathbf{x})}\right\}$$

Defining the family with conditionals. Now let the family $\mathcal{P} = \{p_y(\mathbf{x})); y \in \{0, 1\}\}$ where $p_y(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{y} = y)$ which are conditionals of the joint distribution $p(\mathbf{x}, \mathbf{y})$ we were given in the theorem statement. Next, we show that the set of functions $T(\mathbf{x})$ such that $\mathbf{y} \perp\!\!\!\perp_p \mathbf{x} \mid T(\mathbf{x})$ is exactly the set of the sufficient statistics for the family $\{p(\mathbf{x} \mid \mathbf{y} = \mathrm{y}); \mathrm{y} \in \{0, 1\}\}$.

By definition of sufficiency where $p_y(\mathbf{x} \mid T(\mathbf{x}) = t)$ does not vary with y for any value of t,

$$\forall t, \quad p_1(\mathbf{x} \mid T(\mathbf{x}) = t) = p_0(\mathbf{x} \mid T(\mathbf{x}) = t) \iff \forall t, \quad p(\mathbf{x} \mid T(\mathbf{x}) = t, \mathbf{y} = 1) = p(\mathbf{x} \mid T(\mathbf{x}) = t, \mathbf{y} = 0),$$

where the last statement is equivalent to the conditional independence $\mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid T(\mathbf{x})$.

Minimality of $p(\mathbf{y} = 1 \mid \mathbf{x})$. By definition $p_y(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{y} = y)$. As this family contains only two elements, the minimal sufficient statistic is

$$M(\mathbf{x}) = \frac{p_1(\mathbf{x})}{p_0(\mathbf{x})} = \frac{p(\mathbf{x} \mid \mathbf{y} = 1)}{p(\mathbf{x} \mid \mathbf{y} = 0)} = \frac{p(\mathbf{y} = 0)}{p(\mathbf{y} = 1)} \frac{p(\mathbf{y} = 1 \mid \mathbf{x})}{1 - p(\mathbf{y} = 1 \mid \mathbf{x})}.$$

Thus, $M(\mathbf{x})$ is a bijective transformation of the function $p(\mathbf{y} = 1 \mid \mathbf{x})$ (when $p(\mathbf{y} = 1 \mid \mathbf{x}) \in (0, 1)$) which in turn implies that $p(\mathbf{y} = 1 \mid \mathbf{x})$ is a minimal sufficient statistic for the family $\mathcal{P}$.

Conclusion. We showed that the set of functions that satisfy $\mathbf{y} \perp\!\!\!\perp \mathbf{x} \mid T(\mathbf{x})$ are sufficient statistics for the family $\mathcal{P}$. In turn, because only sufficient statistics $T(\mathbf{x})$ of the family $\mathcal{P}$ satisfy $\mathbf{y} \perp\!\!\!\perp_p \mathbf{x} \mid T(\mathbf{x})$, it follows by definition that $p(\mathbf{y} = 1 \mid \mathbf{x})$ is determined by every $T(\mathbf{x})$, completing the proof.

□

A.1.8 Gaussian example of the information criterion

Proposition 3. *Consider the following family of distributions q_a indexed by $a \in \mathbb{R}$,*

$$\epsilon_y, \epsilon_z \sim \mathcal{N}(0, 1) \quad \mathbf{y} \sim \mathcal{N}(0, 1) \qquad \mathbf{z} \sim \mathcal{N}(a\mathbf{y}, 1/2) \qquad \mathbf{x} = [\mathbf{y} + \epsilon_y, \mathbf{z} + \sqrt{1/2}\epsilon_z]$$

In this family, for any $p_D = q_b(\mathbf{y} \mid \mathbf{z})$ where $\mathbf{y} \perp\!\!\!\perp_{p_D} \mathbf{z}$, there exists a $p'_D = q_a(\mathbf{y} \mid \mathbf{z})$ such that

$$\left[\mathbb{E}_{p'_D(\mathbf{x})} KL\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y})\right] > 0.$$

Proof. (of proposition 3) First, write $\mathbf{z} = a\mathbf{y} + \sqrt{1/2}\delta$ where $\delta \sim \mathcal{N}(0, 1)$. Let $\epsilon = \sqrt{1/2}\,(\delta + \epsilon_z)$; this is a normal variable with mean 0 and variance 1. Then, write $\mathbf{x} = [\mathbf{y} + \epsilon_y, a\mathbf{y} + \epsilon]$ where ϵ_y, ϵ are

Gaussian random variables with joint distribution $q(\epsilon_y)q(\epsilon)$. Therefore, $q_a(\mathbf{y}, \mathbf{x})$ is a multivariate Gaussian distribution, with the following covariance matrix (over $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2$):

$$\Sigma = \begin{pmatrix} 1 & 1 & a \\ 1 & 2 & a \\ a & a & a^2+1 \end{pmatrix} \qquad \Longrightarrow \Sigma_{1,2} = [1, a], \quad \Sigma_{2,2,}^{-1} = \frac{1}{a^2+2}\begin{pmatrix} a^2+1 & -a \\ -a & 2 \end{pmatrix},$$

The conditional mean and variance are:

$$\begin{aligned} \mathbb{E}_{q_a}[\mathbf{y} \mid \mathbf{x} = \mathrm{x}] &= \Sigma_{1,2}\Sigma_{2,2}^{-1}\mathrm{x} = \frac{1}{a^2+2}[1, a]\mathrm{x} \\ \sigma^2_{q_a}(\mathbf{y} \mid \mathbf{x}) &= \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1} = 1 - \frac{a^2+1}{a^2+2} = \frac{1}{a^2+2}. \end{aligned}$$

Rewrite the quantity in the theorem statement as a single expression:

$$\begin{aligned} &\mathbb{E}_{q_a(\mathbf{x})}\mathrm{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q_b(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{q_a}(\mathbf{x}; \mathbf{y}) \\ &\quad = \mathbb{E}_{q_a(\mathbf{x})}\mathrm{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q_b(\mathbf{y} \mid \mathbf{x})\right] - \mathbb{E}_{q_a(\mathbf{x})}\mathrm{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q(\mathbf{y})\right]. \\ &\quad = \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\log\frac{q_a(\mathbf{y} \mid \mathbf{x})}{q_b(\mathbf{y} \mid \mathbf{x})} - \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\log\frac{q_a(\mathbf{y} \mid \mathbf{x})}{q(\mathbf{y})}. \\ &\quad = \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y} \mid \mathbf{x})\right). \end{aligned} \tag{A.15}$$

Expand $(\log q(\mathbf{y}) - \log q_b(\mathbf{y} \mid \mathbf{x}))$ in terms of quantities that vary with $\mathbf{y}, \mathbf{x}$ and those that do not:

$$\begin{aligned} \log q(\mathbf{y} = \mathrm{y}) - \log q_b(\mathbf{y} = \mathrm{y} \mid \mathbf{x}) &= -\frac{\mathrm{y}^2}{2} - \log\sqrt{2\pi} + \frac{\left(\mathrm{y} - \mathbb{E}_{q_b}[\mathbf{y} \mid \mathbf{x}]\right)^2}{2\sigma^2_{q_b}(\mathbf{y} \mid \mathbf{x})} + \log\sqrt{2\pi\sigma^2_{q_b}(\mathbf{y} \mid \mathbf{x})} \\ &= -\frac{\mathrm{y}^2}{2} - \log\sqrt{2\pi} + (b^2+2)\frac{\left(\mathrm{y} - \frac{1}{b^2+2}[1, b]\mathbf{x}\right)^2}{2} + \log\sqrt{\frac{2\pi}{b^2+2}} \\ &= -\frac{\mathrm{y}^2}{2} + (b^2+2)\frac{\left(\mathrm{y} - \frac{1}{b^2+2}[1, b]\mathbf{x}\right)^2}{2} + \log\sqrt{\frac{1}{b^2+2}} \end{aligned}$$

As only the first two terms vary with $\mathbf{y}, \mathbf{x}$, compute the expectations $\mathbb{E}_{q_a}$ over these:

$$\begin{aligned}
&\mathbb{E}_{q_a(\mathbf{x})q_a(\mathbf{y} \mid \mathbf{x})}\left(-\frac{\mathbf{y}^2}{2} + (b^2+2)\frac{\left(\mathbf{y} - \frac{1}{b^2+2}[1, b]\mathbf{x}\right)^2}{2}\right) \\
&= \mathbb{E}_{q(\mathbf{y})}\left(-\frac{\mathbf{y}^2}{2}\right) + (b^2+2)\mathbb{E}_{q(\mathbf{y})q_a(\mathbf{x} \mid \mathbf{y})}\frac{\left(\mathbf{y} - \frac{1}{b^2+2}[1, b]\mathbf{x}\right)^2}{2} \\
&= -\frac{1}{2} + (b^2+2)\mathbb{E}_{q(\mathbf{y})q_a(\mathbf{x} \mid \mathbf{y})}\frac{\left((b^2+2)\mathbf{y} - [1, b]\mathbf{x}\right)^2}{2(b^2+2)^2} \\
&= -\frac{1}{2} + \mathbb{E}_{q(\mathbf{y})q_a(\mathbf{x} \mid \mathbf{y})}\frac{\left((b^2+2)\mathbf{y} - [1, b]\mathbf{x}\right)^2}{2(b^2+2)} \\
&= -\frac{1}{2} + \mathbb{E}_{q(\mathbf{y})q(\epsilon_y)q(\epsilon)}\frac{\left((b^2+2)\mathbf{y} - \mathbf{y} - \epsilon_y - ab\mathbf{y} - b\epsilon\right)^2}{2(b^2+2)} \\
&= -\frac{1}{2} + \mathbb{E}_{q(\mathbf{y})q(\epsilon_y)q(\epsilon)}\frac{\left((b^2+1-ab)\mathbf{y} - \epsilon_y - b\epsilon\right)^2}{2(b^2+2)} \\
&= -\frac{1}{2} + \frac{\mathrm{var}\left((b^2+1-ab)\mathbf{y}\right) + \mathrm{var}(\epsilon_y) + \mathrm{var}(b\epsilon)}{2(b^2+2)} \\
&= -\frac{1}{2} + \frac{(b^2+1-ab)^2\mathrm{var}\left(\mathbf{y}\right) + \mathrm{var}(\epsilon_y) + b^2\mathrm{var}(\epsilon)}{2(b^2+2)} \\
&= -\frac{1}{2} + \frac{(b^2+1-ab)^2 + 1 + b^2}{2(b^2+2)} \\
&= \frac{(b^2+1-ab)^2 - 1}{2(b^2+2)}
\end{aligned}$$

The proof follows for any a such that

$$\frac{(b^2+1-ab)^2-1}{2(b^2+2)} + \log\sqrt{1/b^2+2} = \frac{(b^2+1-ab)^2-1}{2(b^2+2)} - \frac{1}{2}\log\left(b^2+2\right) > 0$$

Let $a = b + \frac{1+\nu}{b}$ for some scalar ν. Then, if $|\nu| > 1 + (b^2+2)\log(b^2+2)$,

$$\frac{(b^2+1-ab)^2-1}{2(b^2+2)} - \frac{1}{2}\log\left(b^2+2\right) = \frac{\nu^2-1}{2(b^2+2)} - \frac{1}{2}\log\left(b^2+2\right) > 0.$$

□

A.1.9 Example showing where uncorrelating representations are necessary

In this section, we motivate nuisance-randomization and the uncorrelating property. Consider the following data generating process for a family $\{q_a\}_{a\in\mathbb{R}}$ and fixed positive scalar σ^2:

$$\mathbf{y} \sim \mathcal{N}(0, 1) \quad \mathbf{z} \sim \mathcal{N}(a\mathbf{y}, 0.5) \quad \mathbf{x} = \left[\mathbf{x}_1 \sim \mathcal{N}(\mathbf{y} - \mathbf{z}, \sigma^2 - 0.5), \mathbf{x}_2 \sim \mathcal{N}(\mathbf{y} + \mathbf{z}, 0.5)\right]. \tag{A.16}$$

Letting $\sigma^2 = 2$ recovers the example in eq. (2.2). We keep σ^2 for ease of readability for the first few steps and substitute in 2 later. We show results here in three parts

1. We derive the performance of $p(\mathbf{y}) = q_b(\mathbf{y}) = q(\mathbf{y})$ relative to $q_b(\mathbf{y} \mid \mathbf{x})$ under q_a. The performance of $q(\mathbf{y})$ (recall `Perf` is negative **KL**) relative to $q_b(\mathbf{y} \mid \mathbf{x})$ on $q_a(\mathbf{y}, \mathbf{x})$ can be written as

$$-\mathbb{E}_{q_a(\mathbf{x})}\text{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q(\mathbf{y})\right] - (-\mathbb{E}_{q_a(\mathbf{x})}\text{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q_b(\mathbf{y} \mid \mathbf{x})\right]) \tag{A.17}$$

$$= \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y} \mid \mathbf{x})\right). \tag{A.18}$$

2. We set $b = 1$ and show the performance gap in eq. (A.17) is positive for any $a > 3.48$ and $a < -0.314$, which means that for those q_a, the performance of guessing randomly with $q(\mathbf{y})$ is better than that of training conditional $q_1(\mathbf{y} \mid \mathbf{x})$.

3. Then, we set $b = 0$ and show the performance gap in eq. (A.17) is positive for any $a > 5.63$ and any $a < -2.13$, which means that for those q_a, the performance of guessing randomly with $q(\mathbf{y})$ is better than that of the nuisance-randomized conditional $q_0(\mathbf{y} \mid \mathbf{x}) = p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$.

Performance of $q(\mathbf{y})$ relative to $q_b(\mathbf{y} \mid \mathbf{x})$ under q_a Define the noise in $\mathbf{z}$ to be ϵ_z and the noises in $\mathbf{x}_1, \mathbf{x}_2$ to be ϵ_1, ϵ_2 respectively. Rewrite

$$\mathbf{x} = \left[(1-a) * \mathbf{y} - \sqrt{0.5}\epsilon_z + \sqrt{\sigma^2 - 0.5}\epsilon_1, (1+a) * \mathbf{y} + \sqrt{0.5}\epsilon_2 + \sqrt{0.5}\epsilon_z)\right].$$

Let the joint distribution over $\mathbf{y}, \epsilon_1, \epsilon_2$ be $q(\mathbf{y})q(\epsilon_1)q(\epsilon_2)$. Then, $q_a(\mathbf{y}, \mathbf{x})$ is a multivariate Gaussian distribution. We will now write the covariance matrix of $[\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2]$. As $\mathbf{y}, \epsilon_z, \epsilon_1, \epsilon_2$ are all mutually independent in any combination,

$$\begin{aligned}
\mathbb{V}\mathrm{ar}(\mathbf{y}) &= 1 \tag{A.19}\\
\mathbb{V}\mathrm{ar}(\mathbf{x}_1) &= (1-a)^2\mathbb{V}\mathrm{ar}(\mathbf{y}) + 0.5\mathbb{V}\mathrm{ar}(\epsilon_z) + (\sigma^2 - 0.5)\mathbb{V}\mathrm{ar}(\epsilon_1) = (1-a)^2 + \sigma^2 \tag{A.20}\\
\mathbb{V}\mathrm{ar}(\mathbf{x}_2) &= (1+a)^2\mathbb{V}\mathrm{ar}(\mathbf{y}) + 0.5\mathbb{V}\mathrm{ar}(\epsilon_z) + 0.5\mathbb{V}\mathrm{ar}(\epsilon_2) = (1+a)^2 + 1 \tag{A.21}\\
\mathbf{E}(\mathbf{yx}1) &= (1-a)\mathbb{V}\mathrm{ar}(\mathbf{y}) = 1 - a \tag{A.22}\\
\mathbf{E}(\mathbf{yx}2) &= (1+a)\mathbb{V}\mathrm{ar}(\mathbf{y}) = 1 + a \tag{A.23}\\
\mathbf{E}(\mathbf{x}_1\mathbf{x}2) &= (1-a^2)\mathbb{V}\mathrm{ar}(\mathbf{y}) - 0.5\mathbb{V}\mathrm{ar}(\epsilon_z) = -0.5 + (1-a^2) \tag{A.24}\\
& \tag{A.25}
\end{aligned}$$

The covariance matrix of the joint Gaussian over $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2$

$$\Sigma = \begin{pmatrix} 1 & (1-a) & (1+a) \\ (1-a) & (1-a)^2 + \sigma^2 & (1-a^2) - 0.5 \\ (1+a) & -0.5 + (1-a^2) & (1+a)^2 + 1 \end{pmatrix},$$

$$\begin{aligned}
\implies |\Sigma_{2,2,}| &= (1-a^2)^2 + \sigma^2(1+a)^2 + (1-a)^2 + \sigma^2 - 0.25 - (1-a^2)^2 + (1-a^2)\\
&= \sigma^2(1+a)^2 + 1 - 2a + a^2 + \sigma^2 - 0.5^2 + 1 - a^2\\
&= \sigma^2(1+a)^2 + \sigma^2 - 2a + 2 - 0.5^2
\end{aligned}$$

Letting $\sigma^2 = 2$, we have that

$$|\Sigma_{2,2}| = 2(1 + 2a + a^2) + 2 + 2 - 2a + 0.25 = 2(a + 0.5)^2 + 5.25.$$

Now we derive the mean and the variance terms of the Gaussian conditional distribution $q_a(\mathbf{y} \mid \mathbf{x})$:

$$\begin{aligned}
&\implies \Sigma_{1,2} = [1 - a, 1 + a], \quad \Sigma_{2,2,}^{-1} = \frac{1}{2(a + 0.5)^2 + 5.25} \begin{pmatrix} (1 + a)^2 + 1 & 0.5 - (1 - a^2) \\ 0.5 - (1 - a^2) & (1 - a)^2 + 2 \end{pmatrix}, \\
&\implies \mathbb{E}_{q_a}[\mathbf{y} \mid \mathbf{x} = \mathsf{x}] = \Sigma_{1,2}\Sigma_{2,2}^{-1}\mathsf{x} = \frac{[(1 - a) + 0.5((1 + a)), 2(1 + a) + 0.5((1 - a))]\mathsf{x}}{2(a + 0.5)^2 + 5.25} \\
&\quad = \mathbb{E}_{q_a}[\mathbf{y} \mid \mathbf{x} = \mathsf{x}] = \Sigma_{1,2}\Sigma_{2,2}^{-1}\mathsf{x} = \frac{1}{2(a + 0.5)^2 + 5.25}[0.5(3 - a), 0.5(5 + 3a)]\mathsf{x} \\
&\implies \sigma_{q_a}^2(\mathbf{y} \mid \mathbf{x}) \\
&\qquad = \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1} \\
&\qquad = 1 - \frac{\sigma^2(1 + a)^2 + (1 - a)^2 + (1 - a^2)}{2(a + 0.5)^2 + 5.25} \\
&\qquad = \frac{2(a + 0.5)^2 + 5.25 - \sigma^2(1 + a)^2 - ((1 - a)^2 + (1 - a^2))}{2(a + 0.5)^2 + 5.25} \\
&\qquad = \frac{\sigma^2 - 2a + 2 - 0.5^2 - (1 - 2a + a^2 + 1 - a^2)}{2(a + 0.5)^2 + 5.25} \\
&\qquad = \frac{\sigma^2 - 2a + 2 - 0.5^2 - (2 - 2a)}{2(a + 0.5)^2 + 5.25} \\
&\qquad = \frac{\sigma^2 - 0.5^2}{2(a + 0.5)^2 + 5.25} \\
&\qquad = \frac{1.75}{2(a + 0.5)^2 + 5.25}
\end{aligned}$$

The performance of $q(\mathbf{y})$ (recall `Perf` is negative **KL**) relative to $q_b(\mathbf{y} \mid \mathbf{x})$ on $q_a(\mathbf{y}, \mathbf{x})$ can be written as

$$\mathbb{E}_{q_a(\mathbf{x})}\mathrm{KL}\left[q_a(\mathbf{y}\mid\mathbf{x}) \parallel q_b(\mathbf{y}\mid\mathbf{x})\right] - \mathbb{E}_{q_a(\mathbf{x})}\mathrm{KL}\left[q_a(\mathbf{y}\mid\mathbf{x}) \parallel q(\mathbf{y})\right] = \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y}\mid\mathbf{x})\right).$$

Expand $(\log q(\mathbf{y}) - \log q_b(\mathbf{y}\mid\mathbf{x}))$ in terms that vary with $\mathbf{y}, \mathbf{x}$ and those that do not:

$$\begin{aligned}
&\log q(\mathbf{y}=\mathrm{y}) - \log q_b(\mathbf{y}=\mathrm{y}\mid\mathbf{x}) \\
&= -\frac{\mathrm{y}^2}{2} - \log\sqrt{2\pi} + \frac{\left(\mathrm{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{2\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} + \log\sqrt{2\pi\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} \\
&= -\frac{\mathrm{y}^2}{2} - \log\sqrt{2\pi} + (2(b+0.5)^2 + 5.25)\frac{\left(\mathrm{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{2(1.75)} + \log\sqrt{2\pi\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} \\
&= -\frac{\mathrm{y}^2}{2} + (2(b+0.5)^2 + 5.25)\frac{\left(\mathrm{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{3.5} + \log\sqrt{\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})}
\end{aligned} \tag{A.26}$$

Next, we want to compute $\mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y}\mid\mathbf{x})\right)$. As $q_a(\mathbf{y})$ is a standard normal distribution, the first term averages to $-\frac{1}{2}$. Then, the only other random quantity is $\left(\mathrm{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2$, and we compute

$$\mathbf{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2.$$

For this, expand $\left(\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)$ noting that $\mathbf{x}$ comes from $q_a(\mathbf{y},\mathbf{x})$ and so is

$$\mathbf{x} = \left[(1-a) * \mathbf{y} - \sqrt{0.5}\epsilon_z + \sqrt{1.5}\epsilon_1, (1+a) * \mathbf{y} + \sqrt{0.5}\epsilon_2 + \sqrt{0.5}\epsilon_z)\right].$$

$$\begin{aligned}
\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}] &= \mathbf{y} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\mathrm{x} \\
&= \mathbf{y} - \frac{1}{2(b+0.5)^2 + 5.25}[0.5(3-b), 0.5(5+3b)]\mathbf{x}
\end{aligned}$$

When does the training conditional perform worse than random? Now let's set $b = 1$. Then, $2(b + 0.5)^2 + 5.25 = 2(1 + 0.5)^2 + 5.25 = 9.75$ and we get

$$\begin{aligned}
\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y} \mid \mathbf{x}] &= \mathbf{y} - \frac{1}{2(b + 0.5)^2 + 5.25}[1, 4]\mathbf{x} \\
&= \mathbf{y} - \frac{1}{2(1 + 0.5)^2 + 5.25}[1, 4]\mathbf{x} \\
&= \mathbf{y} - \frac{1}{9.75}\left(1 * [(1 - a) * \mathbf{y} - \sqrt{0.5}\epsilon_z + \sqrt{1.5}\epsilon_1] + 4 * [(1 + a) * \mathbf{y} + \sqrt{0.5}\epsilon_2 + \sqrt{0.5}\epsilon_z)]\right)
\end{aligned}$$

This expression takes the form $\frac{1}{9.75}\left(\alpha\mathbf{y} + \beta\epsilon_z + \gamma * \epsilon_1 + \theta * \epsilon_2\right)$. As $\mathbf{y}, \epsilon_z, \epsilon_1, \epsilon_2$ are mutually independent standard normal random variables, $\mathbf{E}_{\mathbf{y},\epsilon_z,\epsilon_1,\epsilon_2}\left(\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y} \mid \mathbf{x}(\mathbf{y}, \epsilon_z, \epsilon_1, \epsilon_2)]\right)^2$ will evaluate to

$$\frac{1}{(9.75)^2}\left(\alpha^2 + \beta^2 + \gamma^2 + \theta^2\right).$$

Now,

$$\begin{aligned}
\alpha &= 9.75 - (1 - a) - 4 * (1 + a) = 4.75 - 3a \\
\beta &= -1(-\sqrt{0.5}) - 4 * \sqrt{0.5} = -3\sqrt{0.5} \\
\gamma &= -1 * \sqrt{1.5} \\
\theta &= -4 * \sqrt{0.5}
\end{aligned}$$

The last term to compute is

$$\log\sqrt{\sigma^2_{q_b}(\mathbf{y} \mid \mathbf{x})} = \frac{1}{2}\log\frac{1.75}{2(b + 0.5)^2 + 5.25} = 0.5\log\frac{1.75}{9.75} > -0.8588$$

Thus, following eq. (A.26)

$$
\begin{aligned}
\mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y}\mid\mathbf{x})\right) &= \mathbb{E}_{q(\mathbf{y})} - \frac{\mathbf{y}^2}{2} + (2(b+0.5)^2+5.25)\frac{\left(\mathbf{y}-\mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{3.5} \\
&= \mathbb{E}_{q(\mathbf{y})} - \frac{\mathbf{y}^2}{2} + (9.75)\frac{\left(\mathbf{y}-\mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{3.5} + \log\sqrt{\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} \\
&> -\frac{1}{2} + \frac{9.75}{3.5}\frac{1}{(9.75)^2}\left(\alpha^2+\beta^2+\gamma^2+\theta^2\right) - 0.8588 \\
&= -\frac{1}{2} + \frac{9.75}{3.5}\frac{1}{(9.75)^2}\left((4.75-3a)^2 + (-3\sqrt{0.5})^2 + (-1*\sqrt{1.5})^2 + (-4*\sqrt{0.5})^2\right) - 0.8588 \\
&= \frac{1}{3.5*9.75}\left((4.75-3a)^2 + 4.5 + 1.5 + 8\right) - 1.3588 \\
&= \frac{1}{34.125}\left((4.75-3a)^2 + 14\right) - 1.3588
\end{aligned}
$$

By solving a quadratic equation, this number is greater than 0 for every value $a > 3.48$ and $a < -0.314$. This concludes that the training conditional $q_1(\mathbf{y}\mid\mathbf{x})$ performs worse than predicting randomly with $q(\mathbf{y})$ on some members of the nuisance-varying family.

When does the nuisance-randomized conditional perform worse than random if $\mathbf{x}$ is not uncorrelating? Now let's set $b = 0$. Then, $2(b+0.5)^2 + 5.25 = 2(0.5)^2 + 5.25 = 5.75$ and we get

$$
\begin{aligned}
\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}] &= \mathbf{y} - \frac{1}{2(b+0.5)^2+5.25}[1.5, 2.5]\mathbf{x} \\
&= \mathbf{y} - \frac{1}{2(0.5)^2+5.25}[1.5, 2.5]\mathbf{x} \\
&= \mathbf{y} - \frac{1}{5.75}\Big(1.5*[(1-a)*\mathbf{y} - \sqrt{0.5}\epsilon_z + \sqrt{1.5}\epsilon_1] \\
&\qquad + 2.5*[(1+a)*\mathbf{y} + \sqrt{0.5}\epsilon_2 + \sqrt{0.5}\epsilon_z)]\Big)
\end{aligned}
$$

This expression takes the form $\frac{1}{5.75}\left(\alpha\mathbf{y} + \beta\epsilon_z + \gamma*\epsilon_1 + \theta*\epsilon_2\right)$. As $\mathbf{y}, \epsilon_z, \epsilon_1, \epsilon_2$ are mutually independent standard normal random variables, $\mathbf{E}_{\mathbf{y},\epsilon_z,\epsilon_1,\epsilon_2}\left(\mathbf{y} - \mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}(\mathbf{y},\epsilon_z,\epsilon_1,\epsilon_2)]\right)^2$ will evaluate to

$$\frac{1}{(5.75)^2}\left(\alpha^2+\beta^2+\gamma^2+\theta^2\right).$$

Now,

$$\begin{aligned}
\alpha &= 5.75 - 1.5(1-a) - 2.5 * (1+a) = 1.75 - a \\
\beta &= -1.5 * (-\sqrt{0.5}) - 2.5 * \sqrt{0.5} = -\sqrt{0.5} \\
\gamma &= -1.5 * \sqrt{1.5} \\
\theta &= -2.5 * \sqrt{0.5}
\end{aligned}$$

The last term to compute is

$$\log\sqrt{\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} = \frac{1}{2}\log\frac{1.75}{2(b+0.5)^2+5.25} = 0.5\log\frac{1.75}{5.75} = 0.5\log\frac{7}{23} > -0.5947$$

Thus, following eq. (A.26)

$$\begin{aligned}
\mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y}\mid\mathbf{x})\right) &= \mathbb{E}_{q(\mathbf{y})} - \frac{\mathbf{y}^2}{2} + (2(b+0.5)^2+5.25)\frac{\left(\mathbf{y}-\mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{3.5} \\
&= \mathbb{E}_{q(\mathbf{y})} - \frac{\mathbf{y}^2}{2} + (5.75)\frac{\left(\mathbf{y}-\mathbb{E}_{q_b}[\mathbf{y}\mid\mathbf{x}]\right)^2}{3.5} + \log\sqrt{\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} \\
&> -\frac{1}{2} + \frac{5.75}{3.5}\frac{1}{(5.75)^2}\left(\alpha^2+\beta^2+\gamma^2+\theta^2\right) - 0.5947 \\
&= -\frac{1}{2} + \frac{5.75}{3.5}\frac{1}{(5.75)^2}\left((1.75-a)^2 + (-\sqrt{0.5})^2 + (-1.5 * \sqrt{1.5})^2 + (-2.5 * \sqrt{0.5})^2\right) - 0.5948 \\
&= \frac{1}{3.5 * 5.75}\left((1.75-a)^2 + (-\sqrt{0.5})^2 + (-1.5 * \sqrt{1.5})^2 + (-2.5 * \sqrt{0.5})^2\right) - 1.0947 \\
&= \frac{1}{20.125}\left((1.75-a)^2 + 0.5 + 3.375 + 3.125\right) - 1.0947 \\
&= \frac{1}{20.125}\left((1.75-a)^2 + 7\right) - 1.0947
\end{aligned}$$

By solving a quadratic equation, this number is greater than 0 for every value $a > 5.63$ and $a < -2.13$. This concludes that the nuisance-randomized conditional $q_0(\mathbf{y} \mid \mathbf{x}) = p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ performs worse than predicting randomly with $q(\mathbf{y})$ on some members of the nuisance-varying family when $\mathbf{x}$ is not uncorrelating.

A.2 Further experimental details

Implementation details In section 2.5, the label $\mathbf{y}$ is a binary variable and, consequently, we use the Bernoulli likelihood in the predictive model and the weight model. In reweighting-NuRD in practice, the estimate of the nuisance-randomized distribution $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) \propto {}^{p_{tr}(\mathbf{y})}/_{\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})} p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x})$ with an estimated $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ may have a different marginal distribution $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}) \neq p_{tr}(\mathbf{y})$. To ensure that $p_{tr}(\mathbf{y}) = \hat{p}_{\perp\!\!\!\perp}(\mathbf{y})$, we weight our preliminary estimate $\hat{p}_{\perp\!\!\!\perp}$ again as $\frac{p_{tr}(\mathbf{y})}{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y})} \hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x})$.

In all the experiments, the distribution $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is a Bernoulli distribution parameterized by r_γ and a scaling parameter θ. In general, when the family of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is unknown, learning predictive models requires a parameterization $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$. When the family is known, for example when $\mathbf{y}$ is categorical, the parameters θ are not needed because the distribution $p(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ can be parameterized by the representation itself. For the critic model $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$ in the distillation step, we use a two layer neural network with 16 hidden units and ReLU activations that takes as input $\mathbf{y}, r_\gamma(\mathbf{x})$, and a scalar representation $s_\psi(\mathbf{z})$; the critic model's parameters are ϕ, ψ. The representation $s_\psi(\mathbf{z})$ is different in the different experiments and we give these details below.

In generative-NuRD, we select models for $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ by using the generative objective's value on a heldout subset of the training data. For model selection, we use Gaussian likelihood in the class-conditional Gaussian experiment, binary likelihood in the colored-MNIST experiment, and squared-loss reconstruction error in the Waterbirds and chest X-ray experiments. In reweighting-NuRD, we use a cross-fitting procedure where the training data is split into K folds, and K models are trained: for each fold, we produce weights using a model trained and validated on the other $K-1$ folds. Hyperparameter selection for the distillation step is done using the distillation loss from eq. (2.6) evaluated on a heldout validation subset of the nuisance-randomized data from the first step.

In all experiments, we report results with the distillation step optimized with a fixed $\lambda = 1$ and with 1 or 2 epochs worth of critic model updates per every representation update. In setting the hyperparameter λ, a practitioner should choose the largest λ such that optimization is still stable for different seeds and the validation loss is bounded away from that of marginal prediction. Next, we give details about each individual experiment.

Optimal linear uncorrelating representations in Class Conditional Gaussians. Here, we show that $r^*(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$ is the best linear uncorrelating representation in terms of performance. First let the Gaussian noises in the two coordinates of $\mathbf{x}$ (given $\mathbf{y}, \mathbf{z}$) be $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(0, 9)$ and $\boldsymbol{\epsilon}_2 \sim \mathcal{N}(0, 0.01)$ respectively. Define $r_{u,v}(\mathbf{x}) = u\mathbf{x}_1 + v\mathbf{x}_2 = (u+v)\mathbf{y} + (v-u)\mathbf{z} + u\boldsymbol{\epsilon}_1 + v\boldsymbol{\epsilon}_2$. We will show that $q_0(\mathbf{z} \mid r_{u,v}(\mathbf{x}), \mathbf{y} = 1) \neq q_0(\mathbf{z} \mid r_{u,v}(\mathbf{x}), \mathbf{y} = 0)$ when $u \neq v$ and $u \neq -v$. First, $q_0(\mathbf{z}, r_{u,v}(\mathbf{x}) \mid \mathbf{y} = y)$ is a bivariate Gaussian with the following covariance matrix:

$$\Sigma_y = \begin{pmatrix} 1 & (v-u) \\ (v-u) & (v-u)^2 + 9u^2 + 0.01v^2 \end{pmatrix} \implies \Sigma_{y;1,2} = v - u, \quad \Sigma_{y;2,2}^{-1} = \frac{1}{(v-u)^2 + 9u^2 + 0.01v^2}$$

The conditional mean is:

$$\begin{aligned} \mathbb{E}_{q_a}[\mathbf{z} \mid r_{u,v}(\mathbf{x}) = r, \mathbf{y}] &= \mathbb{E}[\mathbf{z} \mid \mathbf{y} = 1] + \Sigma_{1,2}\Sigma_{2,2}^{-1}\left(r - \mathbb{E}[r_{u,v}(\mathbf{x}) \mid \mathbf{y} = 1]\right) \\ &= \mathbb{E}[\mathbf{z}] + \Sigma_{1,2}\Sigma_{2,2}^{-1}\left(r - (u+v)\mathbf{y}\right) \\ &= \frac{(v-u)(r - (u+v)\mathbf{y})}{(v-u)^2 + 9u^2 + 0.01v^2} \end{aligned}$$

which is independent of $\mathbf{y}$ if and only if $u + v = 0$ or $u - v = 0$. The conditional variance does not change with y because it is determined by Σ_y which does not change with y. Thus $q_0(\mathbf{z} \mid r_{u,v}(\mathbf{x}), \mathbf{y}) = q_0(\mathbf{z} \mid r_{u,v}(\mathbf{x}))$ if and only if $u = v$ or $u = -v$. When $u = v$, $r_{u,v} = 2u\mathbf{y} + \text{noise}$ and $r_{u,u} \not\perp\!\!\!\perp_{q_0} \mathbf{y}$ meaning that $r_{u,u}$ helps predict $\mathbf{y}$. In contrast, when $u = -v$, $r_{u,v} = 2v\mathbf{z} + \text{noise}$ and so $r_{-v,v} \perp\!\!\!\perp_{q_0} \mathbf{y} \implies q_0(\mathbf{y} \mid r_{-v,v}) = q_0(\mathbf{y})$, meaning that $q_0(\mathbf{y} \mid r_{-v,v})$ has the same performance as the marginal. However, for all $u \neq 0$, $r_{u,u} = ur_{1,1}$ is a bijective transform of $r_{1,1}$ and, therefore, $q_0(\mathbf{y} \mid r_{u,u}(\mathbf{x})) = q_0(\mathbf{y} \mid r_{1,1}(\mathbf{x}))$. Thus, within the set of linear uncorrelating representations, $r_{1,1}$ is the best because its performance dominates all others on every $p_{te} \in \mathcal{F}$.

Implementation details for Class Conditional Gaussians. In reweighting-NuRD, the model for $p_{tr}(\mathbf{y} \mid \mathbf{z})$ is a Bernoulli distribution parameterized by a neural network with 1 hidden layer with 16 units and ReLU activations. In generative-NuRD, the model for $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ is an isotropic Gaussian whose mean and covariance are parameterized with a neural network with one layer with 16 units and ReLU activations. We use 5 cross-fitting folds in estimating the weights in reweighting-NuRD. We use weighted sampling with replacement in computing the distillation objective.

In the distillation step in both reweighting and generative-NuRD, the representation $r_\gamma(\mathbf{x})$ is a neural network with one hidden layer with 16 units and ReLU activations. The critic model $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$ consists of a neural network with 2 hidden layers with 16 units each and ReLU activations that takes as input $\mathbf{y}, r_\gamma(\mathbf{x})$, and a scalar representation $s_\psi(\mathbf{z})$ which is again a neural network with a single hidden layer of 16 units and ReLU activations.

We use cross entropy to train $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$, $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$, and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$ using the Adam [90] optimizer with a learning rate of 10^{-2}. We optimized the model for $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ for 100 epochs and the model for $\hat{p}_{tr}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ for 300 epochs. We ran the distillation step for 150 epochs with the Adam optimizer with the default learning rate. We use a batch size of 1000 in both stages of NuRD. We run the distillation step with a fixed $\lambda = 1$ and two epoch's worth of gradient steps

(16) for the critic model for each gradient step of the predictive model and the representation. In this experiment, we do not re-initialize ϕ, ψ after a predictive model update.

Implementation details for Colored-MNIST. For reweighting-NuRD, to use the same architecture for the representation $r_\gamma(\mathbf{x})$ and for $p_{tr}(\mathbf{y} \mid \mathbf{z})$, we construct the nuisance as a 28×28 image with each pixel being equal to the most intense pixel in the original image. In generative-NuRD, we use a PixelCNN model for $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ with 10 masked convolutional layers each with 64 filters. The model was trained using a Bernoulli likelihood with the Adam optimizer and a fixed learning rate of 10^{-3} and batch size 128. We parameterize multiple models in this experiment with the following neural network: 4 convolutional layers (with $32, 64, 128, 256$ channels respectively) with ReLU activations followed by a fully connected linear layer into a single unit. Both $r_\gamma(\mathbf{x}), s_\psi(\mathbf{z})$ are parameterized by this network. Both $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ in reweighting-NuRD and $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{x})$ for erm are Bernoulli distributions parameterized by the network described above. We use 5 cross-fitting folds in estimating the weights in reweighting-NuRD.

For the critic model $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$ in the distillation step, we use a two-hidden-layer neural network with 16 hidden units and ReLU activations that takes as input $\mathbf{y}, r_\gamma(\mathbf{x})$, and the scalar representation $s_\psi(\mathbf{z})$; the parameters ϕ contain ψ and the parameters for the two hidden-layer neural network. The predictive model $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is a Bernoulli distribution parameterized by $r_\gamma(\mathbf{x})$ multiplied by a scalar θ.

We use cross entropy to train $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$, $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$, and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$ using the Adam [90] optimizer with a learning rate of 10^{-3}. We optimized the model for $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ for 20 epochs and ran the distillation step for 20 epochs with the Adam optimizer with the default learning rate. We use a batch size of 300 in both stages of NuRD. We run the distillation step with a fixed $\lambda = 1$ and one epoch's worth of gradient steps (14) for the critic model for each gradient step of the predictive model and the representation. In this experiment, we do not re-initialize ϕ, ψ after a predictive model update.

Implementation details for the Waterbirds experiment. For generative-NuRD, we use VQ-VAE 2 [238] to model $p_{tr}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$. For multiple latent sizes and channels in the encoder and the decoder, we saw that the resulting generated images were insufficient to build classifiers that predict better than chance on real data. This may be because of the small training dataset that consists of only 3000 samples. The model for $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is two feedforward layers stacked on top of the representation $r_\gamma(\mathbf{x})$. The model $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ in reweighting-NuRD is the same model as $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ as a function of $\mathbf{x}$. The model for $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$ consists of a neural network with two feedforward layers that takes as input $\mathbf{y}, r_\gamma(\mathbf{x})$, and a representation $s_\psi(\mathbf{z})$. Both r_γ and s_ψ are Resnet-18 models initialized with weights pretrained on Imagenet; the parameters ϕ contain ψ and the parameters for the two hidden-layer neural network. The model in ERM for $p_{tr}(\mathbf{y} \mid \mathbf{x})$ uses the same architecture as $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ as a function of $\mathbf{x}$. We use 5 cross-fitting folds in estimating the weights in reweighting-NuRD.

We use binary cross entropy as the loss in training $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$, $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$, and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x})))$ using the Adam [90] optimizer with a learning rate of 10^{-3}. We optimized the model for $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ for 10 epochs and ran the distillation step for 5 epochs with the Adam optimizer with the default learning rate for all parameters except γ, which parameterizes the representation r_γ; for γ, we used 0.0005. The predictive model, the critic model, and the weight model are all optimized with a weight decay of 0.01. We use a batch size of 300 for both stages of NuRD. We run the distillation step with a fixed $\lambda = 1$ and two epoch's worth of gradient steps (16) for the critic model for each gradient step of the predictive model and the representation. To prevent the critic model from overfitting, we re-initialize ϕ, ψ after every gradient step of the predictive model.

Implementation details for the chest X-ray experiment. To help with generative modeling, when creating the dataset, we remove X-ray samples from MIMIC that had all white or all black borders. We use a VQ-VAE2 [238] to model $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ using code from here to both train and sample. The encoder takes the lung patch as input, and the decoder takes the quantized embeddings and the non-lung patch as input. VQ-VAE2 is hierarchical with a top latent code and a bottom latent code which are both vector-quantized and fed into the decoder to reconstruct the image. Both latents consist of 8×8 embeddings each of dimension 64. The VQ-VAE is trained for 200 epochs with Adam [90] with a batch size of 256 and dropout rate of 0.1. Generating samples from the VQ-VAE2 involves sampling the top latent code conditioned on the label, followed by sampling the bottom latent code conditioned on the label and the top latent code, and passing both latent codes to the decoder. To generate from the latent codes, we build a PixelSNAIL to generate the top latent code given the label and a PixelCNN to generate the bottom latent code given the label and the top latent code. These models have 5 residual layers with 128 convolutional channels. All other details were default as in here. We train these models for 450 epochs with a batch size of 256 with a learning rate of 5×10^{-5}.

For reweighting-NuRD, the model $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is two feedforward layers stacked on top of the representation $r_\gamma(\mathbf{x})$. The model in ERM for $p_{tr}(\mathbf{y} \mid \mathbf{x})$ uses the same architecture as $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ as a function of $\mathbf{x}$. Next we use a single architecture to parameterize multiple parts in this experiment: 3 convolutional layers (each 64 channels) each followed by batch norm, and dropout with a rate of 0.5 and followed by a linear fully-connected layer into a single unit. We parameterize the two representations $r_\gamma(\mathbf{x}), s_\psi(\mathbf{z})$ with this network. To build $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$, we stack two feedforward layers of 16 hidden units with ReLU activations on top of a concatenation of $\mathbf{y}$, $r_\gamma(\mathbf{x})$,and the scalar representation $s_\psi(\mathbf{z})$ as described above; the parameters ϕ contain ψ and the parameters for the two hidden-layer neural network. We use 5 cross-fitting folds in estimating the weights in reweighting-NuRD.

We use binary cross entropy as the loss in training $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$, $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$, and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x})))$ using the Adam [90] optimizer with a learning rate of 10^{-3}. We use a batch size of 1000 for both stages of NuRD. We optimized the model for $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ for 150 epochs and ran the distillation step for 100 epochs with the Adam optimizer with the default learning rate. Only the optimization for

$\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ has a weight decay of $1e-2$. We run the distillation step with a fixed $\lambda = 1$ and two epoch's worth of gradient steps (20) for the critic model for each gradient step of the predictive model and the representation. To prevent the critic model from overfitting, we re-initialize ϕ, ψ after every gradient step of the predictive model.

A.2.1 Additional experiments

Excluding the boundary from the images (covariates) does not improve erm in general. In both Waterbirds and chest X-rays, we use the easy-to-acquire border as a nuisance in NuRD. Models trained on the central (non-border) regions of the image can exploit the nuisances in the center and consequently fail to generalize when the nuisance-label relationship changes. In fact, classifiers produced by erm on border-less images do not generalize well to the test data, producing test accuracies of $39 \pm 0.5\%$ on chest X-rays and $65 \pm 2.3\%$ on Waterbirds averaged over 10 seeds. However, as independence properties that hold for the border also hold for nuisances in the central region that are determined by the border, NuRD can use the border to control for certain nuisances in the center of the image.

Additional experiments with NuRD. We evaluate reweighting-NuRD further in the following ways:

1. Run NuRD on data from the training data distribution defined in section 2.5 and evaluate on data from test distributions p_{te} with different nuisance-label relationships.
2. Train NuRD with different-sized borders as nuisances.
3. Train NuRD without a nuisance where the training and the test data have the same nuisance-label relationship; we implement this by setting the nuisance $\mathbf{z} = 0$ wherever it is passed as input in the weight model or critic model.
4. Run the distillation step with different λ.

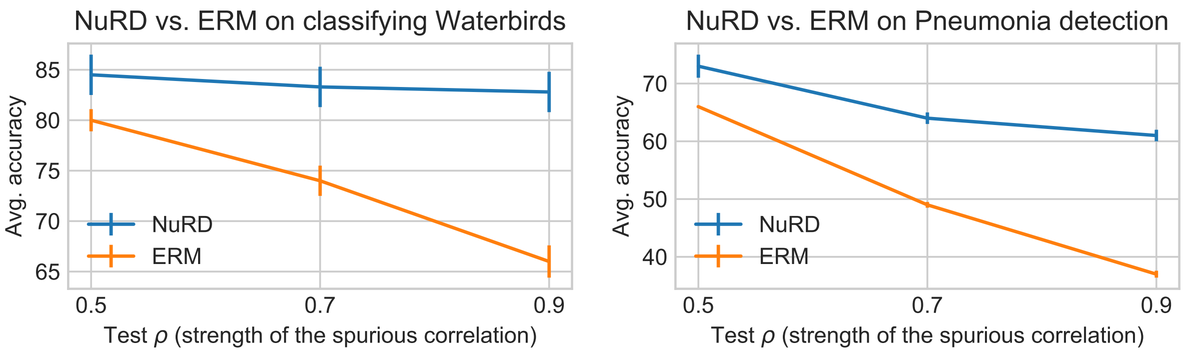


Figure A.2: Plots of average accuracy vs. test ρ for classifying Waterbirds and Pneumonia. A larger ρ implies a larger difference between the nuisance-label relationship in the test data used for evaluation and the training data, which has a $\rho = 0.1$. Nuisance-randomized data corresponds to $\rho = 0.5$. Unlike NuRD, ERM's performance quickly degrades as the difference between the train and the test distributions increases.

Different test distributions. For this experiment, we compute the test accuracies of the models trained in the experiments in section 2.5 on data with different nuisance-label relationships. For both classifying Waterbirds and Pneumonia, a scalar parameter ρ controls nuisance-label relationships in the data generating process. In waterbirds,

$$\rho = p(\mathbf{y} = \textit{waterbird} \mid \text{background} = \text{land}) = p(\mathbf{y} = \textit{landbird} \mid \text{background} = \text{water}).$$

In chest X-rays, ρ corresponds to the fraction of Pneumonia cases that come from CheXpert and normal cases that come from MIMIC in the data; in this task, hospital differences are one source of nuisance-induced spurious correlations. In both tasks, $\rho = 0.1$ in the training data; as test ρ increases, the nuisance-label relationship changes and becomes more different from the training data. We plot the average and standard error of accuracies aggregated over 10 seeds for different test $\rho \in \{0.5, 0.7, 0.9\}$ in fig. A.2.

Nuisance specification with different borders. For Waterbirds, we ran NuRD with the pixels outside the central 168x168 patch (a 56 pixel border) as the nuisance. Averaged over 10 seeds, reweighting-NuRD produced a model with 81% test accuracy which is similar to the accuracy achieved by NuRD using a 28-pixel border as the nuisance. In comparison, ERM achieves an accuracy of 66%.

NuRD without a nuisance and no nuisance-induced spurious correlations in classifying Waterbirds. We performed an additional experiment on classifying Waterbirds where NuRD is given a constant nuisance which is equivalent to not using the nuisance. We generated training and test data with independence between the nuisance and the label; the nuisance-label relationship does not change between training and test. Averaged over 10 seeds, ERM achieved a test accuracy of 89 ± 0.4% and NuRD achieved a test accuracy of 88 ± 1%.

Reweighting-NuRD with different λ. Large λs may make optimization unstable by penalizing even small violations of joint independence. Such instabilities can lead NuRD to build predictive models that do not do better than marginal prediction, resulting in large distillation loss (log-likelihood + information loss) on the validation subset of the training data. However, a small λ may result in NuRD learning non-uncorrelating representations which can also perform worse than chance.

We ran NuRD on the waterbirds and class-conditional Gaussians experiments with $\lambda = 5$ (instead of $\lambda = 1$ like in section 2.5) and found that, on a few seeds, NuRD produces models with close to 50% accuracy (which is the same as majority prediction) or large information loss or both. Excluding seeds with large validation loss, reweighting-NuRD achieves an average test accuracy of 76% on waterbirds and 61% on class-conditional Gaussians. Annealing the λ during training could help stabilize optimization.

In setting the hyperparameter λ in general, a practitioner should choose the largest λ such that optimization is still stable over different seeds and the validation loss is bounded away from that of predicting without any features.

B | Appendices for chapter 3

B.1 Proofs and Discussion on Semantic Corruptions

In this section we give the proofs of Theorem 2 and Proposition 1. The first result shows that even if we know our training and test data are sampled from distributions in a nuisance varying family $\mathcal{F}$, additional assumptions are required in order to learn a predictor that is robust across the entire family.

Theorem 1. *For any learning algorithm, there exists a nuisance-varying family $\mathcal{F}$ where predicting with $p_{\perp\!\!\!\perp}(\mathbf{y} = 1 \mid \mathbf{x})$ achieves 90% accuracy on all members such that given training data $\mathbf{y}, \mathbf{x}$ from one member $p_{tr} \in \mathcal{F}$, the algorithm cannot achieve better accuracy than predicting at random on some $p_{te} \in \mathcal{F}$.*

Proof. At a high-level, we setup two nuisance-varying families $\mathcal{F}_1 = \{p_{1,\rho}\}, \mathcal{F}_2 = \{p_{2,\rho}\}$ where

1. There are members of each family that have the same distribution over $(\mathbf{y}, \mathbf{x})$. We let this distribution over $\mathbf{y}, \mathbf{x}$ be the training data.
2. Thus looking at this training data alone, no algorithm can tell which family the test distribution will come from.
3. Then, the proof concludes by showing any predictor that performs better than the chance

on all members of $\mathcal{F}_1$, will perform worse than chance on a member of $\mathcal{F}_2$.

Defining the two families. We now define two nuisance-varying families $\mathcal{F}_1 = \{p_{1,\rho}\}$ and $\mathcal{F}_2 = \{p_{2,\rho}\}$. For $a \in \{-1, 1\}$, and $\alpha \in [0, 1]$ let $\mathbf{R}_\alpha(a)$ be a probability distribution obtained by randomly flipping the sign of a with probability $1 - \alpha$:

$$r \sim \mathbf{R}_\alpha(a) \implies \begin{cases} p(r = a) = \alpha \\ p(r = -a) = 1 - \alpha \end{cases} \tag{B.1}$$

Then, define the family $\{p_{1,\rho}\}$ as the distributions resulting from the following sampling process:

$$\mathbf{y} \sim \mathbf{R}_{0.5}(1)$$
$$\mathbf{z} \sim \mathbf{R}_\rho(\mathbf{y})$$
$$\mathbf{x}^* \sim \mathbf{R}_{0.9}(\mathbf{y})$$
$$\mathbf{x} = [\mathbf{x}^*, \mathbf{z}]$$

The second family $p_{2,\rho}$ follows the same process except that the positions of the semantic feature and nuisance are flipped $\mathbf{x} = [\mathbf{z}, \mathbf{x}^*]$. **Notice that predicting $\mathbf{y}$ from $\mathbf{x}_1$ in $\mathcal{F}_1$ and from $\mathbf{x}_2$ in $\mathcal{F}_2$, achieves 90% accuracy.** In both families, by construction, the following properties hold

$$p_{1,\rho}(\mathbf{y}) = p_{2,\rho}(\mathbf{y}) \qquad p_{1,\rho}(\mathbf{z}, \mathbf{y}) = p_{2,\rho}(\mathbf{z}, \mathbf{y}), \qquad p_{1,\rho}(\mathbf{x}^*, \mathbf{y}) = p_{2,\rho}(\mathbf{x}^*, \mathbf{y}), \qquad \mathbf{x}_1 \perp\!\!\!\perp_{p_{\cdot,\rho}} \mathbf{x}_2 \mid \mathbf{y}.$$

If $\rho \neq 0.9$, due to the flipping of the positions of $\mathbf{x}^*, \mathbf{z}$ between $p_{1,\rho}$ and $p_{2,\rho}$,

$$p_{1,\rho}(\mathbf{x}_1 \mid \mathbf{y}) \neq p_{2,\rho}(\mathbf{x}_1 \mid \mathbf{y}) \qquad p_{1,\rho}(\mathbf{x}_2 \mid \mathbf{y}) \neq p_{2,\rho}(\mathbf{x}_2 \mid \mathbf{y}).$$

But when $\rho = 0.9$, the distributions are the same: $p_{\cdot,\rho}(\mathbf{x}_1 \mid \mathbf{y}) \overset{\mathrm{d}}{=} p_{\cdot,\rho}(\mathbf{x}_2 \mid \mathbf{y}) \implies p_{1,0.9}(\mathbf{y}, \mathbf{x}) =$

$p_{2,0.9}(\mathbf{y}, \mathbf{x})$. With this we let the training data come from $p_{tr} = p_{1,0.9}$.

Reducing accuracy computation to summing conditional probabilities. Now, we express the accuracy of any predictor $f(x_1, x_2) \in \{-1, 1\}$ of $p_{1,\rho}$:

$$\begin{aligned}
\text{ACC}_f(p_{1,\rho}) &= \mathbb{E}_{p_{1,\rho}(\mathbf{y},\mathbf{x}_1,\mathbf{x})}\mathbf{1}[\mathbf{y} = f(\mathbf{x}_1, \mathbf{x}_2)] \\
&= \sum_{x_1,x_2} p_{1,\rho}(\mathbf{y} = f(x_1, x_2), \mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2) \\
&= \sum_{x_1,x_2} p_{1,\rho}(\mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2 \mid \mathbf{y} = f(x_1, x_2))p_{1,\rho}(\mathbf{y} = f(x_1, x_2)) \\
&= 0.5 \sum_{x_1,x_2} p_{1,\rho}(\mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2 \mid \mathbf{y} = f(x_1, x_2)) \qquad \text{(B.2)}
\end{aligned}$$

With this expression, we have reduced computing the accuracy of a model $f(x_1, x_2)$ to taking one from a pair of numbers — either $p_{1,\rho}(\mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2 \mid \mathbf{y} = 1)$ or $p_{1,\rho}(\mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2 \mid \mathbf{y} = -1)$ based on what $f(x_1, x_2)$ predicts — for each possible value of $x_1, x_1 \in \{-1, 1\}^2$, summing them and multiplying by 0.5.

Showing only a semantic predictor can achieve better accuracy than random chance on $\mathcal{F}_1$. Next, we will show that the only way to achieve better accuracy than random chance on every member of $\mathcal{F}_1$ is to predict with $f(x_1, x_2) = x_1$. To show this, we will express the accuracy computation for two distributions $p_{1,0}$ and $p_{1,1}$ by constructing a table of values of $p_{1,\rho}(\mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2 \mid \mathbf{y} = 1)$ and $p_{1,\rho}(\mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2 \mid \mathbf{y} = -1)$ for $\rho = 0$ and $\rho = 1$ separately.

$p_{1,1}$

		$\mathbf{x}_1$	
		−1	+1
$\mathbf{x}_2$	−1	0, 0.9	0, 0.1
	+1	0.1, 0	0.9, 0

$p_{1,0}$

		$\mathbf{x}_1$	
		−1	+1
$\mathbf{x}_2$	−1	0.1, 0	0.9, 0
	+1	0, 0.9	0, 0.1

(x_1, x_2)	$(-1, -1)$	$(-1, 1)$	$(1, -1)$	$(1, 1)$	$\text{ACC}_f(p_{1,0})$ acc	$\text{ACC}_f(p_{1,1})$	min
0	1	1	1	1	0.50	0.50	0.50
1	1	1	1	-1	0.55	0.05	0.05
2	1	1	-1	1	0.05	0.55	0.05
3	1	1	-1	-1	0.10	0.10	0.10
4	1	-1	1	1	0.95	0.45	0.45
5	1	-1	1	-1	1.00	0.00	0.00
6	1	-1	-1	1	0.50	0.50	0.50
7	1	-1	-1	-1	0.55	0.05	0.05
8	-1	1	1	1	0.45	0.95	0.45
9	-1	1	1	-1	0.50	0.50	0.50
10	-1	1	-1	1	0.00	1.00	0.00
11	-1	1	-1	-1	0.05	0.55	0.05
$\Longrightarrow$ 12	-1	-1	1	1	0.90	0.90	**0.90**
13	-1	-1	1	-1	0.95	0.45	0.45
14	-1	-1	-1	1	0.45	0.95	0.45
15	-1	-1	-1	-1	0.50	0.50	0.50

Table B.1: The 16 different functions that are possible when predicting a label in $\{-1, 1\}$ from $\mathbf{x} \in \{-1, 1\}^2$. We compute the accuracies on $p_{1,0}, p_{1,1}$ and report the minimum of the two. The only predictor that achieves better than random chance accuracy (denoted by $\Longrightarrow$) is $f(x_1, x_2) = x_1$.

By definition of accuracy from eq. (B.2), the accuracy of any predictor $f(x_1, x_2)$ comes down to picking one from the pair of numbers — left one if prediction if 1 and right otherwise — from each element in the table, summing them and multiplying by 0.5. There are 16 possible functions (2 possible predictions each for 4 combinations of x_1, x_2) and we enumerate them in table B.1, showing that only $f^*(x_1, x_2) = x_1$ will perform better than chance on both distributions $p_{1,0}$ and $p_{1,1}$.

No predictor can achieve better accuracy than random on both $\mathcal{F}_1$ and $\mathcal{F}_2$. The earlier parts showed that the only predictor that achieves better accuracy than random chance on all of $\mathcal{F}_1$ is one that only relies on $\mathbf{x}_1$, which equals the semantic feature $\mathbf{x}^*$ under $p_{1,\rho}$. However, under $p_{2,\rho}$, $\mathbf{x}_1$ is the nuisance $\mathbf{z}$. Then, the predictor $f^*(x_1, x_2) = x_1$ has zero accuracy under $p_{2,0}$ because under $p_{2,0}$, we have $\mathbf{z} \sim R_0(\mathbf{y})$ which means $\mathbf{z} \neq \mathbf{y}$ with probability one:

$$\mathrm{ACC}_{f^*}(p_{2,0}) = \sum_{x_1,x_2} p_{2,0}(\mathbf{y} = f(x_1, x_2), \mathbf{x}_1 = x_1, \mathbf{x}_2 = x_2) = \sum_{x_1,x_2} p_{2,0}(\mathbf{y} = x_1, \mathbf{z} = x_1, \mathbf{x}_2 = x_2) = 0 \tag{B.3}$$

□

B.1.1 Semantic corruptions, biased models, and proof of proposition 1

We give the definition of a semantic corruption here and discuss how it implies alternative intuitive definitions before presenting the proof of proposition 1 on using corruptions to build biased models.

Definition 5 (Semantic Corruption). *A semantic corruption is a transformation of the covariates $T(\mathbf{x}, \boldsymbol{\delta})$, where $\boldsymbol{\delta}$ is a random variable such that $\boldsymbol{\delta} \perp\!\!\!\perp (\mathbf{y}, \mathbf{z}, \mathbf{x}, \mathbf{x}^*)$, if*

$$\forall\, p_D \in \mathcal{F} \quad T(\mathbf{x}, \boldsymbol{\delta}) \perp\!\!\!\perp_{p_D} \mathbf{x}^* \mid \mathbf{z}.$$

Two other plausible definitions that come to mind are $T(\mathbf{x}, \boldsymbol{\delta}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x}^*$ and that $\mathbf{y} \perp\!\!\!\perp_{p_D} T(\mathbf{x}, \boldsymbol{\delta}) \mid \mathbf{z}$. These are both intuitive properties that can be asked of a semantic corruption that is supposed to discards all information about semantics, provided that the $\mathbf{z}$ which we wish to retain holds no information on it (which is the case under $p_{\perp\!\!\!\perp}$). We now show that definition 5 implies these two.

From the definition that if $T(\mathbf{x}, \boldsymbol{\delta})$ is a semantic corruption, then it also holds that $T(\mathbf{x}, \boldsymbol{\delta}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x}^*$:

since $\mathbf{x}^* \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z}$

$$p_{\perp\!\!\!\perp}(T(\mathbf{x}, \boldsymbol{\delta}), \mathbf{x}^*) = \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{z})} p_{\perp\!\!\!\perp}(T(\mathbf{x}, \boldsymbol{\delta}), \mathbf{x}^* \mid \mathbf{z}) = \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{z})} p_{\perp\!\!\!\perp}(T(\mathbf{x}, \boldsymbol{\delta}) \mid \mathbf{z}) p_{\perp\!\!\!\perp}(\mathbf{x}^* \mid \mathbf{z}) \tag{B.4}$$

$$= p_{\perp\!\!\!\perp}(\mathbf{x}^*) \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{z})} p_{\perp\!\!\!\perp}(T(\mathbf{x}, \boldsymbol{\delta}) \mid \mathbf{z}) = p_{\perp\!\!\!\perp}(\mathbf{x}^*) p_{\perp\!\!\!\perp}(T(\mathbf{x}, \boldsymbol{\delta})). \tag{B.5}$$

A semantic corruption satisfies the second definition also because

$$\begin{aligned} p_D(\mathbf{y}|T(\mathbf{x}), \mathbf{z}) &= \int p_D(\mathbf{y}|\mathbf{x}^*, T(\mathbf{x}), \mathbf{z}) p_D(\mathbf{x}^*|\mathbf{z}, T(\mathbf{x})) d\mathbf{x}^* = \int p_D(\mathbf{y}|\mathbf{x}^*, \mathbf{z}) p_D(\mathbf{x}^*|\mathbf{z}, T(\mathbf{x})) d\mathbf{x}^* \\ &= \int p_D(\mathbf{y}|\mathbf{x}^*, \mathbf{z}) p_D(\mathbf{x}^*|\mathbf{z}) d\mathbf{x}^* = p_D(\mathbf{y}|\mathbf{z}) \end{aligned} \tag{B.6}$$

First transition adds in integration over the values of $\mathbf{x}^*$, second one uses the property of the nuisance varying family that $\mathbf{x} \perp\!\!\!\perp_{p_D} \mathbf{y}|\mathbf{z}, \mathbf{x}^*$ and therefore it is also conditionally independent for any $T(\mathbf{x}, \boldsymbol{\delta})$. Then the third transition is due to $T(\mathbf{x}, \boldsymbol{\delta})$ being a semantic corruption. The next result shows that the more our semantic corruption captures information about the nuisance that is relevant to predicting $\mathbf{y}$, the better we can approximate learning under $p_{\perp\!\!\!\perp}$, which would yield the optimal risk-invariant predictor over $\mathcal{F}$ [18].

B.1.1.1 Proof of proposition 1.

Now, using the property in eq. (B.6) that holds for semantic corruptions, we prove proposition 1.

Proposition 1. *Let $T : \mathbf{X} \times \mathbf{R}^d \to \mathbf{X}$ be a function. Assume the r.v. $p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))^{-1}$ has a bounded second moment under the distribution $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) p(\boldsymbol{\delta})$, and that $p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))$ and $p_{tr}(\mathbf{y} \mid \mathbf{z})$ satisfy*

$$\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})p(\boldsymbol{\delta})} p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))^{-2} \leq m^2, \qquad \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})p(\boldsymbol{\delta})} \left|p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z})\right|^2 = \epsilon^2.$$

Then, the L_1 distance between $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x})$ and $p_T(\mathbf{y}, \mathbf{x})$ is bounded: $\|p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x}) - p_T(\mathbf{y}, \mathbf{x})\|_1 \leq m\epsilon$. For a semantic corruption that also satisfies $\mathbf{y} \perp\!\!\!\perp_{p_{tr}} \mathbf{z} \mid T(\mathbf{x}, \boldsymbol{\delta})$ the inequalities hold with $\epsilon = 0$.

Proof. The L_1 distance between the distributions is bounded from above by a $p_{\perp\!\!\!\perp}$-weighted L_1 distance between $p_{tr}(\mathbf{y} \mid \mathbf{z})$ and $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$, upto a constant:

$$\int_{y,x} |p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{x}) \; - \; p_T(\mathbf{y}, \mathbf{x}))| \, dydx \tag{B.7}$$

$$= \int_{y,x} \left| \int_z p_{tr}(\mathbf{y}) p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x}) p(\boldsymbol{\delta}) \left[\frac{1}{p_{tr}(\mathbf{y} \mid \mathbf{z})} - \frac{1}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))} \right] dz \right| dydx \tag{B.8}$$

$$= \int_{y,x} \left| \int_z p_{tr}(\mathbf{y}) p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x}) p(\boldsymbol{\delta}) \left[\frac{p_{tr}(\mathbf{y} \mid T(\mathbf{x})) - p_{tr}(\mathbf{y} \mid \mathbf{z})}{p_{tr}(\mathbf{y} \mid \mathbf{z}) p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))} - \right] dz \right| dydx \tag{B.9}$$

$$= \int_{y,x} \left| \mathbb{E}_{p_{tr}(\mathbf{z})p(\boldsymbol{\delta})} \frac{p_{tr}(\mathbf{y})}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))} p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \left[p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z}) \right] \right| dydx \tag{B.10}$$

$$\leq \int_{y,x} \mathbb{E}_{p_{tr}(\mathbf{z})p(\boldsymbol{\delta})} \left| \frac{p_{tr}(\mathbf{y})}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))} p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \left[p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z}) \right] \right| dydx \tag{B.11}$$

$$= \int_{y,x,z} p_{tr}(\mathbf{z}) p_{tr}(\mathbf{y}) p(\boldsymbol{\delta}) p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \frac{1}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))} \left| p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z}) \right| dydxdz \tag{B.12}$$

$$= \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})p(\boldsymbol{\delta})} \frac{1}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))} \left| p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z}) \right| \tag{B.13}$$

$$\leq \left(\sqrt{ \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{x})p(\boldsymbol{\delta})} \frac{1}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}))^2} } \right) \sqrt{ \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})p(\boldsymbol{\delta})} \left| p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z}) \right|^2 } \tag{B.14}$$

Substituting the bounds from the theorem statement completes the proof of the bound.

Finally, if T is a semantic corruption, by eq. (B.6), it holds that

$$p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}), \mathbf{z}) = p_{tr}(\mathbf{y} \mid \mathbf{z}).$$

Then, if it also holds that $\mathbf{y} \perp\!\!\!\perp_{p_{tr}} \mathbf{z} \mid T(\mathbf{x}, \boldsymbol{\delta})$, it holds that

$$p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta}), \mathbf{z}) = p_{tr}(\mathbf{y} \mid T(\mathbf{x}, \boldsymbol{\delta})).$$

Together this implies that almost everywhere in $p_{tr}(\mathbf{y},\mathbf{z},\mathbf{x})p(\boldsymbol{\delta})$

$$p_{tr}(\mathbf{y} \mid T(\mathbf{x},\boldsymbol{\delta})) = p_{tr}(\mathbf{y} \mid \mathbf{z}) \implies \mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x})p(\boldsymbol{\delta})} \left|p_{tr}(\mathbf{y} \mid T(\mathbf{x},\boldsymbol{\delta})) - p_{tr}(\mathbf{y} \mid \mathbf{z})\right|^2 = 0.$$

This shows that for a semantic corruption such that $\mathbf{y} \perp\!\!\!\perp_{p_{tr}} \mathbf{z} \mid T(\mathbf{x},\boldsymbol{\delta})$, it holds that $\epsilon = 0$. □

B.2 Further details about b-scams and related work

NuRD. Focusing on mitigating spurious correlations, Puli et al. [48] identify a conditional that has performance guarantees on every test distribution within a family of distributions with varying nuisance-label relationships: $p_{te} \in \mathcal{F}$. They develop NuRD to learn the conditional using data only from $p_{tr} \neq p_{te}$. NuRD uses 1) the *nuisance-randomized distribution*, $p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x}) = p(\mathbf{y})p_{\perp\!\!\!\perp}(\mathbf{z})p(\mathbf{x} \mid \mathbf{y},\mathbf{z})$, where $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y}$, and 2) an *uncorrelating representation* $r(\mathbf{x})$ for which $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r(\mathbf{x})$. NuRD builds models of the form $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ using $r(\mathbf{x})$ that are most informative of the label.

We run reweighting-NuRD, which uses a biased model $p_{tr}(\mathbf{y} \mid \mathbf{z})$ as an importance weight to compute loss under the nuisance-randomized distribution: $p_{\perp\!\!\!\perp}(\mathbf{y},\mathbf{z},\mathbf{x}) = \frac{p_{tr}(\mathbf{y})}{p_{tr}(\mathbf{y} \mid \mathbf{z})} p_{tr}(\mathbf{y},\mathbf{z},\mathbf{x})$.

To run reweighting-NuRD with semantic corruptions, we replace $p_{tr}(\mathbf{y} \mid \mathbf{z})$ with $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$ for a semantic corruption $T(\mathbf{x})$. Semantic corruptions are noisy functions of $\mathbf{x}$: with noise $\boldsymbol{\epsilon}$ such that $(\mathbf{y},\mathbf{z},\mathbf{x}) \perp\!\!\!\perp_{p_D} \boldsymbol{\epsilon}$, $T(\mathbf{x}) = U(\mathbf{x},\boldsymbol{\epsilon})$. This implies

$$\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \boldsymbol{\epsilon} \mid \mathbf{x} \implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x}, \boldsymbol{\epsilon} \mid \mathbf{x} \implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} T(\mathbf{x}) \mid \mathbf{x}$$

Thus, $r(\mathbf{x}) = \mathbf{x}$ is uncorrelating and $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ achieves the optimality guarantees in Puli et al. [48]. These optimality guarantees imply that regardless of the test nuisance-label relationship, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ will achieve optimal performance within the class of models like $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$.

End-to-end bias mitigation. Mahabadi et al. [47] consider two methods to train a *biased* model $p_{tr}(\mathbf{y} \mid \mathbf{z})$ and a base predictive model jointly to make the base model predict without relying on the biases. The methods use and fine-tune a BERT model [55] and do not propagate the gradients of the biased model to update the common parameters (token embeddings in this case). They propose 1) poe, where the log of the product of the predictions (the output probabilities) of the two models is used to compute the classification loss and 2) dfl, where the biased model is used to weight the cross-entropy loss for the base model.

The intuition for poe is that the samples for which the biased model classifies correctly will not contribute to the gradients of the base model; thus the base model focuses more on classifying samples that the biased model misclassifies. The dfl algorithm weights each sample as the biased model's predicted probability of all but the label, exponentiated with $\gamma > 0$. This downweights samples that the biased model classifies correctly which in turn mitigates the base model's reliance on a nuisance which only helps predict the downweighted samples correctly.

Formally, with a biased model $f_\theta(\mathbf{z})$ and a predictive model $f_\gamma(\mathbf{x})$ that output a vector of logits over classes, σ denoting the soft-max function that maps logits to class-probabilities, and $\sigma(\cdot)_y$ denoting the softmax-probability of label y

$$\textsc{poe} \quad \max_{\theta,\gamma} \sum_{i \in \texttt{training data}} \log \sigma(f_\theta(\mathbf{z}_i))_{y_i} + \log \sigma(f_\gamma(\mathbf{x}_i))_{y_i} \tag{B.15}$$

$$\textsc{dfl} \quad \max_{\theta,\gamma} \sum_{i \in \texttt{training data}} \left(1 - \sigma(f_\theta(\mathbf{z}_i))_{y_i}\right)^\gamma \log \sigma(f_\gamma(\mathbf{x}_i))_{y_i} \tag{B.16}$$

Mahabadi et al. [47] build the biased model f_θ using known nuisances $\mathbf{z}$. We build this model from a semantic corruption $T(\mathbf{x})$.

Just Train Twice (JTT). jtt works in two stages: 1) build an "identification" model via erm on the training data to isolate samples that are misclassified due to reliance on the nuisances and 2) train a model via ERM on data with the loss for the misclassified samples upweighted (by constant λ). The identification model in JTT is built to be a biased model. When the identification model equals $p_{tr}(\mathbf{y} \mid \mathbf{z})$, it exactly misclassifies the samples in the groups in the minority group[*]. Upweighting these samples produces a dataset with lesser dependence between the nuisance and the label. Models learned on the upweighted data depend more on the semantics. See algorithm 3 for pseudocode.

Algorithm 3: JTT.

```
Input: Training set $D$ and hyperparameters $T$ and $\lambda_{\text{up}}$. Stage one: identification
1. Train identification model $f_\theta$ on $D$ via ERM for $T$ steps.
2. Construct the errors set of training examples misclassified by $f_\theta$.
Stage two: upweighting identified points
3. Construct upsampled dataset $D_{\text{up}}$ containing examples in the error set repeated $\lambda_{\text{up}}$ times
and all other examples once.
4. Train final model $f_\gamma$ on $D_{\text{up}}$ via ERM.
```

In this work, we build the identification model on semantic corruptions i.e. we learn f_θ to predict $\mathbf{y}$ from $T(\mathbf{x})$. The training samples to be upweighted are the ones misclassified when predicting with the identification model on semantic-corrupted versions of the sample, i.e. $T(\mathbf{x})$. The second stage is run as in [53] with training data.

Optimization-generalization Dilemma Like many other algorithms in the OOD generalization literature, training B-SCAMSs based on semantic corruptions may also suffer from obstacles due to optimization and generalization: employing statistical constraints to handle distribution shift may not build models that perform well OOD due to overfitting [87], training difficulties [239, 240, 241], or reliance on inappropriate inductive biases [86, 242]. Some approaches in the literature can alleviate these difficulties: two-stage methods incorporate the OOD objective only when training smaller models on top of large ones [239, 240, 241, 243, 244], subsampling instead

[*] The minority group is the set of samples that the nuisance misclassifies. For example, when $p_{tr}(\mathbf{y} = \mathbf{z}) > p_{tr}(\mathbf{y} \neq \mathbf{z})$, then the minority group is the set of samples with $\mathbf{y} \neq \mathbf{z}$ because using only the nuisance results in predicting $\mathbf{y} = b$ where $\mathbf{z} = b$.

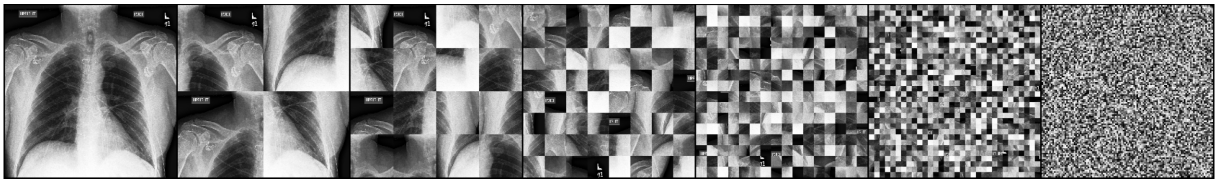


Figure B.1: Example of PR of a chest X-ray image. The image is followed by PRs of size 112, 56, 28, 14, 7, 2.

of weighting [7, 245], or large ℓ_2 regularization [16].

In our implementations we use validation data and regularization to tune parameters for the weighted-ERM algorithm as proposed in the original papers of the B-SCAMS we experiment with. As ERM is standard practice, there are no new optimization difficulties but generalization difficulties can occur due to overfitting [87, 242]. Any improvements in generalization in weighted-ERM will lead to improvements in models built by B-SCAMS with biased models from semantic corruptions.

B.3 Further experimental details

B.3.1 Remark on baseline corruptions

NuRD with the baseline corruption GAUSS-NOISE outperforms ERM and closes 80% of the gap between ERM and known-**z** NuRD in table 3.2. We explain such an improvement as a consequence of GAUSS-NOISE corrupting semantics more than it corrupts nuisances; we explain below. In tasks like waterbirds, nuisances are present in most if not all patches of the image regardless of where the patches appear. On the other hand, semantic features are localized to a few adjacent patches (like the birds parts appearing next to each other). When nuisances are present is many more patches than the semantics, adding gaussian noise to all pixels corrupts semantics more than nuisances. To see why, consider meausurements of a quantity as a gaussian random variable with the quantity as its mean. More measurements lead to better estimates of the mean.

B.3.2 Implementation details

Each experiment in the paper was run on up to 2 RTX8000 GPUs. The hyperparameters for methods that use known nuisances in the training data, like NuRD, POE, DFL are tuned on validation data from the training distribution. For NuRD, we select corruption hyperparameters using the mean of the balanced validation accuracy across 10 seeds. We do the same when using semantic corruptions.

Experimental details for Waterbirds For the NuRD setup, the training, validation, and test datasets have 3020, 756, 800 samples respectively. We use a single architecture to parameterize the predictive model and the weight model in this experiment: two fully connected layers on top of a ResNet18 initialized at weights pretrained on Imagenet. We use the same training procedure for NuRD with known nuisances or with semantic corruptions. Both models are trained with cross-entropy. The weight model is optimized with the default Adam optimizer for 20 epochs with a batch size of 64. The predictive model is optimized with the Adam optimizer for 20 epochs with a learning rate of 0.0002, a weight decay of 0.01, and a batch size of 250.

For the JTT setup, the training, validation, and test datasets have 4795, 1199, 5794 samples respectively. For JTT, we use the same model and model parameters as Liu et al. [53] using their released code. We repeat the details here for completeness. The model for both stages of JTT is a ResNet-50. Both models are optimized by stochastic gradient descent (SGD) with momentum 0.9, weight decay 1.0, and learning rate 1×10^{-5}. Both models are trained for 300 epochs with batch size 64, using batch normalization and no data augmentation. The identification model used to select samples to upweight corresponds to epoch 60 and the upweighting constant is $\lambda = 100$.

Experimental details for cardiomegaly detection. The training, validation, and test datasets are fixed across seeds and have 18000, 2000, 1000 samples respectively. To run reweighting-NuRD, we use a single architecture to parameterize the predictive model and the weight model in this experiment: two fully connected layers on top of a ResNet18 initialized at weights pretrained on Imagenet. In known-nuisance NuRD with the hospital as the nuisance, the biased model is an estimate of $p_{tr}(\mathbf{y} \mid \text{hospital})$, which is obtained by binning the samples based on the hospital and averaging the labels. We use the same training procedure for NuRD with known nuisances or with semantic corruptions. Both weight and predictive models are trained with cross-entropy. The weight model and the predictive model are optimized with the Adam optimizer over 25 epochs with a batch size of 256, and learning rate 0.001.

Implementation details for nli For poe and dfl, we build classifiers by fine-tuning a pretrained BERT model [55] on the data. We follow the same training procedure and hyperparameter details as used in Mahabadi et al. [47] — models were trained on the MNLI training dataset which consists of 392k examples, with a learning rate of 2×10^{-5} with a batch size of 8 using the Adam Optimizer. All models are trained for 3 epochs. The development set contains 9815 examples and the HANS test contains 30000 examples. Since the HANS dataset has only two labels — 'entailment' and 'non-entailment' — we combine the neutral and contradiction classes during inference on HANS.

For the jtt setup, Liu et al. [53] mix the training and development sets from MNLI and create their own training, validation, and test sets of sizes 206175, 82462, 123712 respectively. For jtt, we use the same model and model parameters as Liu et al. [53] using their released code. We use the optimal hyperparameters reported in [53] for the learning rate, weight decay, and the upweighting constant. We repeat the details here for completeness. The model for both stages of jtt is a pretrained BERT model that is finetuned during training. Both models are optimized by the AdamW optimizer with clipping for the predictive model, no weight decay, and an initial learning rate of 2×10^{-5}. Both models are trained for 5 epochs with batch size 32 and dropout. The identification model used to select samples to upweight corresponds to epoch 2 for vanilla jtt (reported optimal in Liu et al. [53]); for jtt with semantic corruption, we select one from 2, 3 using validation group annotations. For both, the upweighting constant is $\lambda = 6$. Our runs with these parameters did not yield the test worst-group accuracy reported in [53] (72.6%); our experiments yielded a test worst-group accuracy 71.3%. We expect this may be due to the differences in the random seed; JTT is sensitive to hyperparameters and differences in order of batches may result in drops in performance.

In NR, when the number of words in the sentence is not a multiple of n, there will be one k-gram ($k < n$). In implementing NR, we ensure that the position of this k-gram is randomized i.e. we make sure that it does not always occur at the end of the sentence, for example. NR is implemented before word-piece tokenization (which BERT uses), to ensure that we randomize words instead of subwords. We also create a small HANS-like development set, which is used to tune the size parameter. This set is constructed by randomly sampling 1000 examples from the HANS training set, which has zero overlap with the HANS test set.

B.3.3 Full results tables and additional experiments

We give the results for all size parameters; see table B.4, table B.5, table B.6, table B.7, and table B.8. To report the same metrics as in [47] for POE and DFL and [48] for NuRD, we report standard error for NuRD and standard deviation for POE and DFL .

B.3.3.1 Results on Adversarial NLI [64] and CAD [65]

In table B.9 and table B.10, we report evaluations of POE and DFL models on the adversarial ANLI [64] and the counterfactually augmented dataset [65].

B.3.3.2 Additional experiments

Experiments with weaker spurious correlations. To verify the effectiveness of the semantic corruptions for powering B-SCAMS like JTT that rely on assumptions on ERM-trained models, we experiment with a modified version of the Waterbirds dataset. In the modified dataset, the spurious feature predicts the label only 75% of the time; this is weaker than the 93% in the original dataset and the invariant relationship which achieves $>$ 85% accuracy across all groups. We ran ERM, JTT, and corruption-powered JTT. For both versions of JTT, we tune over the same hyperparameters as in Liu et al. [53].

The results in table B.2 show that corruption-powered JTT is better than vanilla JTT and ERM. The improvement of corruption-powered JTT over vanilla JTT increases from 0.5% in table 3.3 to 4.4% in table B.2; this indicates that vanilla JTT is more sensitive to the strength of the spurious correlation than corruption-powered JTT.

Table B.2: Test worst-group (WG) accuracies of JTT on modified waterbirds where the spurious correlation is weaker than the invariant relationship. Corruption-powered JTT outperforms ERM, vanilla JTT, and JTT with baseline corruptions (RAND-CROP, GAUSS-NOISE) by $\geq$ 4.4%.

Method	test WG acc.
Vanilla JTT	78.6%
PR	84.6%
ROI-MASK	85.2%
FREQ-FILT	83.2%
INT-FILT	83.0%
RAND-CROP	76.2%
GAUSS-NOISE	75.9%
ERM	76.1%

Experiments with multiple spurious features. We run ROI-MASK-powered NuRD with a modified version of the ColorFulMNIST dataset [243]. The images consist of $42 \times 42 \times 3$ pixels, with the middle 14×14 forming the MNIST image showing a 0 or a 1 and the rest being background patches. The digit in the middle predicts the binary label 1 or 0 with 75% accuracy. Given some $p \in [0, 1]$, this dataset sets each of the background patch colors deterministically based on the image in the middle with probability p; with probability $1 - p$, each background is a random color (see figure 5 in [243].) We generate the training data with $p = 0.9$, and the validation and test data with $p = 0$.

Roi-mask-powered NuRD with central-roi sizes 14 and 28 achieves test accuracies 71.1% and 70.3% respectively, beating erm which achieves 51.7% because it relies more on the background colors. pr is not suited for this experiment because the different nuisance colors are chosen based on the patch position, and pr randomizes patch positions which corrupt these nuisances.

Experiments showing that corrupting the semantics is the reason behind the improved ood performance in corruption-powered b-scams. First, we show that corruptions actually do corrupt semantics, taking pr as the example. We focus on the Waterbirds dataset to show how patch size affects semantics. For this investigation, we construct training and test datasets where the label and nuisance are independent and build models for predicting the label.

Table B.3: Accuracy of predicting the label from the image corrupted by pr as patch-size decreases. As the label is independent of the nuisance, a lower accuracy means that more semantic information is corrupted.

pr size	Accuracy
Full image	86%
112	76%
56	73%
28	64%
14	58%
7	57%

The results are in table B.3 and show that as patch-size decreases, more semantic information is lost. These results mean that for patch sizes < 28, a biased model built from the corrupted image cannot predict the label well using semantics alone; the accuracy of random chance is 50%. As the label is independent of the nuisance, a lower accuracy means more semantic information is corrupted. However, on the original dataset, our biased models at these patch sizes achieve at least 85% accuracy in predicting the label from the corrupted images, meaning that they rely mostly on the nuisance.

Second, to show that corruptions actually do help, we ran the full NuRD algorithm on the Waterbirds dataset from [48] with a biased model built directly on the uncorrupted covariates; that is we train a model with erm to predict $\mathbf{y}$ from $\mathbf{x}$ and use it as the biased model in NuRD. The resulting test accuracy is $< 70\%$. When using patch-sizes under 28, the pr-powered NuRD algorithm achieves a test accuracy of nearly 87%. This shows that the corruption of semantics is directly responsible for improving model robustness.

Table B.4: Mean and standard error of test accuracy across 10 seeds of NuRD on classifying waterbirds. *Known*-nuisance NuRD uses a label for the type of background as the nuisance. Selecting the size hyperparameter based on the average accuracy over 10 seeds on the validation dataset gives 14 for PR, 196 for ROI-MASK, 168 for FREQ-FILT, and 0.2 for INT-FILT. Consider the gap between ERM and known-nuisance NuRD. NuRD with PR, ROI-MASK, FREQ-FILT, and INT-FILT close 99%, 99%, 82%, 99% of the gap respectively. NuRD with these semantic corruptions outperforms ERM and NuRD with RAND-CROP and GAUSS-NOISE. NuRD with all semantic corruptions outperforms ERM (69.2%).

	known	RM	RM	RM	RM	PR	PR	PR	PR	
	Z	196	168	140	112	7	14	28	56	ERM
Mean	87.2%	86.9%	86.6%	86.2%	86.3%	85.6%	86.9%	82.5%	84.9%	68.0%
Std. err.	1.0%	1.1%	1.2%	1.8%	1.6%	1.4%	1.2%	2.0%	1.4%	1.9%
		FF	FF	FF	FF	IF	IF	IF	IF	
		196	168	140	112	0.1	0.2	0.3	0.4	
Mean		83.8%	83.5%	81.0%	80.3%	81.2%	86.9%	85.0%	81.9%	
Std. err.		1.2%	1.1%	1.4%	1.7%	1.7%	1.1%	1.5%	1.7%	
		RAND-CROP				GAUSS	GAUSS	GAUSS	GAUSS	
						0.01	0.25	1	4	
Mean		73.7%				75.8%	74.1%	78.0%	83.9%	
Std. err.		2.0%				3.2%	3.1%	3.4%	1.4%	

Table B.5: Average accuracies and standard deviation over 4 seeds of POE and DFL with semantic corruptions on the HANS dataset. The results for *known* POE and DFL from [47], where both methods use known nuisances. For both methods, selecting the size hyperparameter based on the average accuracy on a small dataset (1000 samples) from the test distribution gives $n = 3$. With this size, POE with NR performs better than known-nuisance POE while DFL with NR closes 84% of the gap between ERM and known-Z DFL .

Z	POE	DFL
Known	66.3 ± 0.6%	69.3 ± 0.2%
1-gram	65.7 ± 2.0%	66.5 ± 1.5%
2-gram	66.0 ± 0.9%	68.5 ± 0.7%
3-gram	66.7 ± 1.5%	68.4 ± 1.5%
4-gram	66.2 ± 2.9%	65.0 ± 2.0%
ERM	–	63.6%.

Table B.6: Mean and standard error of test accuracy across 10 seeds of NuRD on detecting cardiomegaly from chest X-rays. *Known*-nuisance NuRD uses the hospital as the nuisance. Selecting the corruption parameters based on the mean accuracy over 10 seeds on the validation dataset gives 14 for pr, 196 for roi-mask, 168 for freq-filt, and 0.1 for the int-filt. Consider the gap between erm and known-nuisance NuRD. NuRD with pr, roi-mask, freq-filt, and int-filt close 72%, 82%, 65%, 35% of the gap respectively. NuRD with semantic corruptions outperforms NuRD with baseline augmentations rand-crop and gauss-noise. NuRD with pr and roi-mask outperforms erm for all size parameters.

	known	rm	rm	rm	rm	pr	pr	pr	pr	
	z	196	168	140	112	7	14	28	56	erm
Mean	81.7%	78.7%	78.3%	77.2%	73.6%	76.2%	77.0%	74.9%	74.3%	65.3%
Std. err.	0.3%	0.3%	0.8%	0.8%	0.7%	1.2%	1.2%	1.0%	1.4%	1.1%
		ff	ff	ff	ff	if	if	if	if	
		196	168	140	112	0.1	0.2	0.3	0.4	
Mean		74.4%	76.0%	75.3%	71.3%	71.0%	68.0%	62.0%	57.1%	
Std. err.		1.5%	0.6%	0.9%	1.6%	1.0%	1.6%	1.8%	3.2%	
		rand-crop				gauss	gauss	gauss	gauss	
						0.01	0.25	1	4	
Mean		59.9%				62.3%	63.5%	68.0%	69.0%	
Std. err.		2.1%				3.7%	3.4%	1.1%	1.9%	

Table B.7: Test worst-group accuracies of JTT with semantic corruptions on waterbirds. Selecting the corruption hyperparameters on the validation worst-group accuracy gives size 14 for PR, size 196 for ROI-MASK, size 112 for FREQ-FILT, and threshold 0.4 for INT-FILT. JTT with these semantic corruptions outperforms ERM, vanilla JTT, and JTT with the baseline corruptions RAND-CROP and GAUSS-NOISE. JTT with PR and ROI-MASK outperforms JTT with the baseline corruptions and ERM for all sizes.

Vanilla	RM	RM	RM	RM	PR	PR	PR	PR	
JTT	196	168	140	112	7	14	28	56	ERM
86.5%	88.2%	88.0%	86.9%	86.2%	89.3%	89.0%	88.9%	89.1%	72%
	FF	FF	FF	FF	IF	IF	IF	IF	
	196	168	140	112	0.1	0.2	0.3	0.4	
	82.5%	84.5%	85.2%	87.2%	69.1%	80.0%	81.7%	87.0%	
	RAND-CROP				GAUSS	GAUSS	GAUSS	GAUSS	
					0.01	0.25	1	4	
	75%				0.0%	0.0%	71.0%	0.0%	

Table B.8: Worst-group and average test accuracies of JTT with semantic corruptions on NLI. JTT with PREM-MASK and NR of every size outperforms vanilla JTT. Selecting the size hyperparameter for NR using validation worst-group accuracy, like Liu et al. [53] do for vanilla JTT, gives $n = 1$. At this size, JTT with NR outperforms vanilla JTT by 3% accuracy.

	Worst-group	Average
Vanilla JTT	71.3%	79.1%
PREM-MASK	72.1%	79.9%
1-gram	74.3%	79.7%
2-gram	71.9%	80.0%
3-gram	72.0%	80.1%
4-gram	73.4%	80.4%
ERM	67.9%	–

Table B.9: ANLI [64] evaluations of models trained on MultiNLI. With a t-test to measure statistical significance, at the standard significance level of 0.05, we found that POE with NR gave a statistically significant improvement over the baseline on ANLI-R1 and ANLI-R2, while DFL gave a statistically significant improvement on ANLI-R1.

Model	ANLI - R1	ANLI - R2	ANLI - R3
ERM	23.1 ± 0.9	28.2 ± 0.8	29.8 ± 0.4
POE-known	23.5 ± 0.6	27.8 ± 0.8	29.8 ± 0.8
DFL-known	23.7 ± 1.3	27.8 ± 1.1	30.4 ± 0.9
POE - n3	24.8 ± 1.1	29.2 ± 0.4	30.4 ± 1.2
DFL - n3	24.9 ± 0.6	29.0 ± 1.2	29.9 ± 0.3
POE - PREM-MASK	23.6 ± 1.2	27.3 ± 0.8	29.8 ± 0.8
DFL - PREM-MASK	22.3 ± 0.7	27.7 ± 0.6	29.3 ± 1.1

Table B.10: Mean and standard deviation of CAD [65] test accuracy over 4 seeds. At the end, we also report the results of finetuning BERT on CAD training data from [65]. When trained on MNLI, on average over the CAD subsets RH and RH, DFL and POE with semantic corruptions, DFL and POE with known-nuisances, and ERM perform on par (within one std.) or better than finetuning directly on the training CAD dataset. The improvement over finetuning directly on CAD may be due to the fact that the CAD dataset is much smaller than MNLI ($7k$ vs. $400k$).

Method	RP	RH	Avg. on RP and RH
ERM on MNLI	61.1 ± 0.3	76.5 ± 0.4	68.8 ± 0.2
POE-known	60.6 ± 0.5	77.0 ± 1.1	68.8 ± 0.3
POE 3-gram	60.8 ± 0.5	76.1 ± 0.7	68.4 ± 0.2
POE PREM-MASK	61.7 ± 0.6	75.6 ± 1.0	68.6 ± 0.5
DFL-known	60.6 ± 0.8	76.2 ± 0.7	68.4 ± 0.4
DFL 3-gram	58.4 ± 1.8	72.7 ± 1.0	65.5 ± 1.4
DFL PREM-MASK	62.4 ± 0.7	76.1 ± 0.8	69.3 ± 0.6
ERM on CAD (from [65])	64.6	67.8	66.2

C | Appendices for chapter 4

C.1 Proof of Theorem 3, Corollary 1, and Theorem 4

C.1.1 Helper Lemmas

C.1.1.1 Bounding norms and inner products of isotropic random vectors.

The main lemmas of this section are lemma 8 and lemma 9. We will then use these two to bound norms of sums of random vectors and inner products between the sum and a single random vector in lemma 10. We first list some facts from [246] that we will use to bound the probability with which norms and inner products of Gaussian random vectors deviate far from their mean.

Definition 6. *(Sub-Gaussian norm) For an r.v. $\mathbf{x}$, the sub-Gaussian norm, or ψ_2-norm, is*

$$\|\mathbf{x}\|_{\psi_2} = \inf\{t > 0, \mathbb{E}[\exp(\mathrm{x}^2/t^2)] \leq 2\}.$$

An r.v. is called sub-Gaussian if its ψ_2-norm is finite and for some fixed constant c

$$p(|\mathbf{x}| > t) \leq 2\exp(-ct^2/\|\mathbf{x}\|_{\psi_2}).$$

A Gaussian r.v. $\mathbf{x} \sim \mathcal{N}(0, \sigma^2)$ has an ψ_2-norm of $G\sigma$ for a constant $G = \sqrt{\frac{8}{3}}$.[*]

Definition 7. *(Sub-exponential norm) For an r.v.* $\mathbf{x}$*, the sub-exponential norm, or* ψ_1*-norm, is*

$$\|\mathbf{x}\|_{\psi_1} = \inf\{t > 0, \mathbb{E}[\exp({|\mathbf{x}|}/{t})] \leq 2\}.$$

A sub-exponential r.v. is one that has finite ψ_1*-norm.*

Lemma 6. *(Lemma 2.7.7 from [246]) Products of sub-Gaussian random variables* $\mathbf{x}, \mathbf{y}$ *is a sub-exponential random variable with it's* ψ_1*-norm bounded by the product of the* ψ_2*-norm*

$$\|\mathbf{x}\mathbf{y}\|_{\psi_1} \leq \|\mathbf{x}\|_{\psi_2}\|\mathbf{y}\|_{\psi_2}$$

Lemma 6 implies that the product of two mean-zero standard normal vectors is a sub-exponential random variable with ψ_1-norm less than G^2.

Lemma 7. *(Bernstein inequality, Theorem 2.8.2 [246]) For i.i.d sub-exponential random variables*

[*] $G = \sqrt{\frac{8}{3}}$. This follows from:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,\sigma^2)}[\exp({\mathbf{x}^2}/{t^2})] = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp({-x^2}/{2\sigma^2}) \exp({x^2}/{t^2}) dx = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-x^2 \frac{(t^2 - 2\sigma^2)}{2\sigma^2 t^2}\right) dx$$

$$= \frac{1}{\sigma\sqrt{2\pi}} \sqrt{\frac{\pi}{\frac{(t^2-2\sigma^2)}{2\sigma^2 t^2}}} = \frac{1}{\sigma\sqrt{\pi}} \sqrt{\frac{\pi\sigma^2 t^2}{(t^2 - 2\sigma^2)}} = \sqrt{\frac{t^2}{(t^2 - 2\sigma^2)}}$$

$$\sqrt{\frac{t^2}{(t^2 - 2\sigma^2)}} \leq 2 \implies t^2 \leq 4(t^2 - 2\sigma^2) \implies 8\sigma^2 \leq 3t^2 \implies \inf\{t : 8\sigma^2 \leq 3t^2\} = \sqrt{\frac{8}{3}}\sigma.$$

$\mathbf{x}_1, \cdots, \mathbf{x}_d$, *for a fixed constant* $c = \frac{1}{(2e)^2}$ *and* $K = \|\mathbf{x}_1\|_{\psi_1}$

$$p\left(\left|\sum_{i=1}^{d} \mathbf{x}_i\right| > t\right) \leq 2\exp\left(-c\min\left\{\frac{t^2}{K^2 d}, \frac{t}{K}\right\}\right)$$

Next, we apply these facts to bound the sizes of inner products between two unit-variance Gaussian vectors.

Lemma 8. *(Bounds on inner products of Gaussian vectors) Let* $\mathbf{u}, \mathbf{v}$ *be* d*-dimensional random vectors where each coordinate is an i.i.d standard normal r.v. Then, for any scalar* $\epsilon > 0$ *such that* $\epsilon \leq G^2\sqrt{d}$, *for a fixed constant* $c = \frac{1}{(2e)^2}$

$$p\left(\left|\mathbf{u}^\top \mathbf{v}\right| > \epsilon\sqrt{d}\right) \leq 2\exp\left(-c\frac{\epsilon^2}{G^4}\right).$$

Proof. First, the inner product is $\mathbf{u}^\top \mathbf{v} = \sum_i^d \mathbf{u}_i \mathbf{v}_i$; it is the sum of products of i.i.d. standard normal r.v. ($\sigma = 1$). Then, by lemma 6, each term in the sum is a sub-exponential r.v. with ψ_1-norm bounded as follows:

$$K = \|\mathbf{u}_i \mathbf{v}_i\|_{\psi_1} \leq \|\mathbf{u}_i\|_{\psi_2} \|\mathbf{u}_i\|_{\psi_2} = G \times G = G^2. \tag{C.1}$$

We can apply Bernstein inequality lemma 7 to sub-exponential r.v. to the inner product and then upper bound the probability by replacing K with the larger G^2 in eq. (C.1)

$$p\left(|\mathbf{u}^\top \mathbf{v}| > t\right) \leq 2\exp\left(-c\min\left\{\frac{t^2}{K^2 d}, \frac{t}{K}\right\}\right) \leq 2\exp\left(-c\min\left\{\frac{t^2}{G^4 d}, \frac{t}{G^2}\right\}\right)$$

Substituting $t = \epsilon\sqrt{d}$ in the above gives us:

$$p\left(|\mathbf{u}^\top \mathbf{v}| > \epsilon\sqrt{d}\right) \le 2\exp\left(-c\min\left\{\frac{\epsilon^2 d}{G^4 d}, \frac{\epsilon\sqrt{d}}{G^2}\right\}\right)$$

Using the fact that $\epsilon \le G^2\sqrt{d}$ to achieve the minimum concludes the proof:

$$\epsilon \le G^2\sqrt{d} \implies \epsilon^2 \le \epsilon G^2\sqrt{d} \implies \frac{\epsilon^2}{G^4} \le \frac{\epsilon\sqrt{d}}{G^2} \implies \min\left\{\frac{\epsilon^2 d}{G^4 d}, \frac{\epsilon\sqrt{d}}{G^2}\right\} = \frac{\epsilon^2}{G^4}$$

□

Lemma 9. *Let* $\mathbf{x}$ *be a Gaussian vector of size* d *where each element is a standard normal, meaning that* $\|\mathbf{x}_i\|_{\psi_2} = G$*. Then, for any* $t > 0$ *and a fixed constant* $c = \frac{1}{(2e)^2}$ *, the norm of the vector concentrates around* $\sqrt{d}$ *according to*

$$p\left(\left|\|\mathbf{x}\| - \sqrt{d}\right| > t\right) \le 2\exp({}^{-ct^2}/_{G^4}).$$

Proof. Equation 3.3 from the proof of theorem 3.1.1 in [246] shows that

$$p(|\|\mathbf{x}\| - \sqrt{d}| > t) \le 2\exp({}^{-ct^2}/_{(\max_i \|\mathbf{x}_i\|_{\psi_2})^4}).$$

As $\mathbf{x}$ has i.i.d standard normal entries, $\max_i \|\mathbf{x}_i\|_{\psi_2} = G$, concluding the proof. □

C.1.1.2 Concentration of norms of sums of random vectors and their inner products

This is the main lemma that we will use in proving theorem 3.

Lemma 10. *Consider a set of vectors* $V = \{\boldsymbol{\delta}_i\}$ *where* $\boldsymbol{\delta}_i \in \mathbf{R}^d$ *of size* $T_V \ge 1$ *where each element of each vector is drawn independently from the standard normal distribution* $\mathcal{N}(0,1)$*. Then, for a fixed* *constant* $c = \frac{1}{(2e)^2}$ *and any* $\epsilon \in (0, G^2\sqrt{d})$ *with probability* $\geq 1 - 2\exp(-\epsilon^2 \frac{c}{G^4})$

$$\left\| \frac{1}{\sqrt{T_V}} \sum_{i \in V} \boldsymbol{\delta}_i \right\| \leq \sqrt{d} + \epsilon \tag{C.2}$$

and with probability $\geq 1 - 4T_V \exp(-\epsilon^2 \frac{c}{G^4})$

$$\forall \boldsymbol{\delta}_j \in V \quad \left\langle \boldsymbol{\delta}_j, \sum_{i \in V} \boldsymbol{\delta}_i \right\rangle \geq d - 3\epsilon\sqrt{T_V d} \tag{C.3}$$

Further, consider any set U *of vectors* $U = \{\boldsymbol{\delta}_i\}$ *of size* T_U*, where each vector also has coordinates drawn i.i.d from the standard normal distribution* $\mathcal{N}(0, 1)$*. Then, with probability* $\geq 1 - 2T_u \exp(-\epsilon^2 \frac{c}{G^4})$

$$\forall \boldsymbol{\delta}_j \in U \quad \left| \left\langle \boldsymbol{\delta}_j, \sum_{i \in V} \boldsymbol{\delta}_i \right\rangle \right| \leq \epsilon\sqrt{T_V d}, \tag{C.4}$$

By union bound, the three events above hold at once with a probability at least $1 - 2(2T_V + T_u + 1)\exp(-\epsilon^2 \frac{c}{G^4})$.

Proof. We split the proof into three parts one each for eqs. (C.2) to (C.4).

Proof of eq. (C.2). As $\boldsymbol{\delta}$ is a vector of random i.i.d standard normal random variables, note that $\frac{1}{\sqrt{T_V}} \sum_i \boldsymbol{\delta}_i$ is also a vector of i.i.d standard normal random variables. This follows from the fact that the sum of T_V standard normal random variables is a mean-zero Gaussian random variable with standard deviation $\sqrt{T_V}$. Thus dividing by the standard deviation makes the variance 1, making it standard normal.

Then, applying lemma 9 with $t = \epsilon$ gives us the following bound:

$$p\left(\left\|\frac{1}{\sqrt{T_V}}\sum_i \boldsymbol{\delta}_i\right\| > \sqrt{d}+\epsilon\right) \leq p\left(\left|\left\|\frac{1}{\sqrt{T_V}}\sum_i \boldsymbol{\delta}_i\right\| - \sqrt{d}\right| > \epsilon\right) \leq 2\exp({}^{-c\epsilon^2}/G^4)$$

Proof of eq. (C.3) We split the inner product into two cases: $T_V = 1$ and $T_V \geq 2$.

Case $T_V = 1$. First note that due to lemma 9,

$$\forall j \in V, \qquad p\left(\|\boldsymbol{\delta}_j\| < \sqrt{d}-\epsilon\right) \leq p\left(\left|\ \|\boldsymbol{\delta}_j\| - \sqrt{d}\right| > \epsilon\right) \leq 2\exp({}^{-c\epsilon^2}/G^4).$$

Then, the following lower bound holds with probability at least $1-2\exp({}^{-c\epsilon^2}/G^4)$

$$\begin{aligned}\forall j \in V, \qquad \left\langle \boldsymbol{\delta}_j, \sum_{i\in V}\boldsymbol{\delta}_i\right\rangle &= \|\boldsymbol{\delta}_j\|^2\\ &\geq (\sqrt{d}-\epsilon)^2\\ &\geq d - 2\epsilon\sqrt{d}\\ &\geq d - 3\epsilon\sqrt{T_V d},\end{aligned}$$

To summarize this case, with the fact that $1-2\exp({}^{-c\epsilon^2}/G^4) \geq 1-4T_V\exp({}^{-c\epsilon^2}/G^4)$, we have that

$$\forall j \in V, \qquad \left\langle \boldsymbol{\delta}_j, \sum_{i\in V}\boldsymbol{\delta}_i\right\rangle \geq d - 3\epsilon\sqrt{T_V d},$$

with probability at least $1-4T_V\exp({}^{-c\epsilon^2}/G^4)$.

Case $T_V \geq 2$. First note that,

$$\forall j \in V \qquad \left\langle \boldsymbol{\delta}_j, \sum_{i\in V}\boldsymbol{\delta}_i\right\rangle = \|\boldsymbol{\delta}_j\|^2 + \left\langle \boldsymbol{\delta}_j, \sum_{i\in V, i\neq j}\boldsymbol{\delta}_i\right\rangle$$

For each of the T_V different $\boldsymbol{\delta}_j$'s, using lemma 9 bounds the probability of the norm $\|\boldsymbol{\delta}_j\|$ being larger than $\sqrt{d} - \epsilon$:

$$p\left(\|\boldsymbol{\delta}_j\| < \sqrt{d} - \epsilon\right) \leq p\left(\left|\ \|\boldsymbol{\delta}_j\| - \sqrt{d}\right| > \epsilon\right) \leq 2\exp({-c\epsilon^2}/{G^4}).$$

In the case where $T_V \geq 2$, we express the inner product of a vector and a sum of vectors as follows

$$\left\langle \boldsymbol{\delta}_j, \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i \right\rangle = \sqrt{T_V - 1}\left\langle \boldsymbol{\delta}_j, \frac{1}{\sqrt{T_V - 1}} \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i \right\rangle,$$

and noting that like above, $\frac{1}{\sqrt{T_V - 1}} \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i$ is a vector of standard normal random variables, we apply lemma 8 to get

$$\forall i \in V \qquad p\left(\left|\left\langle \boldsymbol{\delta}_j, \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i \right\rangle\right| \geq \epsilon\sqrt{(T_V - 1)d}\right) \leq 2\exp\left(-c\frac{\epsilon^2}{G^4}\right).$$

Putting these together, by union bound over V

$$\begin{aligned}
&p\left[\forall j \in V \qquad \left(\|\boldsymbol{\delta}_j\| < \sqrt{d} - \epsilon\right) \text{ or } \left(\left|\left\langle \boldsymbol{\delta}_j, \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i \right\rangle\right| \geq \epsilon\sqrt{(T_V - 1)d}\right)\right] \\
&\leq \sum_{j \in V} p\left(\|\boldsymbol{\delta}_j\| < \sqrt{d} - \epsilon\right) + p\left(\left|\left\langle \boldsymbol{\delta}_j, \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i \right\rangle\right| \geq \epsilon\sqrt{(T_V - 1)d}\right) \\
&\leq \sum_{j \in V} 2\exp\left(-c\frac{\epsilon^2}{G^4}\right) + 2\exp\left(-c\frac{\epsilon^2}{G^4}\right) \\
&\leq 4T_V \exp\left(-c\frac{\epsilon^2}{G^4}\right).
\end{aligned}$$

Thus, with probability at least $1 - 4T_V \exp\left(-c\frac{\epsilon^2}{G^4}\right)$, none of the events happen and

$$\begin{aligned}
\forall j \in V \qquad \left\langle \boldsymbol{\delta}_j, \sum_{i \in V} \boldsymbol{\delta}_i \right\rangle &= \|\boldsymbol{\delta}_j\|^2 + \left\langle \boldsymbol{\delta}_j, \sum_{i \in V, i \neq j} \boldsymbol{\delta}_i \right\rangle \\
&\geq (\sqrt{d} - \epsilon)^2 - \epsilon\sqrt{(T_V - 1)d} \\
&= d - 2\epsilon\sqrt{d} + \epsilon^2 - \epsilon\sqrt{(T_V - 1)d} \\
&\geq d - 2\epsilon\sqrt{(T_V - 1)d} - \epsilon\sqrt{(T_V - 1)d} \\
&\geq d - 3\epsilon\sqrt{T_V d}
\end{aligned}$$

Thus, putting the analysis in the two cases together, as long as $T_V \geq 1$

$$\forall j \in V \qquad \left\langle \boldsymbol{\delta}_j, \sum_{i \in V} \boldsymbol{\delta}_i \right\rangle \geq d - 3\epsilon\sqrt{T_V d},$$

with probability at least $1 - 4T_V \exp\left(-c\frac{\epsilon^2}{G^4}\right)$.

Proof of eq. (C.4) Next, we apply lemma 8 again to the inner product of two vectors of i.i.d standard normal random variables:

$$\forall j \in U \qquad p\left(\quad \left|\left\langle \boldsymbol{\delta}_j, \frac{1}{\sqrt{T_V}} \sum_{i \in V} \boldsymbol{\delta}_i \right\rangle\right| \geq \epsilon\sqrt{d} \right) < 2\exp({}^{-c\epsilon^2}/{}_{G^4}).$$

By union bound over U

$$p\left[\forall j \in U \qquad \left(\quad \left|\left\langle \boldsymbol{\delta}_j, \frac{1}{\sqrt{T_V}} \sum_{i \in V} \boldsymbol{\delta}_i \right\rangle\right| \geq \epsilon\sqrt{d} \right) \right] < 2T_u \exp({}^{-c\epsilon^2}/{}_{G^4}).$$

Thus, with probability at least $1 - 2T_u \exp\left(-c\frac{\epsilon^2}{G^4}\right)$, the following holds, concluding the proof

$$\forall j \in U \qquad \left|\left\langle \boldsymbol{\delta}_j, \frac{1}{\sqrt{T_V}} \sum_{i\in V} \boldsymbol{\delta}_i \right\rangle\right| \leq \epsilon\sqrt{d}.$$

□

Lemma 11. *Let $\{\mathbf{x}_i, \mathbf{y}_i\}_{i\leq n}$ be a collection of d dimensional covariates $\mathbf{x}_i$ and label $\mathbf{y}_i$ sampled according to p_ρ in eq. (4.1). The covariates $\mathbf{x}_i = [\pm B\mathbf{y}_i, \mathbf{y}_i\boldsymbol{\delta}_i]$, where $+B$ in the middle coordinate for $i \in S_{shortcut}$ and $-B$ for $i \in S_{leftover}$. The dual formulation of the following norm-minimization problem*

$$\begin{aligned} \mathbf{w}_{stable} = \arg\min_{\mathbf{w}} \quad & \mathbf{w}_y^2 + \mathbf{w}_z^2 + \|\mathbf{w}_e\|^2 \\ s.t.\ i \in S_{shortcut} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i\boldsymbol{\delta}_i > 1 \\ s.t.\ i \in S_{leftover} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i\boldsymbol{\delta}_i > 1 \\ \mathbf{w}_y \geq B\mathbf{w}_z & \end{aligned}$$

is the following with $\zeta^\top = [-B, 1, \mathbf{0}^{d-2}]$,

$$\max_{\lambda\geq 0, \nu\geq 0} -\frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 + \mathbf{1}^\top\lambda, \tag{C.5}$$

where X is a matrix with $\mathbf{y}_i\mathbf{x}_i$ as its rows.

Proof. We use Lagrange multipliers $\lambda \in \mathbb{R}^n$, $\nu \in \mathbb{R}$ to absorb the constraints and then use strong duality. Letting $\zeta^\top = [-B, 1, \mathbf{0}^{d-2}]$, X be a matrix where the ith row is $\mathbf{x}_i\mathbf{y}_i$,

$$\min_{\mathbf{w}} \quad \|\mathbf{w}\|^2 \qquad \text{s.t.} \qquad X\mathbf{w} - \mathbf{1} \geq 0 \qquad \zeta^\top\mathbf{w} \geq 0$$

has the same solution as

$$\max_{\lambda\geq 0, \nu\geq 0} \min_{\mathbf{w}} \quad \|\mathbf{w}\|^2 - (X\mathbf{w} - \mathbf{1})^\top\lambda - \nu\zeta^\top\mathbf{w} \tag{C.6}$$

Now, we solve the inner minimization to write the dual problem only in terms of λ, ν. Solving the inner minimization involves solving a quadratic program, which is done by setting its gradient to zero,

$$\begin{aligned} &\nabla_{\mathbf{w}} \left(\|\mathbf{w}\|^2 - (X\mathbf{w} - \mathbf{1})^\top \lambda - \nu\zeta^\top \mathbf{w}\right) = 2\mathbf{w} - X^\top \lambda - \nu\zeta = 0 \\ \implies &\mathbf{w} = \frac{1}{2}(\zeta\nu + X^\top \lambda) \end{aligned}$$

Substituting $\mathbf{w} = \frac{1}{2}(\zeta\nu + X^\top\lambda)$ in eq. (C.6)

$$\begin{aligned} \|\mathbf{w}\|^2 - &(X\mathbf{w} - \mathbf{1})^\top \lambda - \nu\zeta^\top \mathbf{w} = \\ &\frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - (\frac{1}{2}(X(\zeta\nu + X^\top\lambda) - \mathbf{1})^\top\lambda - \frac{1}{2}\nu\zeta^\top(\zeta\nu + X^\top\lambda) \\ &= \frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - (\frac{1}{2}(X(\zeta\nu + X^\top\lambda) - \mathbf{1})^\top\lambda - \frac{1}{2}\nu^2\|\zeta\|^2 - \frac{1}{2}\nu\zeta^\top X^\top\lambda \\ &= \frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - \frac{1}{2}(X(X^\top\lambda))^\top\lambda - \frac{1}{2}(X(\zeta\nu))^\top\lambda + \mathbf{1}^\top\lambda - \frac{1}{2}\nu^2\|\zeta\|^2 - \frac{1}{2}\nu\zeta^\top X^\top\lambda \\ &= \frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - \left(\frac{1}{2}(X(X^\top\lambda))^\top\lambda + \frac{1}{2}\nu^2\|\zeta\|^2 + \nu\zeta^\top X^\top\lambda\right) + \mathbf{1}^\top\lambda \\ &= \frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - \left(\frac{1}{2}(X^\top\lambda)^\top X^\top\lambda + \frac{1}{2}\nu^2\|\zeta\|^2 + \nu\zeta^\top X^\top\lambda\right) + \mathbf{1}^\top\lambda \\ &= \frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - \frac{1}{2}\left(\|X^\top\lambda\|^2 + \|\nu\zeta\|^2 + 2\nu\zeta^\top X^\top\lambda\right) + \mathbf{1}^\top\lambda \\ &= \frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 - \frac{1}{2}\|\zeta\nu + X^\top\lambda\|^2 + \mathbf{1}^\top\lambda \\ &= -\frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 + \mathbf{1}^\top\lambda \end{aligned}$$

□

C.1.2 Shortcut learning in max-margin classification

We repeat the DGP from the linear perception task in eq. (4.1) here.

$$\mathbf{y} \sim \text{Rad}, \quad \mathbf{z} \sim \begin{cases} p_\rho(\mathbf{z} = y \mid \mathbf{y} = y) = \rho \\ \\ p_\rho(\mathbf{z} = -y \mid \mathbf{y} = y) = (1-\rho) \end{cases}, \quad \boldsymbol{\delta} \sim \mathcal{N}(0, \mathbf{I}^{d-2}), \quad \mathbf{x} = [B * \mathbf{z}, \mathbf{y}, \boldsymbol{\delta}]\,. \tag{C.7}$$

Theorem 1. *Let* $\mathbf{w}^*$ *be the max-margin predictor on n training samples from* eq. (C.7) *with a leftover group of size* k. *There exist constants* $C_1, C_2, N_0 > 0$ *such that*

$$\forall \ \textit{integers} \ \ k \in \left(0, \frac{n}{10}\right) \tag{C.8}$$

$$\forall \ d \geq C_1 k \log(3n), \tag{C.9}$$

$$\forall \ B > C_2 \sqrt{d/k}, \tag{C.10}$$

with probability at least $1 - \frac{1}{3n}$ *over draws of the training data, it holds that* $B\mathbf{w}_z^* > \mathbf{w}_y^*$.

Before giving the proof of theorem 3, we first give the corollary showing overparameterization is not necessary for theorem 3 to hold.

Corollary 1. *For all* $n > N_0$ *— where the constant* N_0 *is from theorem 3 — with scalar* $\tau \in (0,1)$ *such that the dimension* $d = \tau n < n$, *theorem 3 holds.*

$$\forall k \leq n \times \min\left\{\frac{1}{10}, \frac{\tau}{C_1 \log 3n}\right\},$$

a linear model trained via default-ERM yields a predictor $\mathbf{w}^*$ *such that* $B\mathbf{w}_z^* > \mathbf{w}_y^*$.

Proof. We show that for a range of k, for all $n \geq N_0$ theorem 3 holds for some $d < n$. Note that theorem 3 holds for $n \geq N_0, d = C_1 k \log(3n)$ and

$$\forall k < \frac{n}{10}.$$

Setting $d \leq \tau n$ for some $\tau \in (0,1)$ such that $d < n$ means that theorem 3 holds if

$$C_1 k \log(3n) = d \leq \tau n \implies k \leq \frac{\tau n}{C_1 \log(3n)}.$$

Absorbing this new upper bound into the requirements on k for theorem 3 to hold, we get that for any scalar $n > N_0, \tau \in (0,1), d = \tau n$, theorem 3 holds for

$$\forall k < n \times \min\left\{\frac{1}{10}, \frac{\tau}{C_1 \log(3n)}\right\}.$$

In turn, even though $d < n$, a linear model trained via default-ERM converges in direction to a max-margin classifier such that $\mathbf{w}^*$ with $B\mathbf{w}_z^* > \mathbf{w}_y^*$. □

Proof. (of theorem 3) We consider two norm-minimization problems over $\mathbf{w}$, one under constraint $\mathbf{w}_y \geq B\mathbf{w}_z$ and another under $\mathbf{w}_y < B\mathbf{w}_z$. We show that the latter achieves lower norm and therefore, max-margin will achieve solutions $\mathbf{w}_y < B\mathbf{w}_z$. The two minimization problems are as follows:

$$\begin{aligned} \mathbf{w}_{\text{stable}} = \arg\min_{\mathbf{w}} \quad & w_y^2 + w_z^2 + \|\mathbf{w}_e\|^2 \\ \text{s.t. } i \in S_{\text{shortcut}} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ \text{s.t. } i \in S_{\text{leftover}} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ \mathbf{w}_y \geq B\mathbf{w}_z & \end{aligned} \tag{C.11}$$

$$\begin{aligned} \mathbf{w}_{\text{shortcut}} = \arg\min_{\mathbf{w}} \quad & w_y^2 + w_z^2 + \|\mathbf{w}_e\|^2 \\ \text{s.t. } i \in S_{\text{shortcut}} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ i \in S_{\text{leftover}} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ \mathbf{w}_y < B\mathbf{w}_z & \end{aligned} \tag{C.12}$$

From eq. (C.11), any $\mathbf{w}$ that satisfy the constraints of the dual maximization problem will lower bound the value of the optimum of the primal, $\|\mathbf{w}_{\text{stable}}\|^2 \geq W_{\text{stable}}$. From the eq. (C.12), substituting a guess in $\mathbf{w}_{\text{shortcut}}$ that satisfies the constraints yields an upper bound, $\|\mathbf{w}_{\text{shortcut}}\|^2 \leq W_{\text{shortcut}}$. The actual computation of the bounds $W_{\text{shortcut}}, W_{\text{stable}}$ is in lemmas 12 and 13 which are proved in appendix C.1.3 and appendix C.1.4 respectively. We reproduce the lemmas here for convenience.

Lemma. (7) *Consider the following optimization problem from eq. (C.11) where n samples of $\mathbf{x}_i, \mathbf{y}_i$ come from eq. (4.1) where $\mathbf{x}_i \in \mathbf{R}^d$:*

$$\begin{aligned} \mathbf{w}_{stable} = \arg\min_{\mathbf{w}} \quad & w_y^2 + w_z^2 + \|\mathbf{w}_e\|^2 \\ s.t.\ i \in S_{shortcut} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ s.t.\ i \in S_{leftover} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ \mathbf{w}_y \geq B\mathbf{w}_z & \end{aligned} \tag{C.13}$$

Let $k = |S_{leftover}| > 1$. Then, for a fixed constant $c = \frac{1}{(2e)^2}$, with any scalar $\epsilon < \sqrt{d}$, with probability at least $1 - 2\exp(-c\epsilon^2/G^4)$ and $\forall$ integers $M \in \left[1, \lfloor \frac{n}{2k} \rfloor\right]$,

$$\|\mathbf{w}_{stable}\|^2 \geq W_{stable} = \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}}.$$

Lemma. **(8)** *Consider the following optimization problem from* eq. (C.11) *where n samples of* $\mathbf{x}_i, \mathbf{y}_i$ *come from* eq. (4.1) *where* $\mathbf{x}_i \in \mathbf{R}^d$*:*

$$\begin{aligned} \mathbf{w}_{shortcut} = \arg\min_{\mathbf{w}} \quad & w_y^2 + w_z^2 + \|\mathbf{w}_e\|^2 \\ s.t.\ i \in S_{shortcut} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ i \in S_{leftover} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\ \mathbf{w}_y < B\mathbf{w}_z & \end{aligned} \tag{C.14}$$

Let $k = |S_{leftover}| \geq 1$. *Then, for a fixed constant* $c = \frac{1}{(2e)^2}$*, with any scalar* $\epsilon < \frac{1}{3}\sqrt{\frac{d}{k}} < \sqrt{d}$*, with probability at least* $1 - 2(2k + (n-k) + 1)\exp(-c\frac{\epsilon^2}{G^4})$*, for* $\gamma = \frac{2}{d - 4\epsilon\sqrt{kd}}$*,*

$$\|\mathbf{w}_{shortcut}\|^2 \leq W_{shortcut} = \gamma^2 k(\sqrt{d} + \epsilon)^2 + \frac{\left(1 + \gamma\epsilon\sqrt{dk}\right)^2}{B^2}$$

Together, the lemmas say that for any ∀ integers $M \in \left[1, \lfloor \frac{n}{2k} \rfloor\right]$ and $\epsilon < \frac{1}{3}\sqrt{\frac{d}{k}}$, with probability $\geq 1 - 2\exp(^{-c\epsilon^2}/_{G^4})$

$$\|\mathbf{w}_{\text{stable}}\|^2 \geq W_{\text{stable}} = \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}}.$$

and with probability at least $1 - 2(2k + (n-k) + 1)\exp(-c\frac{\epsilon^2}{G^4})$, for $\gamma = \frac{2}{d - 4\epsilon\sqrt{kd}} > 0$,

$$\|\mathbf{w}_{\text{shortcut}}\|^2 \leq W_{\text{shortcut}} = \gamma^2 k(\sqrt{d} + \epsilon)^2 + \frac{\left(1 + \gamma\epsilon\sqrt{dk}\right)^2}{B^2}$$

First, we choose $\epsilon^2 = 2\frac{G^4}{c}\log(3n)$. This gives us the probability with which these bounds hold: as $k < 0.1n$ we have $k + 2 < \frac{n}{2}$ and

$$1 - 2(2k + (n-k) + 2)\exp(-c\frac{\epsilon^2}{G^4}) = 1 - 2(n + k + 2)\exp(-2\log(3n))$$

$$\begin{aligned}
&\geq 1 - 2(\frac{3n}{2}) \exp(-2\log(3n)) \\
&= 1 - \exp(-2\log(3n) + \log(3n)) \\
&= 1 - \exp(-\log(3n)) \\
&= 1 - \frac{1}{3n}.
\end{aligned}$$

Next, we will instantiate the parameter M and set the constants C_1, C_2 and the upper bound on k in theorem 3 to guarantee the following eq. (separation inequality):

$$W_{\text{shortcut}} = \gamma^2 k(\sqrt{d} + \epsilon)^2 + \frac{\left(1 + \gamma\epsilon\sqrt{dk}\right)^2}{B^2} \quad < \quad \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}} = W_{\text{stable}}, \quad \text{(separation inequality)}$$

which then implies that $\|\mathbf{w}_{\text{shortcut}}\|^2 < \|\mathbf{w}_{\text{stable}}\|^2$, concluding the proof.

Invoking the conditions in theorem 3 and setting the upper bound on k. We will keep the ϵ as is for simplicity of reading but invoke the inequalities satisfied by $\log(3n)$ from theorem 3:

$$\exists \text{ constant } C_1, \qquad d \geq C_1 k \log(3n).$$

Now we let $C_1 = 2\frac{G^4}{cC^2}$ for a constant $C \in \left(0, \frac{1}{3}\right)^{\dagger}$, such that

$$\epsilon^2 = 2\frac{G^4}{c}\log(3n) < C^2 \frac{d}{k} \implies \epsilon < C\sqrt{\frac{d}{k}} \text{ and } \epsilon\sqrt{kd} < Cd. \tag{C.15}$$

We next find a $C \in \left(0, \frac{1}{3}\right)$ such that eq. (separation inequality) holds with $M = 5$, which upper bounds k:

$$M < \frac{n}{2k} \implies \frac{k}{n} < \frac{1}{2M} = \frac{1}{10} \implies k < \frac{n}{10}.$$

[†] The $\frac{1}{3}$ comes from requiring that $\epsilon < \frac{1}{3}\sqrt{\frac{d}{k}}$ from lemma 13.

Simplifying W_{shortcut} and W_{stable}. To actually show $W_{\text{shortcut}} < W_{\text{stable}}$ in eq. (separation inequality), we compare a simplified strict upper bound on the LHS W_{shortcut} and a simplified strict lower bound on the RHS W_{stable}

For the simplification of the RHS W_{stable} of eq. (separation inequality), we will use the fact that $d \geq 2\frac{G^4}{cC^2}\log(3n)k$. Given the assumption $n > N_0$, choosing N_0 to be an integer such that $\log(3N_0) \geq \frac{40cC^2}{G^4}$ means that $\log(3n) > \frac{40cC^2}{G^4}$ and we have

$$\frac{d}{k} > 80 \implies \frac{d}{10k} > 8 \implies \frac{1}{2}\frac{d}{10k} > 4 \tag{C.16}$$

which gives us, for $M = 5$,

$$\begin{aligned}
W_{\text{stable}} &= \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}} && && \text{(C.17)}\\
&= \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{10k}} && && \text{(C.18)}\\
&\geq \frac{1}{\frac{3}{2}\frac{\left(\sqrt{d}+\epsilon\right)^2}{10k}} && \{4 < \frac{1}{2}\frac{d}{10k} < \frac{1}{2}\frac{(\sqrt{d}+\epsilon)^2}{10k} \text{ from eq. (C.16) }\} && \text{(C.19)}\\
&= \frac{20k}{3(\sqrt{d}+\epsilon)^2} && && \text{(C.20)}\\
&\geq \frac{20k}{3(\sqrt{d}+C\frac{\sqrt{d}}{\sqrt{k}})^2} && \{\epsilon < \frac{C\sqrt{d}}{\sqrt{k}} \text{ from eq. (C.15) }\} && \text{(C.21)}\\
&= \frac{20k}{3(1+\frac{C}{\sqrt{k}})^2 d} && && \text{(C.22)}\\
&> \frac{20k}{3(1+C)^2 d} && \{k \geq 1\} && \text{(C.23)}
\end{aligned}$$

Now, we produce a simpler upper bound on the first part of the LHS of eq. (separation inequality):

recalling that $\gamma = \frac{2}{d-4\epsilon\sqrt{kd}}$, and substituting in the upper bounds on ϵ,

$$\begin{aligned}
\gamma^2 k(\sqrt{d}+\epsilon)^2 &= \left(\frac{2(\sqrt{d}+\epsilon)}{d-4\epsilon\sqrt{kd}}\right)^2 k \\
&< 4\left(\frac{(\sqrt{d}+C\sqrt{\frac{d}{k}})}{d-4Cd}\right)^2 k && \{\epsilon < \frac{C\sqrt{d}}{\sqrt{k}} \text{ from eq. (C.15) }\} \\
&= \frac{4k}{d}\left(\frac{(1+\frac{C}{\sqrt{k}})}{1-4C}\right)^2 \\
&\leq \frac{4k}{d}\left(\frac{1+C}{1-4C}\right)^2, && \{k \geq 1\} \qquad \text{(C.24)}
\end{aligned}$$

Next is a simpler upper bound on the second part of the LHS of eq. (separation inequality). Again with $\gamma = \frac{2}{d-4\epsilon\sqrt{kd}}$,

$$\begin{aligned}
\frac{\left(1+\gamma\epsilon\sqrt{dk}\right)^2}{B^2} &= \frac{\left(1+\frac{2\epsilon\sqrt{dk}}{d-4\epsilon\sqrt{kd}}\right)^2}{B^2} \\
&\leq \frac{\left(1+\frac{2Cd}{d-4Cd}\right)^2}{B^2} \\
&= \frac{\left(1+\frac{2C}{1-4C}\right)^2}{B^2}
\end{aligned}$$

Now setting

$$B > \sqrt{2}\frac{\left(1+\frac{2C}{1-4C}\right)}{\sqrt{\frac{4k}{d}}\left(\frac{1+C}{1-4C}\right)}$$

gives the lower bound on B from theorem 3:

$$B > C_2\sqrt{\frac{d}{k}}, \qquad \text{where} \qquad C_2 = \frac{\left(1+\frac{2C}{1-4C}\right)}{\sqrt{2}\left(\frac{1+C}{1-4C}\right)} = \frac{(1-2C)}{\sqrt{2}(1+C)}.$$

Formally,

$$B > C_2\sqrt{\frac{d}{k}} \implies \frac{\left(1+\frac{2C}{1-4C}\right)^2}{B^2} < \frac{1}{2}\left(\sqrt{\frac{4k}{d}}\left(\frac{1+C}{1-4C}\right)\right)^2 = \frac{1}{2}\frac{4k}{d}\left(\frac{1+C}{1-4C}\right)^2. \tag{C.25}$$

By combining the upper bound from eq. (C.25) and the upper bound from eq. (C.24), we get an upper bound on the whole of the LHS of eq. (separation inequality), which in turn provides an upper bound on W_{shortcut}:

$$W_{\text{shortcut}} = \gamma^2 k(\sqrt{d}+\epsilon)^2 + \frac{\left(1+\gamma\epsilon\sqrt{dk}\right)^2}{B^2} \quad < \frac{3}{2}\frac{4k}{d}\left(\frac{(1+C)}{1-4C}\right)^2 \leq \frac{3}{2}\frac{4k}{d}\left(\frac{(1+C)}{1-4C}\right)^2,$$

because $k \geq 1$. Note the upper bound is strict.

Concluding the proof. Now, we show that a C exists such that the following holds, which implies $W_{\text{shortcut}} < W_{\text{stable}}$, which in turn implies eq. (separation inequality) and the proof concludes:

$$W_{\text{shortcut}} < \frac{3}{2}\frac{4k}{d}\left(\frac{(1+C)}{1-4C}\right)^2 \leq \frac{20k}{3(1+C)^2 d} < W_{\text{stable}}.$$

The above inequality holds when

$$6\left(\frac{(1+C)}{1-4C}\right)^2 \leq \frac{20}{3(1+C)^2} \quad \iff \quad (1+C)^2 - \sqrt{\frac{10}{9}}(1-4C) \leq 0.$$

The right hand side holds when the quadratic equation $(1+C)^2 - \sqrt{\frac{10}{9}}(1-4C)$ is non-positive, which holds between the roots of the equation. The equation's positive solution is

$$C = \frac{-3+\sqrt{10}}{3+2\sqrt{10}+\sqrt{5(8+3\sqrt{10})}} \approx 0.008.$$

Setting C to this quantity satisfies the requirement that $C \in (0, \frac{1}{3})$.

Thus, a C exists such that eq. (separation inequality) holds which concludes the proof of theorem 3 for the following constants and constraints implied by C and $M = 5$:

$$C_2 = \frac{(1-2C)}{\sqrt{2}(1+C)} \qquad C_1 = 2\frac{G^4}{cC^2} \qquad k < \frac{n}{10},$$

where G is the ψ_2-norm of a standard normal r.v. and c is the absolute constant from the Bernstein inequality in lemma 7. □

C.1.3 Lower bounding the norm of solutions that rely more on the stable feature

Lemma 12. *Consider the following optimization problem from eq. (C.11) where n samples of $\mathbf{x}_i, \mathbf{y}_i$ come from eq. (4.1) where $\mathbf{x}_i \in \mathbf{R}^d$:*

$$\begin{aligned}
\mathbf{w}_{stable} = \underset{\mathbf{w}}{\arg\min} \quad & w_y^2 + w_z^2 + \|\mathbf{w}_e\|^2 \\
s.t.\ i \in S_{shortcut} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\
s.t.\ i \in S_{leftover} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\
\mathbf{w}_y \geq B\mathbf{w}_z &
\end{aligned} \tag{C.26}$$

Let $k = |S_{leftover}| > 1$. Then, for a fixed constant $c = \frac{1}{(2e)^2}$, with any scalar $\epsilon < \sqrt{d}$, with probability at least $1 - 2\exp(-c\epsilon^2/G^4)$ and $\forall$ integers $M \in \left[1, \lfloor \frac{n}{2k} \rfloor\right]$,

$$\|\mathbf{w}_{stable}\|^2 \geq W_{stable} = \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}}.$$

Proof. By lemma 11, the dual of eq. (C.11) is the following for $\zeta = [-B, 1, \mathbf{0}^{d-2}]$ and X is an $n \times d$

matrix with rows $\mathbf{y}_i\mathbf{x}_i$:

$$\max_{\lambda\geq 0, \nu \geq 0} -\frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 + \mathbf{1}^\top\lambda \tag{C.27}$$

Now by duality

$$\|\mathbf{w}_{\text{stable}}\|^2 \geq \max_{\lambda\geq 0, \nu \geq 0} -\frac{1}{4}\|\zeta\nu + X^\top\lambda\|^2 + \mathbf{1}^\top\lambda,$$

which means any feasible candidate to eq. (C.27) gives a lower bound on $\|\mathbf{w}_{\text{stable}}\|^2$.

Feasible Candidates for λ, ν. We now define a set $U \subset [n]$, and let $\lambda_i = \frac{\alpha}{|U|} > 0$ for $i \in U$ and 0 otherwise. For $M \in (1, \lfloor \frac{n}{2k} \rfloor]$, we take $2Mk$ samples from the training data to be included in U. Formally,

$$U = S_{\text{leftover}} \cup (2M-1)k \text{ a random samples from } S_{\text{shortcut}},$$

which gives the size $|U| = 2Mk$. Then, we let $\nu = \alpha\frac{2(M-1)}{2M} > 0$.

Note that for the above choice of λ, $X^\top\lambda$ is a sum of the rows from U scaled by $\frac{\alpha}{|U|}$. Adding up k rows from S_{leftover} and k rows from S_{shortcut} cancels out the Bs and, so in the B is accumulated $|U| - 2k = 2(M-1)k$ times, and so

$$X^\top\lambda = \left[\alpha * \frac{|U|-2k}{|U|}B, \alpha, \frac{\alpha}{|U|}\sum_{i\in U}\boldsymbol{\delta}_i\right] = \left[\alpha B\frac{2(M-1)}{2M}, \alpha, \frac{\alpha}{|U|}\sum_{i\in U}\boldsymbol{\delta}_i\right].$$

As λ has $\frac{\alpha}{|U|}$ on $|U|$ elements and 0 otherwise, $\lambda^\top\mathbf{1} = \alpha$

As we set $\nu = \alpha\frac{2(M-1)}{2M}$,

$$\nu\zeta + X^\top\lambda = \left[-\alpha B\frac{2(M-1)}{2M} + \alpha\frac{2(M-1)}{2M}B, \alpha\frac{2(M-1)}{2M} + \alpha, 0 + \frac{\alpha}{|U|}\sum_i \boldsymbol{\delta}_i\right] \tag{C.28}$$

$$= \left[0 \quad , \alpha \left(1 + \frac{2(M-1)}{2M} \right), \quad \frac{\alpha}{|U|} \sum_{i} \boldsymbol{\delta}_i \right] \tag{C.29}$$

$$\implies \|\zeta \nu + X^\top \lambda\|^2 = \left\| \left[0, \alpha \left(1 + \frac{2(M-1)}{2M} \right), \frac{\alpha}{|U|} \sum_{i \in U} \boldsymbol{\delta}_i \right] \right\|^2 \tag{C.30}$$

$$= \alpha^2 \left\| \left[0, \left(1 + \frac{2(M-1)}{2M} \right), \frac{1}{|U|} \sum_{i \in U} \boldsymbol{\delta}_i \right] \right\| \tag{C.31}$$

For the chosen values of ν, λ the value of the objective in eq. (C.27) is

$$\frac{-\alpha^2}{4} \left\| \left[0, \left(1 + \frac{2(M-1)}{2M} \right), \frac{1}{|U|} \sum_{i \in U} \boldsymbol{\delta}_i \right] \right\|^2 + \alpha \tag{C.32}$$

Letting

$$\Gamma = \left\| \left[0, \left(1 + \frac{2(M-1)}{2M} \right), \frac{1}{|U|} \sum_{i \in U} \boldsymbol{\delta}_i \right] \right\|^2,$$

the objective is of the form $\alpha - \frac{\alpha^2 \Gamma}{4}$. To maximize with respect to α, setting the derivative of the objective w.r.t α to 0 gives:

$$1 - \frac{2\alpha\Gamma}{4} = 0 \implies \alpha = \frac{2}{\Gamma} \implies \alpha - \frac{\alpha^2 \Gamma}{4} = \frac{2}{\Gamma} - \frac{4}{\Gamma^2} \frac{\Gamma}{4} = \frac{1}{\Gamma}.$$

This immediately gives us

$$\|\mathbf{w}_{\text{stable}}\|^2 \geq \frac{1}{\Gamma},$$

and we lower bound this quantity by upper bounding Γ.

By concentration of gaussian norm as in lemma 9, with probability at least $1 - 2\exp(-c\frac{\epsilon^2}{G^4})$

$$\left\| \frac{1}{|U|} \sum_{i \in U} \boldsymbol{\delta}_i \right\| = \frac{1}{\sqrt{|U|}} \left\| \frac{1}{\sqrt{|U|}} \sum_{i \in U} \boldsymbol{\delta}_i \right\| \leq \frac{1}{\sqrt{|U|}} (\sqrt{d} + \epsilon).$$

In turn, recalling that $|U| = 2Mk$

$$\Gamma \leq \left(\frac{(2(M-1)+2M)}{2M}\right)^2 + \left(\frac{\sqrt{d}+\epsilon}{\sqrt{|U|}}\right)^2 < 4 + \left(\frac{\sqrt{d}+\epsilon}{\sqrt{|U|}}\right)^2 \leq 4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}$$

The upper bound on Γ gives the following lower bound on $\|\mathbf{w}_{\text{stable}}\|^2$:

$$\|\mathbf{w}_{\text{stable}}\|^2 \geq \frac{1}{\Gamma} \geq \frac{1}{4 + \frac{\left(\sqrt{d}+\epsilon\right)^2}{2Mk}}$$

□

C.1.4 Upper bounding the norm of solutions that rely more on the shortcut.

Lemma 13. *Consider the following optimization problem from eq. (C.11) where n samples of $\mathbf{x}_i, \mathbf{y}_i$ come from eq. (4.1) where $\mathbf{x}_i \in \mathbf{R}^d$:*

$$\begin{aligned}
\mathbf{w}_{shortcut} = \arg\min_{\mathbf{w}} \quad & w_y^2 + w_z^2 + \|\mathbf{w}_e\|^2 \\
s.t.\ i \in S_{shortcut} \quad & w_y + Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\
i \in S_{leftover} \quad & w_y - Bw_z + \mathbf{w}_e^\top \mathbf{y}_i \boldsymbol{\delta}_i > 1 \\
\mathbf{w}_y < B\mathbf{w}_z &
\end{aligned} \tag{C.33}$$

Let $k = |S_{leftover}| \geq 1$. Then, for a fixed constant $c = \frac{1}{(2e)^2}$, with any scalar $\epsilon < \frac{1}{3}\sqrt{\frac{d}{k}} < \sqrt{d}$, with probability at least $1 - 2(2k + (n-k) + 1)\exp(-c\frac{\epsilon^2}{G^4})$, for $\gamma = \frac{2}{d - 4\epsilon\sqrt{kd}}$,

$$\|\mathbf{w}_{shortcut}\|^2 \leq W_{shortcut} = \gamma^2 k(\sqrt{d}+\epsilon)^2 + \frac{\left(1+\gamma\epsilon\sqrt{dk}\right)^2}{B^2}$$

Proof. Let $k = |S_{\text{leftover}}|$. The candidate we will evaluate the objective for is

$$\mathbf{w} = \left[\frac{\beta}{B}, 0, \gamma \sum_{j \in S_{\text{leftover}}} \mathbf{y}_j \boldsymbol{\delta}_j\right]. \tag{C.34}$$

High-probability bounds on the margin achieved by the candidate and norm of $\mathbf{w}$ The margins on the shortcut group and the leftover group along with the constraints are as follows:

$$\begin{aligned}
\forall j \in S_{\text{shortcut}} \quad & m_j = 0 + B * \frac{\beta}{B} + \left\langle \mathbf{y}_j \boldsymbol{\delta}_j, \gamma \sum_{i \in S_{\text{leftover}}} \mathbf{y}_i \boldsymbol{\delta}_i \right\rangle \geq 1 \\
\forall j \in S_{\text{leftover}} \quad & m_j = 0 - B * \frac{\beta}{B} + \left\langle \mathbf{y}_j \boldsymbol{\delta}_j, \gamma \sum_{i \in S_{\text{leftover}}} \mathbf{y}_i \boldsymbol{\delta}_i \right\rangle \geq 1.
\end{aligned} \tag{C.35}$$

Due to the standard normal distribution being isotropic, and $\mathbf{y}_j \in \{-1, 1\}$, $\mathbf{y}_j \boldsymbol{\delta}_j$ has the same distribution as $\boldsymbol{\delta}_j$. Then, we apply lemma 10 with $V = S_{\text{leftover}}, U = S_{\text{shortcut}}$ — which means $T_v = k$ and $T_u = (n - k)$ — to bound the margin terms in eq. (C.35) and $\|\mathbf{w}\|^2$ with probability at least

$$1 - 2(2k + (n-k) + 2)\exp(-c\frac{\epsilon^2}{G^4}).$$

Applying the bound in eq. (C.4) in lemma 10 between a sum of vectors and a different i.i.d vector,

$$\forall j \in S_{\text{shortcut}} \qquad \left|\left\langle \mathbf{y}_j \boldsymbol{\delta}_j, \gamma \sum_{i \in S_{\text{leftover}}} \mathbf{y}_i \boldsymbol{\delta}_i \right\rangle\right| \leq \gamma\epsilon\sqrt{kd} \tag{C.36}$$

Applying the bound in eq. (C.3) from lemma 10

$$\forall j \in S_{\text{leftover}} \qquad \left\langle \mathbf{y}_j \boldsymbol{\delta}_j, \gamma \sum_{i \in S_{\text{leftover}}} \mathbf{y}_i \boldsymbol{\delta}_i \right\rangle \geq \gamma\left(d - 3\epsilon\sqrt{kd}\right) \tag{C.37}$$

The margin constraints on the shortcut and leftover from eq. (C.35) respectively imply

$$\beta - \gamma\epsilon\sqrt{dk} \geq 1 \qquad\qquad -\beta + \gamma\left(d - 3\epsilon\sqrt{kd}\right) \geq 1$$

We choose $\beta = 1 + \gamma\epsilon\sqrt{dk}$, which implies an inequality that γ has to satisfy the following, which is due to $d - 3\epsilon\sqrt{kd} > 0$,

$$-(1 + \gamma\epsilon\sqrt{dk}) + \gamma\left(d - 3\epsilon\sqrt{kd}\right) \geq 1 \implies \gamma \geq \frac{2}{d - 4\epsilon\sqrt{kd}}$$

Now, we choose

$$\gamma = \frac{2}{d - 4\epsilon\sqrt{kd}}.$$

Computing the upper bound on the value of the objective in the primal problem in eq. (C.12) The feasible candidate's norm $\|\mathbf{w}\|^2$ is an upper bound on the solution's norm $\|\mathbf{w}_{\text{shortcut}}\|^2$ and so

$$\|\mathbf{w}_{\text{shortcut}}\|^2 \leq \|\mathbf{w}\|^2 = \frac{1}{B^2}\beta^2 + \left\|\gamma \sum_{j \in S_{\text{leftover}}} \mathbf{y}_j \boldsymbol{\delta}_j\right\|^2 = \gamma^2 k \left\|\frac{1}{\sqrt{k}} \sum_{j \in S_{\text{leftover}}} \boldsymbol{\delta}_j\right\|^2 + \frac{\beta^2}{B^2}$$

By lemma 10 which we invoked,

$$\left\|\frac{1}{\sqrt{k}} \sum_{j \in S_{\text{leftover}}} \boldsymbol{\delta}_j\right\|^2 \leq (\sqrt{d} + \epsilon)^2.$$

To conclude the proof, substitute $\beta = 1 + \gamma\epsilon\sqrt{dk}$ and get the following upper bound with $\gamma = \frac{2}{d - 3\epsilon\sqrt{kd}}$:

$$\|\mathbf{w}_{\text{shortcut}}\|^2 \leq \gamma^2 k(\sqrt{d} + \epsilon)^2 + \frac{\beta^2}{B^2} = \gamma^2 k(\sqrt{d} + \epsilon)^2 + \frac{\left(1 + \gamma\epsilon\sqrt{dk}\right)^2}{B^2}.$$

□

C.1.5 Concentration of k and intuition behind theorem 3

Concentration of k around $(1-\rho)n$. Denote the event that the ith sample lies in the leftover group as I_i: then $E[I_i] = 1 - \rho$ and the leftover group size is $k = \sum_i I_i$. Hoeffding's inequality (Theorem 2.2.6 in [246]) shows that for any $t > 0$, k is at most $(1-\rho)n + t\sqrt{n}$ with probability at least $1 - \exp(-2t^2)$:

$$p\left(k - (1-\rho)n > t\sqrt{n}\right) = p\left(\sum_i (I_i - (1-\rho)) > t\sqrt{n}\right) = p\left(\sum_i (I_i - E[I_i]) > t\sqrt{n}\right) \leq \exp(-2t^2).$$

Letting $\rho = 0.9 + \sqrt{\frac{\log 3n}{n}}$ and $t = \sqrt{\log 3n}$, gives us

$$\begin{aligned}
p\left(k - (1-\rho)n > t\sqrt{n}\right) &= p\left(k - 0.1n + \sqrt{n\log 3n} > \sqrt{\log 3n}\sqrt{n}\right) \\
&= p\left(k - 0.1n > 0\right) \\
&\leq \exp(-2t^2) \\
&= \exp(-2\log 3n). \\
&= \left(\frac{1}{3n}\right)^2 \\
&< \frac{1}{3n}
\end{aligned}$$

To connect ρ to shortcut learning due to max-margin classification, we take a union bound of the event that $k < 0.1n$, which occurs with probability at least $1 - \frac{1}{3n}$ and theorem 3 which occurs with probability at least $1 - \frac{1}{3n}$. This union bound guarantees that with probability at least $1 - \frac{2}{3n}$ over sampling the training data, max-margin classification on n training samples from eq. (4.1) relies more on the shortcut feature if ρ is above a threshold; and this threshold converges to 0.9

at the rate of $\sqrt{\log 3n/n}$.

C.1.6 Bumpy losses improve ERM in the under-parameterized setting

Theorem 2. *Consider n samples of training data from* DGP *in eq. (4.1) with $d < n$. Consider a linear classifier $f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ such that for all samples in the training data $\mathbf{y}_i \mathbf{w}^\top \mathbf{x}_i = b$ for any $b \in (0, \infty)$. With probability 1 over draws of samples, $\mathbf{w} = [0, b, 0^{d-2}]$.*

Proof. Letting X be the matrix where each row is $\mathbf{y}_i \mathbf{x}_i$, the theorem statement says the solution $\mathbf{w}^*$

$$X\mathbf{w}^* = b\mathbf{1} \tag{C.38}$$

First, split $\mathbf{w}^* = [w_z^*, w_y^*, \mathbf{w}_{-y}^*]$. Equation (C.38) says that the margin of the model on any sample satisfies

$$\mathbf{y}(\mathbf{w}^*)^\top \mathbf{x} = w_y^* \mathbf{y}^2 + w_z^* \mathbf{y}\mathbf{z} + \mathbf{y}(\mathbf{w}_{-y}^*)^\top \boldsymbol{\delta} = b \qquad \Longrightarrow \qquad \mathbf{y}(\mathbf{w}_{-y}^*)^\top \boldsymbol{\delta} = b - w_y^* \mathbf{y}^2 - w_z^* \mathbf{y}\mathbf{z}$$

We collect these equations for the whole training data by splitting X into columns: denoting Y, Z as vectors of $\mathbf{y}_i$ and $\mathbf{z}_i$ and using $\cdot$ to denote element wise operation, split X into columns that correspond to $\mathbf{y}, \mathbf{z}$ and $\boldsymbol{\delta}$ respectively as $X = [Y \cdot Y \mid Y \cdot Z \mid X_\delta]$. Rearranging terms gives us

$$w_z^* Y \cdot Z + w_y^* \mathbf{1} + X_\delta \mathbf{w}_\delta^* = b\mathbf{1} \qquad \Longrightarrow \qquad X_\delta \mathbf{w}_\delta^* = (b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z.$$

The elements of $Y \cdot Z$ lie in $\{-1, 1\}$ and, as the shortcut feature does not always equal the label, the elements of $Y \cdot Z$ are not all the same sign.

Solutions do not exist when one non-zero element exists in $(b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z$ By definition of $\mathbf{w}^*$

$$X_\delta \mathbf{w}_\delta^* = (b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z.$$

Denote $r = (b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z$. and $A = X_\delta$. Now we show that w.p. 1 solutions do not exist for the following system of linear equations:

$$Aw = r.$$

First, note that $A = X_\delta$ has $\mathbf{y}_i \boldsymbol{\delta}_i$ for rows and as $\mathbf{y}_i \perp\!\!\!\perp \boldsymbol{\delta}_i$ and $\mathbf{y}_i \in \{-1, 1\}$, each vector $\mathbf{y}_i \boldsymbol{\delta}_i$ is distributed identically to a vector of independent standard Gaussian random variables. Thus, A is a matrix of IID standard Gaussian random variables.

Let U denote $D - 2$ indices such that the corresponding rows of A form a matrix $D - 2 \times D - 2$ matrix and r_U has at least one non-zero element; let A_U denote the resulting matrix. Now A_U is a $D - 2 \times D - 2$ sized matrix where each element is a standard Gaussian random variable. Such matrices have rank $D - 2$ with probability 1 because square singular matrices form a measure zero set under the Lebesgue measure over $\mathbf{R}^{D-2 \times D-2}$[247].

We use subscript $\cdot_{-U}$ to denote all but the indices in U. The equation $Aw = r$ implies the following two equations:

$$A_U w = r_U \qquad\qquad A_{-U} w = r_{-U}.$$

As A_U is has full rank $(D - 2)$, $A_U w = r_U$ admits a unique solution $\mathbf{w}_U^* \neq 0$ — because r_U has at least one non-zero element by construction. Then, it must hold that

$$A_{-U} \mathbf{w}_U^* = r_{-U}. \tag{C.39}$$

For any row $v^\top \in A_{-U}$, eq. (C.39) implies that $v^\top \mathbf{w}^*$ equals a fixed constant. As v is a vector of i.i.d standard normal random variables, $v^\top \mathbf{w}^*$ is a gaussian random variable with mean $\sum(\mathbf{w}_i^*)$ and variance $\|\mathbf{w}^*\|^2$. Then with probability 1, $v^\top \mathbf{w}^*$ will not equal a constant. Thus, w.p.1 $A_{-U}\mathbf{w}_U^* = r_{-U}$ is not satisfied, which means w.p.1 there are no solutions to $A\mathbf{w} = r$.

Case where $(b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z$ is zero element-wise As X has rank $D-2$, $X_\delta \mathbf{w}_\delta^* = 0$ only when $\mathbf{w}_\delta^* = 0$.

Each element in $(b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z$ is either $b - w_y^* + w_z^*$ or $b - w_y^* - w_z^*$. Thus,

$$(b - w_y^*)\mathbf{1} - w_z^* Y \cdot Z = 0 \quad \implies \begin{cases} b - w_y^* + w_z^* = 0, \\ b - w_y^* - w_z^* = 0 \end{cases} \tag{C.40}$$

Adding and subtracting the two equations on the right gives

$$2(b - w_y^*) = 0 \qquad \text{and} \qquad 2w_z^* = 0.$$

Thus, $\mathbf{w}_\delta^* = 0, w_z^* = 0, b = w_y^*$. □

C.2 Further experimental details and results

C.2.1 Default-erm with ℓ_2-regularization.

In section 4.3, we show default-ERM achieves zero training loss by using the shortcut to classify the shortcut group and noise to classify the leftover group, meaning the leftover group is overfit. The usual way to mitigate overfitting is via ℓ_2-regularization, which, one can posit, may encourage models to rely on the perfect stable feature instead of the imperfect shortcut and noise.

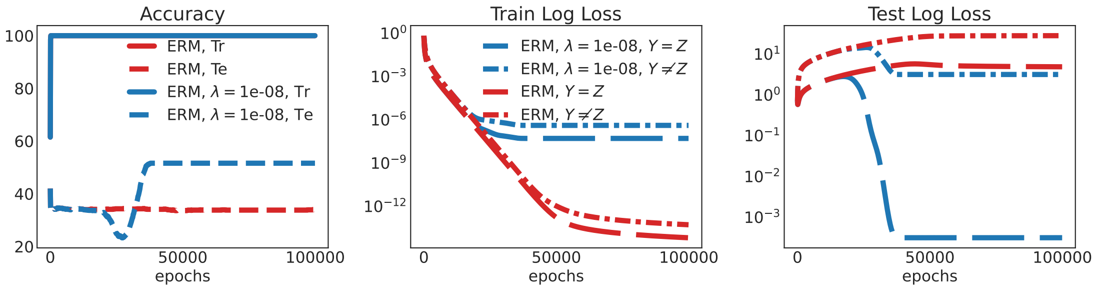


Figure C.1: Default-ERM with ℓ_2-regularization with a penalty coefficient of $\lambda = 10^{-8}$ achieves a test accuracy of $\approx$ 50% , outperforming default-ERM. The right panel shows that ℓ_2-regularization leads to lower test loss on the minority group, meaning that the regularization does mitigate some overfitting. However, the difference between the shortcut and leftover test losses shows that the model still relies on the shortcut.

We train the linear model from section 4.3 with default-ERM and ℓ_2-regularization — implemented as weight decay in the AdamW optimizer [248] — on data from eq. (4.1) with $d = 800, B = 10, n = 1000$. Figure C.1 plots accuracy and losses for the ℓ_2-regularized default-ERM with the penalty coefficient set to 10^{-8}; it shows that ℓ_2-regularization leads default-ERM to build models that only achieve $\approx$ 50% test accuracy.

For smaller penalty coefficients, default-ERM performs similar to how it does without regularization, and for larger ones, the test accuracy gets worse than default-ERM without regularization. We give an intuitive reason for why larger ℓ_2 penalties may lead to larger reliance on the shortcut feature. Due to the scaling factor $B = 10$ in the synthetic experiment, for a fixed norm budget, the model achieves lower loss when using the shortcut and noise compared to using the stable feature. In turn, heavy ℓ_2-regularization forces the model to rely more on the shortcut to avoid the cost of larger weight needed by the model to rely on the stable feature and the noise.

C.2.2 Margin control (marg-ctrl)

In fig. C.2, we plot the different MARG-CTRL losses along with log-loss. Each MARG-CTRL loss has a "bump" which characterizes the loss function's transition from a decreasing function of

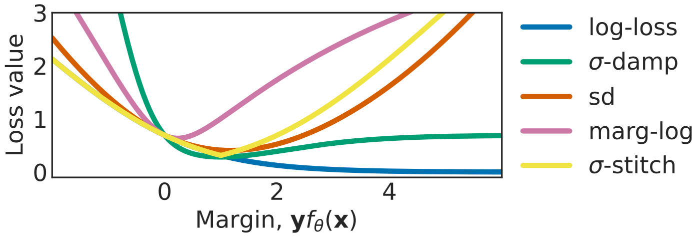


Figure C.2: Comparing log-loss with MARG-CTRL as functions of the margin. Each MARG-CTRL loss has a "bump" which characterizes the loss function's transition from a decreasing function of the margin to an increasing one. These bumps push models to have uniform margins because the loss function's derivative after the bump is negative which discourages large margins. The hyperparameters (temperature in σ-damp or function output target in MARG-LOG.) affect the location of the bump and the slopes of the function on either side of the bump.

the margin to an increasing one. These bumps push models to have uniform margins because the loss function's derivative after the bump is negative which discourages large margins. The hyperparameters — like temperature in σ-damp or function output target in MARG-LOG — affect the location of the bump and the slopes of the function on either side of the bump.

C.2.3 MARG-CTRL ON A LINEAR MODEL

In fig. C.3, we compare default-ERM to σ-stitch. In fig. C.4 and fig. C.5, compare SD and MARG-LOG respectively to default-ERM. The left panel of all figures shows that MARG-CTRL achieves better test accuracy than default-ERM, while the right most panel shows that the test loss is better on the leftover group using MARG-CTRL. Finally, the middle panel shows the effect of controlling margins in training; namely, the margins on the training data do not go to ∞, evidenced by the training loss being bounded away from 0. Depending on the shortcut feature leads to different margins and therefore test losses between the shortcut and leftover groups; the right panel in each plot shows that the the test losses on both groups reach similar values, meaning MARG-CTRL mitigates dependence on the shortcut. While default-ERM fails to perform better than chance (50%) even after $100,000$ epochs (see fig. 4.1), MARG-CTRL mitigates shortcut learning within 5000 epochs and achieves 100% test accuracy.

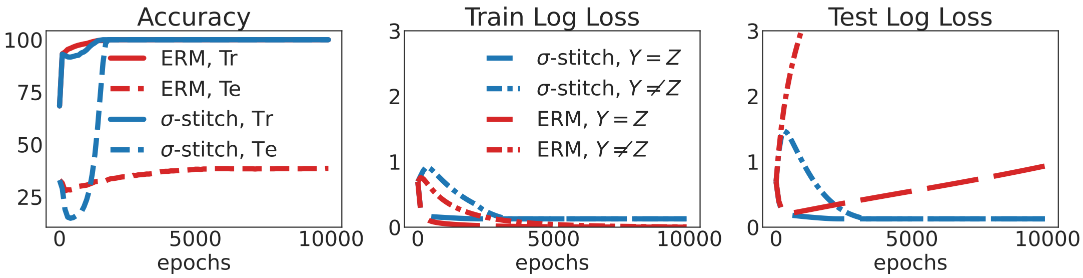


Figure C.3: A linear trained with σ-stitch depend on the perfect stable feature to achieve perfect test accuracy, unlike default-ERM. The middle panel shows that σ-stitch does not let the loss on the training shortcut group to go to zero, unlike default-ERM, and the right panel shows the test leftover group loss is better.

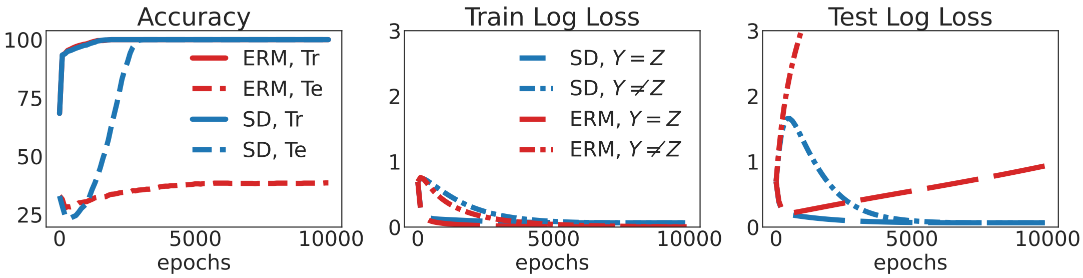


Figure C.4: A linear model trained with SD depend on the perfect stable feature to achieve perfect test accuracy whereas default-ERM performs worse than random chance. The middle panel shows that SD does not let the loss on the training shortcut group to go to zero, unlike vanilla default-ERM, and the right panel shows the test-loss is better for the leftover group.

C.2.4 MARG-CTRL VS. DEFAULT-ERM WITH A NEURAL NETWORK

With $d = 100$ and $B = 10$ in eq. (4.1), we train a two layer neural network on 3000 samples from the training distribution. The two layer neural network has a 200 unit hidden layer that outputs a scalar. Figure C.6 shows that a neural network trained via default-ERM fails to cross 50% test accuracy even after 40, 000 epochs, while achieving less than 10^{-10} in training loss.

In fig. C.7, we compare default-ERM to σ-stitch. In fig. C.9 and fig. C.10, compare SD and MARG-LOG respectively to default-ERM. The left panel of all figures shows that MARG-CTRL achieves better test accuracy than default-ERM, while the right most panel shows that the test loss is better on the leftover group using MARG-CTRL. Finally, the middle panel shows the effect of controlling margins in training; namely, the margins on the training data do not go to ∞, evidenced by the training loss being bounded away from 0.

C.2.5 Spectral decoupling for a linear model on the linear DGP in eq. (4.1).

We first show that a linear classifier trained with SD achieves 100% test accuracy while default-ERM performs worse than chance on the test data; so, SD builds models with more dependence on the stable perfect feature, compared to ERM. Next, we outline the assumptions for the gradient starvation (GS) regime from Pezeshki et al. [88] and then instantiate it for a linear model under the data generating process in eq. (4.1), showing that the assumptions for the GS-regime are violated.

Figure C.4 shows the results of training a linear model with SD on training data of size 1000 sampled as per eq. (4.1) from $p_{\rho=0.9}$ with $d = 300$; the test data also has a 1000 samples but comes from $p_{\rho=0.1}$. Figure C.4 shows that SD builds models with improved dependence on the perfect stable feature, as compared to ERM, to achieve 100% test accuracy.

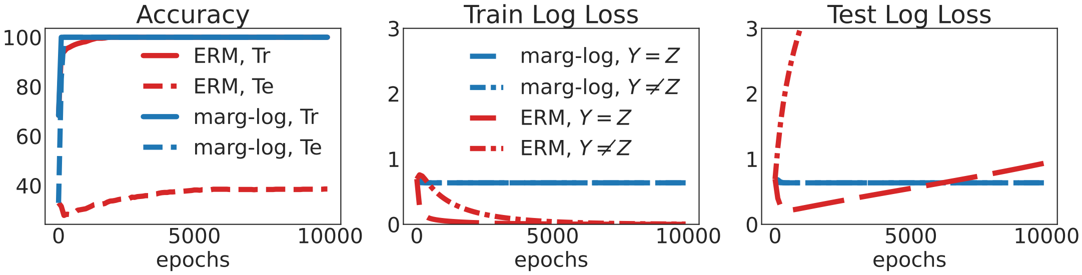
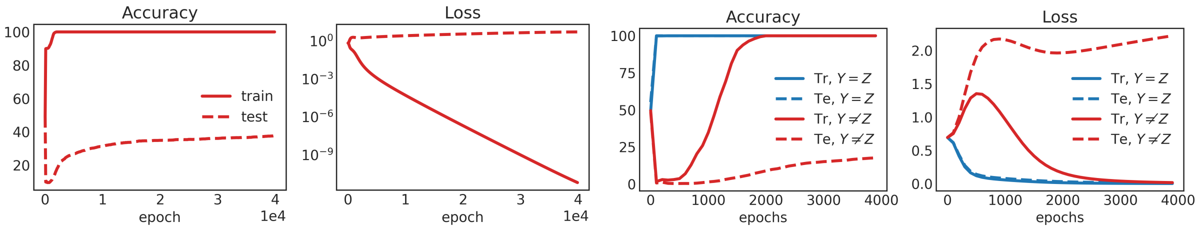


Figure C.5: A linear model trained with MARG-LOG depend on the perfect stable feature to achieve perfect test accuracy whereas default-ERM performs worse than random chance. The middle panel shows that MARG-LOG does not let the loss on the training shortcut group to go to zero, unlike default-ERM, and the right panel shows the test-loss is better for the leftover group.



(a) Average accuracy and loss curves. **(b)** Accuracy/loss on shortcut and leftover groups.

Figure C.6: Training a two-layer neural network with default-ERM on data from eq. (4.1). The model achieves 100% train accuracy but < 40% test accuracy even after 40,000 epochs. The plot below zooms in on the first 4000 epochs and shows that the model drives down loss on the test shortcut groups but not on the test leftover group. This shows that the model uses the shortcut to classify the shortcut group and noise for the leftover.

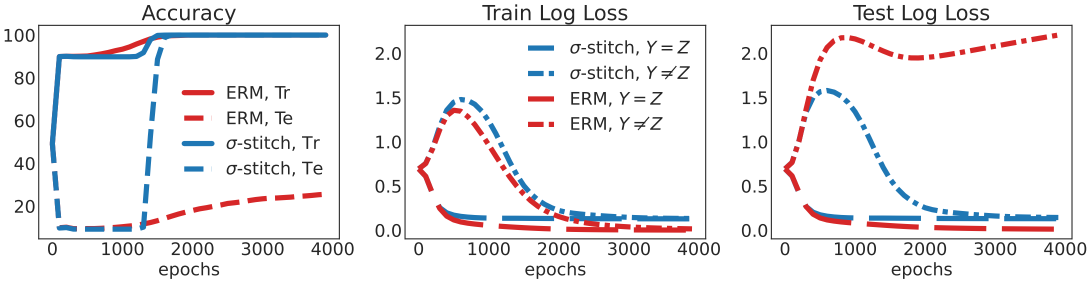


Figure C.7: A neural network trained with σ-stitch depend on the perfect stable feature to achieve perfect test accuracy, unlike default-ERM. The middle panel shows that σ-stitch does not let the loss on the training shortcut group to go to zero, unlike default-ERM, and the right panel shows the test leftover group loss is better.

C.2.5.1 The linear example in Equation (4.1) violates the gradient starvation regime.

Background on [88]. With the aim of explaining why ERM-trained neural networks depend more on one feature over a more informative one, Pezeshki et al. [88] derive solutions to ℓ_2-regularized logistic regression in the NTK; they let the regularization coefficient be small enough for the regularized solution to be similar in direction to the unregularized solution. Given n samples $\mathbf{y}^i, \mathbf{x}^i$, let $\mathbf{Y}$ be a diagonal matrix with the labels on its diagonal, $\mathbf{X}$ be a matrix with $\mathbf{x}^i$ as its rows, and $\hat{\mathbf{y}}(\mathbf{X}, \theta) = f_\theta(\mathbf{X})$ be the n-dimensional vector of function outputs where each element is $\hat{\mathbf{y}}^i = f_\theta(\mathbf{x}^i)$. In gradient-based training in the NTK regime, the vector of function outputs of the

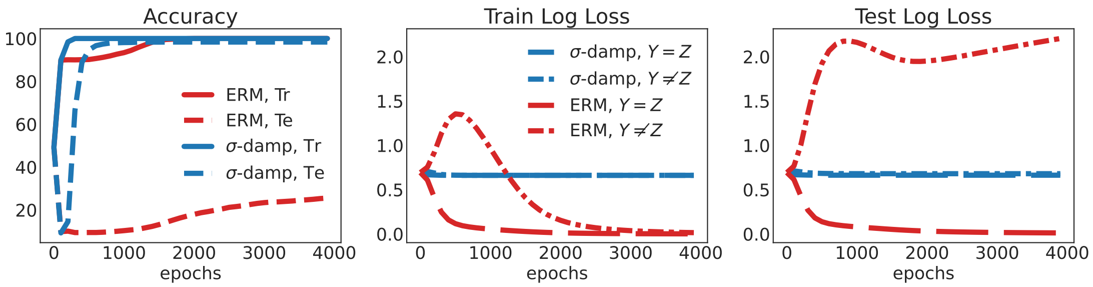


Figure C.8: A neural network trained with σ-damp depend on the perfect stable feature to achieve perfect test accuracy whereas default-ERM performs worse than random chance. The middle panel shows that σ-damp does not let the loss on the training shortcut group to go to zero, unlike vanilla default-ERM, and the right panel shows the test-loss is better for the leftover group.

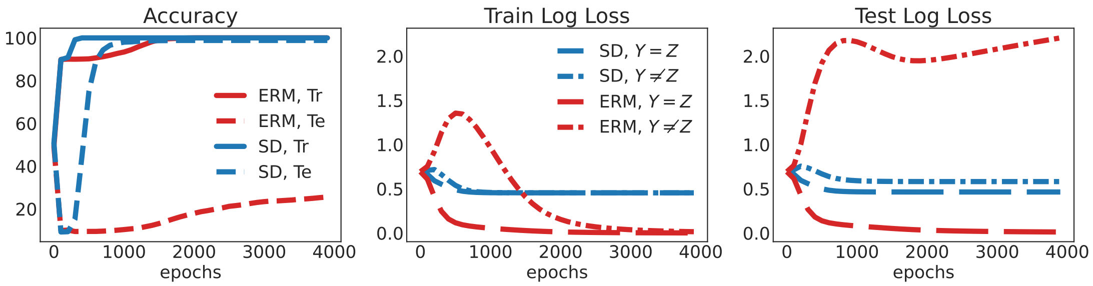


Figure C.9: A neural network trained with SD depend on the perfect stable feature to achieve perfect test accuracy whereas default-ERM performs worse than random chance. The middle panel shows that SD does not let the loss on the training shortcut group to go to zero, unlike vanilla default-ERM, and the right panel shows the test-loss is better for the leftover group.

network with parameters θ can be approximated as $\hat{\mathbf{y}} = \Phi_0\theta$, where Φ_0 is the neural-tangent-random-feature (NTRF) matrix at initialization:

$$\Phi_0 = \frac{\partial \hat{\mathbf{y}}(\mathbf{X}, \theta_0)}{\partial \theta_0}$$

To define the features, the strength (margin) of each feature, and how features appear in each sample, [88] compute the singular value decomposition (SVD) of the NTRF Φ_0 multiplied by the diagonal-label matrix $\mathbf{Y}$:

$$\mathbf{Y}\Phi_0 = \mathbf{U}\mathbf{S}\mathbf{V}^\top. \tag{C.41}$$

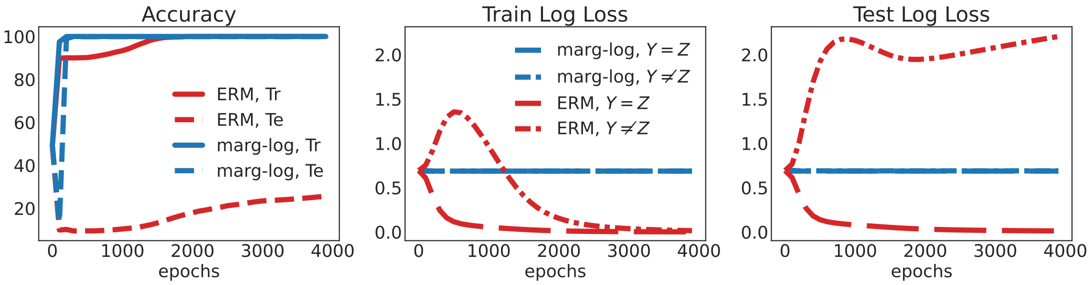


Figure C.10: A neural network trained with MARG-LOG depend on the perfect stable feature to achieve perfect test accuracy whereas default-ERM performs worse than random chance. The middle panel shows that MARG-LOG does not let the loss on the training shortcut group to go to zero, unlike default-ERM, and the right panel shows the test-loss is better for the leftover group.

The rows of $\mathbf{V}$ are features, the diagonal elements of $\mathbf{S}$ are the strengths of each feature and the ith row of $\mathbf{U}$ denotes how each feature appears in the NTRF representation of the ith sample.

To study issues with the solution to ℓ_2-regularized logistic regression, Pezeshki et al. [88] define the gradient starvation (GS) regime. Under the GS regime, they assume $\mathbf{U}$ is a perturbed identity matrix that is also unitary: for a small constant $\delta << 1$, such a matrix has all diagonal elements $\sqrt{1-\delta^2}$ and the rest of the elements are of the order δ such that the rows have unit ℓ_2-norm.

THE GS REGIME IS VIOLATED IN EQ. (4.1). When f_θ is linear, $f_\theta(\mathbf{x}) = \theta^\top \mathbf{x}$, the NTRF matrix is

$$\frac{\partial \hat{\mathbf{y}}(\mathbf{X}, \theta_0)}{\partial \theta_0} = \frac{\partial \mathbf{X}\theta_0}{\partial \theta_0} = \mathbf{X}.$$

In this case, let us look at an implication of $\mathbf{U}$ being a perturbed identity matrix that is also unitary, as Pezeshki et al. [88] assume. With $(\mathbf{u}^i)^\top$ as the ith row of $\mathbf{U}$, the transpose of ith sample can be written as $(\mathbf{x}^i)^\top = (\mathbf{u}^i)^\top \mathbf{S}\mathbf{V}$. [88] assume that $\delta << 1$ in that the off-diagonal terms of $\mathbf{U}$ are small perturbations such that off-diagonal terms of $\mathbf{U}(\mathbf{S}^2 + \lambda \mathbf{I})\mathbf{U}^\top$ have magnitude much smaller than 1, meaning that the terms $|(\mathbf{u}^i)^\top \mathbf{S}^2(\mathbf{u}^j) + \lambda| << 1$ for $i \neq j$ and positive and small $\lambda << 1$.

Then,

$$\begin{aligned}|\mathbf{y}^i\mathbf{y}^j(\mathbf{x}^i)^\top\mathbf{x}^j| &= |(\mathbf{x}^i)^\top\mathbf{x}^j| \\ &= |(\mathbf{u}^i)^\top\mathbf{S}\mathbf{V}^\top\mathbf{V}\mathbf{S}\mathbf{u}^j| \\ &= |(\mathbf{u}^i)^\top\mathbf{S}^2\mathbf{u}^j| \\ &<< 1\end{aligned}$$

In words, this means that any two samples $\mathbf{x}^i, \mathbf{x}^j$ are nearly orthogonal. Now, for samples from eq. (4.1), for any i, j such that $\mathbf{z}^j = \mathbf{z}^i$ and $\mathbf{y}^i = \mathbf{y}^j$,

$$\left|(\mathbf{x}^i)^\top\mathbf{x}^j\right| = \left|B^2\mathbf{z}^i\mathbf{z}^j + \mathbf{y}^i\mathbf{y}^j + (\boldsymbol{\delta}^i)^\top\boldsymbol{\delta}^j\right| \geq |100 + 1 + (\boldsymbol{\delta}^i)^\top\boldsymbol{\delta}^j|$$

As $\boldsymbol{\delta}$ are isotropic Gaussian vectors, around half the pairs i, j will have $(\boldsymbol{\delta}^i)^\top\boldsymbol{\delta}^j > 0$ meaning $\left|(\mathbf{x}^i)^\top\mathbf{x}^j\right| > 101$. This lower bound implies that $\mathbf{U}$ is not a perturbed identity matrix for samples from eq. (4.1). This violates the setup of the gradient starvation regime from [88].

Thus, the linear DGP in eq. (4.1) does not satisfy the conditions for the GS regime that is proposed in [88]. The GS regime blames the coupled learning dynamics for the different features as the cause for default-ERM-trained models depending more on the less informative feature. Pezeshki et al. [88] derive spectral decoupling (SD) to avoid coupling the training dynamics, which in turn can improve a model's dependence on the perfect feature. SD adds a penalty to the function outputs which [88] show decouples training dynamics for the different features as defined by the NTRF matrix:

$$\ell_{\text{SD}}(\mathbf{y}, f_\theta(\mathbf{x})) = \log(1 + \exp(\mathbf{y}f_\theta)) + \lambda|f_\theta(\mathbf{x})|^2$$

As eq. (4.1) lies outside the GS regime, the success of SD on data from eq. (4.1) cannot be explained as a consequence of avoiding the coupled training dynamics in the GS regime Pezeshki et al.

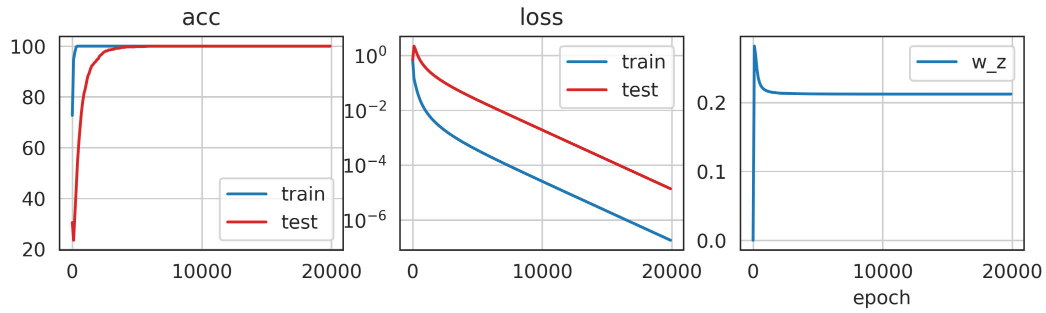


Figure C.11: With $d = 200$ and $n = 1000$, a linear classifier can still depend on the shortcut feature and achieve 100% test accuracy. Nagarajan et al. [86] consider linearly separable data and formalize geometric properties of the data that make max-margin classifiers give non-zero weight to the shortcut feature ($\mathbf{w}_z > 0$). In their example, it is unclear when $\mathbf{w}_z > 0$ leads to poor accuracy in the leftover group because Nagarajan et al. [86] do not separate the model's dependence on the stable feature from the dependence on noise. The example here gives an example where $\mathbf{w}_z > 0$ but test accuracy is 100%. demonstrating that guarantees on test leftover group error require comparing $\mathbf{w}_y$ and $\mathbf{w}_z$; the condition $\mathbf{w}_z > 0$ alone is insufficient.

[88]. However, looking at SD as MARG-CTRL, the success of SD, as in fig. C.4, is explained as a consequence encouraging uniform margins.

An example of perfect test accuracy even with dependence on the shortcut. In fig. C.11, we train a linear model with default-ERM on data from eq. (4.1), showing that even when shortcut dependence is non-zero, test leftover group accuracy can be 100%. Nagarajan et al. [86] consider linearly separable data and formalize geometric properties of the data that make max-margin classifiers give non-zero weight to the shortcut feature ($\mathbf{w}_z > 0$). In their example, it is unclear when $\mathbf{w}_z > 0$ leads to poor accuracy in the leftover group because Nagarajan et al. [86] do not separate the model's dependence on the stable feature from the dependence on noise. The example in fig. C.11 gives an example where $\mathbf{w}_z > 0$ but test accuracy is 100%, demonstrating that guarantees on test leftover group error require comparing $\mathbf{w}_y$ and $\mathbf{w}_z$; the condition $\mathbf{w}_z > 0$ alone is insufficient. In contrast, theorem 1 characterizes cases where leftover group accuracy is worse than random even without overparameterization.

C.2.6 Experimental details

C.2.6.1 Background on Just Train Twice (jtt) and Correct-n-Contrast (cnc)

jtt Liu et al. [53] develop jtt with the aim of building models robust to subgroup shift, where the mass of disjoint subgroups of the data changes between training and test times. To work without training group annotations, jtt assumes erm builds models with high worst-group error. With this assumption, jtt first builds an "identification" model via erm to pick out samples that are misclassified due to model's dependence on the shortcut. Then, jtt trains a second model again via erm on the same training data with the loss for the misclassified samples upweighted (by constant λ). As Liu et al. [53] point out, the number of epochs to train the identification model and the upweighting constant are hyperparameters that require tuning using group annotations. As Liu et al. [53], Zhang et al. [81] show that jtt and cnc outperforms lff and other two-stage shortcut-mitigating methods ([81]), so we do not compare against them.

Correct-n-Contrast (cnc) In a fashion similar to jtt, the first stage of cnc is to train a model with regularized erm to predict based on spurious attributes, i.e. shortcut features. Zhang et al. [81] develop a contrastive loss to force the model to have similar representations across samples that share a label but come from different groups (approximately inferred by the first-stage erm model). Formally, the first-stage model is used to approximate the spurious attributes in one of two ways: 1) predict the label with the model, 2) cluster the representations into as many clusters as there are classes, and then use the cluster identity. The latter technique was first proposed in [249]. For an anchor sample $(\mathbf{y}^i, \mathbf{x}^i)$ of label $\mathbf{y} = y$, positive samples P_i are those than have the same label but have the predicted spurious attribute is a different value: $\hat{z} \neq y$. Negatives N_i are those that have a different label but the spurious attribution is the same: $\hat{z} = y$. For a temperature parameter τ and representation function r_θ, the per-sample contrastive loss for CNC is:

$$\ell_{cont}(r_\theta, i) = \mathbb{E}_{\mathbf{x}^p \sim P_i}\left[-\log \frac{\exp\left(r_\theta(\mathbf{x}^i)^\top r_\theta(\mathbf{x}^p)/\tau\right)}{\sum_{n \in N_i} \exp\left(r_\theta(\mathbf{x}^i)^\top r_\theta(\mathbf{x}^n)/\tau\right) + \sum_{p \in P_i} \exp\left(r_\theta(\mathbf{x}^i)^\top r_\theta(\mathbf{x}^p)/\tau\right)}\right].$$

The samples i are called *anchors*. For a scalar λ to trade off between contrastive and predictive loss, the overall per-sample loss in the second-stage in CNC is

$$\lambda \ell_{cont}(r_\theta, i) + (1 - \lambda)\ell_{log-loss}(\mathbf{y}^i w^\top r_\theta(\mathbf{x}^i)).$$

CNC USES HYPERPARAMETERS INFORMED BY DATASET-SPECIFIC EMPIRICAL RESULTS FROM PRIOR WORK. The original implementation of CNC from Zhang et al. [81] uses specific values of first-stage hyperparameters like weight decay and early stopping epoch for each dataset by using empirical results from prior work [16, 53]. The prior work finds weight-decay and early stopping epoch which lead default-ERM models to achieve low test worst-group accuracy, implying that the model depends on the spurious attribute. This means the first-stage models built in CNC are pre-selected to pay attention to the spurious attributes. For example, [81] point out that the first-stage model they use for Waterbirds predicts the spurious feature with an accuracy of 94.7%.

Without using dataset-specific empirical results from prior work, choosing LR and WD requires validating through the whole CNC procedure. We let CNC use the same LR and WD for both stages and then validate the choice using validation performance of the second-stage model. This choice of hyperparameter validation leads to a similar number of validation queries for all methods that mitigate shortcuts.

C.2.6.2 Training details

Variants of marg-ctrl to handle label imbalance. The three datasets that we use in our experiments — Waterbirds, CelebA, and Civilcomments — all have an imbalanced (non-uniform) marginal distribution over the label; for each dataset,

$$\max_{\text{class}\in\{-1,1\}} p(\mathbf{y} = \text{class}) > 0.75.$$

When there is sufficiently large imbalance, restricting the margins on all samples could bias the training to reduce loss on samples in the most-frequent class first and overfit on the rest of the samples. This could force a model to predict the most frequent class for all samples, resulting in high worst-group error.

To prevent such a failure mode, we follow [88] and define variants of σ-damp, marg-log, and σ-stitch that have either 1) different maximum margins for different classes or 2) different per-class loss values for the same margin value. Mechanically, these variants encourage uniform margins within each class, thus encouraging the model to rely less on the shortcut feature. We give the variants here for labels taking values in $\{-1, 1\}$:

1. With per-class temperatures $T_{-1}, T_1 > 0$ the variant of σ-damp is

$$\begin{aligned} &\text{with } f_\theta = {w_f}^\top r_\theta(\mathbf{x}),\\ &\quad \ell_{\sigma\text{-damp}}(\mathbf{y}, f_\theta) = \ell_{log}\left[T_{\mathbf{y}} * 1.278\mathbf{y}f_\theta\left(1 - \sigma\left(1.278 * \mathbf{y}f_\theta\right)\right)\right]\end{aligned}$$

   The 1.278 comes in to make sure the maximum input to log-loss occurs at $f_\theta = 1$. However, due to the different temperatures $T_1 \neq T_{-1}$, achieving the same margin on all samples produces lower loss on the class with the larger temperature.

2. With per-class temperatures $T_{-1}, T_1 > 0$ the variant of σ-stitch is

$$\text{with } f_\theta = {w_f}^\top r_\theta(\mathbf{x}),$$
$$\ell_{\sigma\text{-stitch}}(\mathbf{y}f_\theta) = \ell_{log}\left(T_\mathbf{y}\left[\quad \mathbf{1}[\mathbf{y}f_\theta(\mathbf{x}) < 1] \times \mathbf{y}f_\theta(\mathbf{x}) + \mathbf{1}[\mathbf{y}f_\theta(\mathbf{x}) > 1] \times (2 - \mathbf{y}f_\theta(\mathbf{x}))\ \right]\right)$$

3. With per-class function output targets $\gamma_{-1}, \gamma_1 > 0$ the variant of MARG-LOG is

$$\text{with } f_\theta = {w_f}^\top r_\theta(\mathbf{x}),$$
$$\ell_{\text{MARG-LOG}}(\mathbf{y}f_\theta) = \ell_{log}(\mathbf{y}f_\theta) + \lambda \log(1 + |f_\theta - \gamma_\mathbf{y}|^2).$$

These per-class variants are only for training; at test time, the predicted label is $\mathtt{sign}(f_\theta)$.

Details of the vision and language experiments. We use the same datasets from [53], downloaded via the scripts in the code from [83]; see [83] for sample sizes and the group proportions. For the vision datasets, we finetune a resnet50 from Imagenet-pretrained weights and for Civilcomments, we finetune a BERT model.

Optimization details. For all methods and datasets, we tune over the following weight decay (WD) parameters: $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ For the vision datasets, we tune learning rate (LR) over $10^{-4}, 10^{-5}$ and for CivilComments, we tune over $10^{-5}, 10^{-6}$. For CivilComments, we use the AdamW optimizer while for the vision datasets, we use the Adam optimizer; these are the standard optimizers for their respective tasks [3, 80]. We use a batch size of 128 for both CelebA and Waterbirds, and train for 20 and 100 epochs respectively. For CivilComments we train for 10 epochs with a batch size of 16.

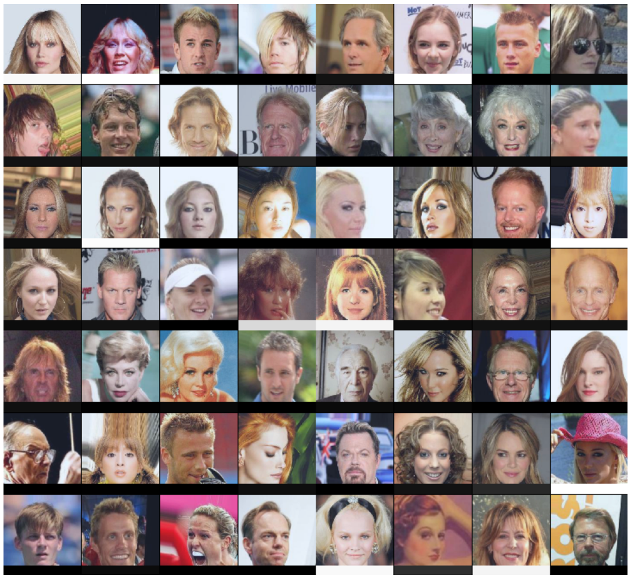


Figure C.12: Images mis-classified by a model trained on CelebA data with equal group sizes, i.e. without a shortcut. Samples with blonde as the true label have a white strip at the bottom while samples with non-blonde as the true label have a black strip at the bottom. The figure demonstrates that many images with blonde people in the image have the non-blonde label, thus demonstrating label noise. For example, see a blonde man in the first row that is labelled non-blonde and a non-blonde lady in the third row that is lablled blonde. Yet, MARG-CTRL improves over ERM for many LR and WD combinations; see fig. C.13.

PER-METHOD HYPERPARAMETERS. Like in [88], the per-class temperatures T_{-1}, T_1 for σ-damp and σ-stitch, and the function output targets γ_{-1}, γ_1 for MARG-LOG are hyperparameters that we tune using the worst-group accuracy or label-balanced average accuracy computed on the validation dataset, averaged over 2 seeds.

1. For σ-stitch, we select from $T_{-1} \in \{1, 2\}$ and $T_1 \in \{2, 4, 8, 12\}$ such that $T_1 > T_{-1}$.
2. For σ-damp, we search over $T_{-1} \in \{1, 2\}$ and $T_1 \in \{2, 4\}$ such that $T_1 > T_{-1}$.
3. For SD and MARG-LOG, we search over $\gamma_{-1} \in \{-1, 0, 1\}$ and $\gamma_1 \in \{1, 2, 2.5, 3\}$ for the image datasets and $\gamma_1 \in \{1, 2\}$ for the text dataset, and the penalty coefficient is set to be $\lambda = 0.1$
4. For JTT, we search over the following parameters: the number of epochs $T \in \{1, 2\}$ for CelebA and Civilcomments and $T \in \{10, 20, 30\}$ for Waterbirds, and the upweighting constant $\lambda \in \{20, 50, 100\}$ for the vision datasets and $\lambda \in \{4, 5, 6\}$ for Civilcomments.

5. For CNC, we search over the same hyperparameter as [81] : the temperature in $\tau \in \{0.05, 0.1\}$, the contrastive weight $\lambda \in \{0.5, 0.75\}$, and the gradient accumulation steps $s \in \{32, 64\}$. For the language task in Civilcomments, we also try one additional $s = 128$.

C.2.7 Marg-ctrl improves over default-erm on CelebA even without the stable feature being perfect.

CelebA is a perception task in that the stable feature is the color of the hair in the image. But unlike the synthetic experiments, MARG-CTRL does not achieve a 100% test accuracy on CelebA. We investigated this and found that CelebA in fact has some label noise.

We trained a model via the MARG-CTRL method σ-damp on CelebA data with no shortcut; this data is constructed by subsampling the groups to all equal size, (5000 samples). This achieves a test worst-group accuracy of 89%. We visualized the images that were misclassified by this model and found that many images with blond-haired people were classified as having non-blonde hair. Figure C.12 shows 56 misclassified images where samples with blonde as the true label have a white strip at the bottom while samples with non-blonde as the true label have a black strip at the bottom. The figure shows that images with blonde people can have the non-blonde label, thus demonstrating label noise. Thus, MARG-CTRL improves over ERM even on datasets like CelebA where the stable features do not determine the label.

C.2.8 Sensitivity of erm and marg-ctrl to varying lr and wd

In fig. C.13, we compare the test worst-group accuracy of default-ERM and MARG-CTRL on CelebA, for different values of LR and WD. There are 8 combinations of LR and WD for which ERM is run. For each combination of LR and WD, the hyperparameters of the MARG-CTRL method (values of λ, T, v) are tuned using validation group annotations, and the test worst-group accuracy corresponds to the best method hyperparameters. Default-ERM's performance changes more with LR

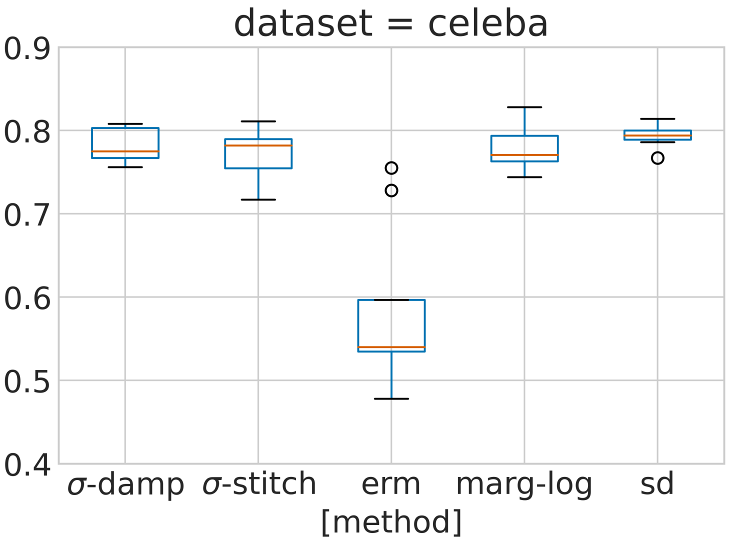


Figure C.13: Test worst-group accuracy on CelebA of default-ERM and MARG-CTRL for different values of LR and WD. Default-ERM's performance changes more with LR and WD than MARG-CTRL, which shows that default-ERM is more sensitive than MARG-CTRL. Only 2 combinations of LR and WD improve ERM beyond a test worst-group accuracy of 60%, while every MARG-CTRL method achieves more than 70% test worst-group accuracy for every combination of LR and WD.

and WD than MARG-CTRL, which shows that default-ERM is more sensitive than MARG-CTRL. Only 2 combinations of LR and WD improve ERM beyond a test worst-group accuracy of 60%, while every MARG-CTRL method achieves more than 70% test worst-group accuracy for every combination of LR and WD.

D | Appendices for chapter 5

D.1 Theoretical Details and Proofs

Notation We use the expectation operator in different contexts in the proof. $\mathbb{E}_q$ denotes expectation with respect to the density q and $\mathbb{E}_{\mathbf{z}}$ denotes expectation with respect to the density of the random variable $\mathbf{z}$. When the density function or the random variable are clear from the context, we drop the subscript and use $\mathbb{E}$.

D.1.1 The general IV causal graph with covariates/observed confounders

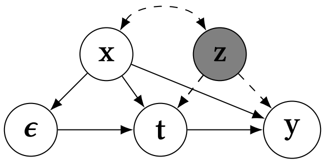


Figure D.1: Causal graph with hidden confounder $\mathbf{z}$, outcome $\mathbf{y}$, IV $\boldsymbol{\epsilon}$, treatment $\mathbf{t}$ and covariates $\mathbf{x}$.

Figure D.1 is the general version of the IV problem where the instrumental variable property holds true after conditioning on $\mathbf{x}$. This is sometimes called a conditional instrument. All our proofs and results carry over to the situation with covariates after conditioning all estimables and distributions on $\mathbf{x}$. VDE in this setting with covariates is re-written as:

$$\max_{\theta,\phi} \mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon},\mathbf{x})} \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon},\mathbf{x})} \log p_\phi(\mathbf{t} \mid \hat{\mathbf{z}}, \boldsymbol{\epsilon}, \mathbf{x}) - \lambda \mathbf{I}_\theta(\hat{\mathbf{z}}; \boldsymbol{\epsilon} | \mathbf{x}) \tag{D.1}$$

D.1.2 Mutual Information lower bound

Here, we show the full derivation of the lower bound for negative mutual-information. We derive the lower bound for the general case where there are both observed and unobserved confounders. A simple lower bound can be obtained by using $\mathbf{H}(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, \mathbf{x}) \geq \mathbf{H}(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, \mathbf{t}, \mathbf{x})$, but this cannot be made tight unless $\boldsymbol{\epsilon}$ completely determines $\mathbf{t}$. Therefore, we cannot guarantee independence unless the data at hand is not confounded. Instead we introduce two auxiliary distributions $r_v(\hat{\mathbf{z}} \mid \mathbf{x})$ and $p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, \mathbf{x})$, following the work in variational inference [250, 251, 252, 253] and causal inference [150].

We let $F(\mathbf{t}, \mathbf{x}, \boldsymbol{\epsilon}, \mathbf{y})$ be the true data distribution and $q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}, \mathbf{x} = x)$ be the control function distribution. We overload notation and also use q_θ to refer to any distribution that involves operations with $q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}, \mathbf{x} = x)$. We use $\overset{c}{=}$ to denote that the LHS and RHS are equal up to constants that are ignored during optimization. In the following, both $\mathbf{H}(\mathbf{t}, \boldsymbol{\epsilon} \mid \mathbf{x} = x)$, $\mathbf{H}(\boldsymbol{\epsilon} | \mathbf{x} = x)$ are constants with respect to the parameters of interest ϕ, θ, v and we will drop them from the lower bound when encountered. For a given $\mathbf{x} = x$, we lower-bound the negative instantaneous conditional mutual information:

$$\begin{aligned}
-\lambda \mathbf{I}(\hat{\mathbf{z}}; \boldsymbol{\epsilon} \mid x) &= -\lambda \mathrm{KL}\left[q_\theta(\hat{\mathbf{z}}, \boldsymbol{\epsilon} \mid x) \parallel q_\theta(\hat{\mathbf{z}} \mid x) F(\boldsymbol{\epsilon} \mid x)\right] \\
&= -\lambda \left[\mathbb{E}_{q_\theta(\boldsymbol{\epsilon}, \hat{\mathbf{z}} \mid x)} \left[\log q_\theta(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, x) - \log F(\boldsymbol{\epsilon} \mid x)\right]\right] \\
&= -\lambda \left[\mathbb{E}_{q_\theta(\boldsymbol{\epsilon}, \hat{\mathbf{z}} \mid x)} \left[\log q_\theta(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, x)\right] + \mathbf{H}(\boldsymbol{\epsilon} \mid x)\right] \\
&\overset{c}{=} -\lambda \Big[\mathbb{E}_{q_\theta(\boldsymbol{\epsilon}, \hat{\mathbf{z}} \mid x)} \Big[\mathrm{KL}\left[q_\theta(\hat{\mathbf{z}} \mid x) \parallel q_\theta(\hat{\mathbf{z}} \mid x)\right] + \mathrm{KL}\left[q_\theta(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x) \parallel q_\theta(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x)\right] + \log q_\theta(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, x)\Big]\Big]
\end{aligned}$$

$$
\begin{aligned}
&\geq -\lambda\Big[\mathbb{E}_{q_\theta(\boldsymbol{\epsilon},\hat{\mathbf{z}} \mid x)}\Big[\mathrm{KL}\left[q_\theta(\hat{\mathbf{z}} \mid x) \parallel r_\nu(\hat{\mathbf{z}} \mid x)\right] + \mathrm{KL}\left[q_\theta(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x) \parallel p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x)\right] + \log q_\theta(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, x)\Big]\Big] \\
&= -\lambda\Big[\mathbb{E}_{q_\theta(\boldsymbol{\epsilon},\hat{\mathbf{z}} \mid x)}\Big[\log\left[q_\theta(\hat{\mathbf{z}}, \boldsymbol{\epsilon} \mid x)\right] + \mathbb{E}_{q_\theta(\mathbf{t} \mid \boldsymbol{\epsilon},\hat{\mathbf{z}},x)} \log q_\theta(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x) \\
&\qquad - \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid x)} \log r_\nu(\hat{\mathbf{z}} \mid x) - \mathbb{E}_{q_\theta(\mathbf{t},\boldsymbol{\epsilon},\hat{\mathbf{z}} \mid x)} \log p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x)\Big] \\
&= -\lambda\Big[\mathbb{E}_{q_\theta(\boldsymbol{\epsilon},\hat{\mathbf{z}},\mathbf{t} \mid x)} \log\left[q_\theta(\hat{\mathbf{z}}, \boldsymbol{\epsilon}, \mathbf{t} \mid x)\right] - \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid x)} \log r_\nu(\hat{\mathbf{z}} \mid x) - \mathbb{E}_{q_\theta(\mathbf{t},\boldsymbol{\epsilon},\hat{\mathbf{z}} \mid x)} \log p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x)\Big] \\
&= -\lambda\Big[\mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon} \mid x)}\mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon},\mathbf{t},x)} \log\left[q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}, x) - \log p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x)\right] - \mathbf{H}(\mathbf{t}, \boldsymbol{\epsilon} \mid x) \\
&\qquad - \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid x)} \log r_\nu(\hat{\mathbf{z}} \mid x)\Big] \\
&\overset{c}{=} -\lambda\mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon} \mid x)}\left[\mathrm{KL}\left[q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}, x) \parallel r_\nu(\hat{\mathbf{z}} \mid x)\right] - \mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon},\mathbf{t},x)} \log p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, x)\right],
\end{aligned}
$$

where the hidden term $-\lambda\left[\mathbf{H}(\boldsymbol{\epsilon} \mid \mathbf{x} = x) - \mathbf{H}(\mathbf{t}, \boldsymbol{\epsilon} \mid \mathbf{x} = x)\right]$ is a constant for a given instance of the problem. We do not need access to the distribution $\mathbf{t}, \hat{\mathbf{z}}, \boldsymbol{\epsilon} \mid \mathbf{x} = x$ because the information that we lower bounded, $\mathbf{I}(\hat{\mathbf{z}}; \boldsymbol{\epsilon} \mid \mathbf{x} = x)$, is averaged over $\mathbf{x} = x$ in our objective. Recall that $p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, \mathbf{x} = x)$ is the reconstruction term in VDE. This lower bound is tight when the introduced KL terms are 0, which occurs when $r_\nu(\hat{\mathbf{z}} \mid \mathbf{x} = x) = q_\theta(\hat{\mathbf{z}} \mid \mathbf{x} = x)$ and $p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \mathbf{x} = x, \hat{\mathbf{z}}) = q_\theta(\mathbf{t} \mid \boldsymbol{\epsilon}, \mathbf{x} = x, \hat{\mathbf{z}})$. This means that if the models p_ϕ, r_ν are rich enough, the gap between the lower bound and mutual information can be optimized to be zero. The second term $\mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon},\mathbf{t},\mathbf{x}=x)} \log p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, \mathbf{x} = x)$ is the same as the reconstruction likelihood. Thus substituting the lower bound into the full objective with given covariates gives

$$
\mathbb{E}_{F(\mathbf{t},\boldsymbol{\epsilon},\mathbf{x})}\left[(1+\lambda)\mathbb{E}_{q_\theta(\hat{\mathbf{z}} \mid \mathbf{t},\boldsymbol{\epsilon},\mathbf{x})} \log p_\phi(\mathbf{t} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, \mathbf{x}) - \lambda\mathrm{KL}\left[q_\theta(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}, \mathbf{x}) \parallel r_\nu(\hat{\mathbf{z}} \mid \mathbf{x})\right]\right]
$$

Optimization for variational decoupling (VDE). The VDE optimization involves the expectations of distributions with parameters with respect to a distribution that also has parameters. For distributions that are not being integrated against, we can move the gradient inside the expectation. For distributions that are integrated against, score-function methods provide a general tool to compute stochastic gradients; Glasserman [254], Williams [255], Ranganath et al.

[256], Mnih and Gregor [257]. In our experiments, we let the control function be a categorical variable. This allows us to marginalize out the control function and compute the gradient.

D.1.3 Proof of Theorem 1

Theorem 1. ***(Meta-identification result for general control functions)***

Let $F(\mathbf{t}, \boldsymbol{\epsilon}, \mathbf{y})$ be the true data distribution. Let control function $\hat{\mathbf{z}}$ be sampled conditionally on $\mathbf{t}, \boldsymbol{\epsilon}$. Let $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}) = q(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})F(\mathbf{t}, \boldsymbol{\epsilon})$ be the joint distribution over $\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}$. Further, let g be a deterministic function and $\boldsymbol{\delta}$ be independent noise such that $\mathbf{t} = g(\mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\delta})$ and let the implied true joint be $F'(\mathbf{t}, \mathbf{z}, \boldsymbol{\delta})$. Assume the following:

1. *(A1) $\hat{\mathbf{z}}$ satisfies the **reconstruction** property: $\exists d,\ \hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon} \sim q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}) \implies \mathbf{t} = d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$.*

2. *(A2) The IV is **jointly independent** of control function, true confounder, and noise $\boldsymbol{\delta}$: $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}}, \boldsymbol{\delta})$.*

3. *(A3) **Strong** IV. For any compact $B \subseteq supp(\mathbf{t})$, $\exists c_B$ s.t. a.e. $t \in B$, $F'(\mathbf{t} = t \mid \mathbf{z}, \boldsymbol{\delta}) \geq c_B > 0$.*

Then, the control function $\hat{\mathbf{z}}$ satisfies ignorability and positivity:

$$q(\mathbf{y} \mid \mathbf{t} = t, \hat{\mathbf{z}}) = q(\mathbf{y} \mid \mathrm{do}(\mathbf{t} = t), \hat{\mathbf{z}}) \qquad \text{a.e. in } supp(\mathbf{t}) \quad q(\hat{\mathbf{z}}) > 0 \implies q(\mathbf{t} = t \mid \hat{\mathbf{z}}) > 0.$$

Therefore, the true causal effect is uniquely determined by $q(\hat{\mathbf{z}}, \mathbf{t}, \mathbf{y})$ for almost every $t \in supp(\mathbf{t})$:

$$\mathbb{E}_{\hat{\mathbf{z}}}[\mathbf{y} \mid \mathbf{t} = t, \hat{\mathbf{z}}] = \mathbb{E}_{\hat{\mathbf{z}}}[\mathbf{y} \mid \mathrm{do}(\mathbf{t} = t), \hat{\mathbf{z}}] = \mathbb{E}[\mathbf{y} \mid \mathrm{do}(\mathbf{t} = t)].$$

We prove this for the setting without covariates. The proof adapts to the setting with covariates (observed confounders) by conditioning all terms on them.

Proof. (Theorem 5) The proof shows that reconstruction (A1) and joint independence (A2) together imply ignorability, and strong IV (A3) together with the joint independence (A2) imply positivity.

Ignorability. To establish ignorability we need to show that $\mathbf{y}_t \perp\!\!\!\perp \mathbf{t} \mid \hat{\mathbf{z}}$ where $\mathbf{y}_t$ is the potential outcome for a unit when the treatment given is $\mathbf{t} = t$. The outcome $\mathbf{y}$ is constructed from the potential outcomes by indexing the one $\mathbf{y}_{t^*}$ corresponding to the observed treatment $\mathbf{t} = t^*$.

By assumption A2, we have the joint independence $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}})$ which implies

$$\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}}) \quad \Longrightarrow \quad \boldsymbol{\epsilon} \perp\!\!\!\perp \mathbf{z} \mid \hat{\mathbf{z}} = \hat{z} \quad \forall \hat{z} \in \mathrm{supp}(\hat{\mathbf{z}}).$$

Note that by the reconstruction property (from assumption A1) $\mathbf{t} = d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$. So given $\hat{\mathbf{z}}$, $\mathbf{t}$ is purely a function of $\boldsymbol{\epsilon}$. Thus, given $\hat{\mathbf{z}}$, $\mathbf{t}$ satisfies the same conditional independence as $\boldsymbol{\epsilon}$: $\boldsymbol{\epsilon} \perp\!\!\!\perp \mathbf{z} \mid \hat{\mathbf{z}}$. Using this, we have

$$\boldsymbol{\epsilon} \perp\!\!\!\perp \mathbf{z} \mid \hat{\mathbf{z}} \implies d(\hat{\mathbf{z}}, \boldsymbol{\epsilon}) \perp\!\!\!\perp \mathbf{z} \mid \hat{\mathbf{z}} \implies \mathbf{t} \perp\!\!\!\perp \mathbf{z} \mid \hat{\mathbf{z}}.$$

The potential outcome $\mathbf{y}_t$ depends only on $\mathbf{z}$ and some noise $\boldsymbol{\eta}$ that is jointly independent of all other variables. This means for some function m_t such that $\mathbf{y}_t = m_t(\mathbf{z}, \boldsymbol{\eta})$.

$$\mathbf{t} \perp\!\!\!\perp \mathbf{z} \ \mid \hat{\mathbf{z}} \implies \mathbf{t} \perp\!\!\!\perp m_t(\mathbf{z}, \boldsymbol{\eta}) \mid \hat{\mathbf{z}} \implies \mathbf{t} \perp\!\!\!\perp \mathbf{y}_t \mid \hat{\mathbf{z}}.$$

This shows ignorability.

Strength of IV and Positivity. Positivity means that for almost every $t \in \mathrm{supp}(\mathbf{t})$,

$$q(\hat{\mathbf{z}}) > 0, \implies q(\mathbf{t} = t \mid \hat{\mathbf{z}}) > 0.$$

We start with $q(\mathbf{t} \mid \hat{\mathbf{z}})$ and expand it as an integral over the full joint.

$$\begin{aligned}
q(\mathbf{t} \mid \hat{\mathbf{z}}) &= \int q(\mathbf{t} \mid \mathbf{z} = z, \hat{\mathbf{z}}, \boldsymbol{\epsilon} = \epsilon, \boldsymbol{\delta} = \delta, \mathbf{t}) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z} = z, \hat{\mathbf{z}}, \boldsymbol{\delta} = \delta) q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta d\epsilon \\
&= \int q(\mathbf{t} \mid \mathbf{z} = z, \boldsymbol{\epsilon} = \epsilon, \boldsymbol{\delta} = \delta) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z} = z, \hat{\mathbf{z}}, \boldsymbol{\delta} = \delta) q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta d\epsilon \\
&\qquad \{\text{by } \mathbf{t} = g(\mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\delta})\} \\
&= \int q(\mathbf{t} \mid \mathbf{z} = z, \boldsymbol{\epsilon} = \epsilon, \boldsymbol{\delta} = \delta) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z} = z, \boldsymbol{\delta} = \delta) q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta d\epsilon \\
&\qquad \{\text{by A2: } \boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}}, \boldsymbol{\delta})\} \\
&= \int \left[\int q(\mathbf{t} \mid \mathbf{z} = z, \boldsymbol{\epsilon} = \epsilon, \boldsymbol{\delta} = \delta) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z} = z, \boldsymbol{\delta} = \delta) d\epsilon \right] q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta \\
&= \int F'(\mathbf{t} \mid \mathbf{z} = z, \boldsymbol{\delta} = \delta) q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta
\end{aligned} \tag{D.2}$$

Note that $q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}})$ is a valid density over $(\mathbf{z} = z, \boldsymbol{\delta} = \delta)$ [*]. Under assumption A3, for any compact set $B \subseteq \text{supp}(\mathbf{t})$ and for almost every $t \in B$,

$$\begin{aligned}
q(\mathbf{t} = t \mid \hat{\mathbf{z}}) &= \int F'(\mathbf{t} = t \mid \mathbf{z} = z, \boldsymbol{\delta} = \delta) q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta \\
&\geq c_B \int q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) dz d\delta \\
&= c_B > 0
\end{aligned} \tag{D.3}$$

However, almost every $t \in \text{supp}(\mathbf{t})$ is contained in some compact subset $B \subseteq \text{supp}(\mathbf{t})$. Thus, eq. (D.3) holds for almost every $t \in \text{supp}(\mathbf{t})$, meaning that positivity is satisfied.

[*] If $q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}}) = 0$ everywhere then no pair $(\mathbf{z} = z, \boldsymbol{\delta} = \delta)$ maps to $\hat{\mathbf{z}}$ and $\hat{\mathbf{z}}$ cannot be observed and we cannot condition on it. But $\hat{\mathbf{z}}$ is constructed explicitly as part of the algorithm, so it's observed. Thus $q(\mathbf{z} = z, \boldsymbol{\delta} = \delta \mid \hat{\mathbf{z}})$ is a valid conditional density.

Computing the causal effect. Given ignorability and positivity, the true causal effect (a.e. in $\text{supp}(\mathbf{t})$) is determined as a property of the distribution $q(\hat{\mathbf{z}}, \mathbf{t}, \mathbf{y})$ as follows:

$$\mathbb{E}_{q(\hat{\mathbf{z}})}\mathbb{E}[\mathbf{y} \mid \hat{\mathbf{z}}, \mathbf{t} = t] = \mathbb{E}_{q(\hat{\mathbf{z}})}\mathbb{E}[\mathbf{y} \mid \hat{\mathbf{z}}, \text{do}(\mathbf{t} = t)] = \mathbb{E}[\mathbf{y} \mid \text{do}(\mathbf{t} = t)]$$

□

Assumptions for continuous $\mathbf{t}$. When $\mathbf{t}$ has non-zero density rather than non-zero probability given the general control function, the true expected outcome being continuous everywhere as a function of the treatment is a sufficient condition for the causal effect estimation for almost all treatment values.

D.1.4 Marginal Independence does not imply joint independence

Here, we build an example of a function of two independent variables $\mathbf{a}, \mathbf{b}$ that is marginally independent of both. Let 1_e be one if e is true and zero if not,

$$\begin{aligned}
&\mathbf{a}, \mathbf{b} \sim \text{uniform}(0, 1),\\
&\mathbf{c}(\mathbf{a}, \mathbf{b}) = 1_{\mathbf{a}+\mathbf{b}>1}(\mathbf{a} + \mathbf{b} - 1) + 1_{\mathbf{a}+\mathbf{b}\leq 1}(\mathbf{a} + \mathbf{b}).
\end{aligned}$$

First, $\mathbf{c}$ is marginally a uniform variable.[†] The distribution $\mathbf{c} \mid \mathbf{a} = x$ can be obtained by translating the distribution of $\mathbf{b}$ up by x, then translating the part greater than one down to zero, meaning $\mathbf{c} \mid \mathbf{a}$ is uniformly distributed. Thus $p(\mathbf{c} \mid \mathbf{a}) = p(\mathbf{c})$ meaning $\mathbf{c} \perp\!\!\!\perp \mathbf{a}$. However, $\mathbf{c}$ is a deterministic function of $\mathbf{a}$ and $\mathbf{b}$. Therefore, while $\mathbf{c} \mid \mathbf{a}$ is uniformly distributed, $\mathbf{c} \mid (\mathbf{a}, \mathbf{b})$ is a dirac-delta distribution, meaning $p(\mathbf{c} \mid \mathbf{a}, \mathbf{b}) \neq p(\mathbf{c} \mid \mathbf{a})$ implying $\mathbf{c} \not\perp\!\!\!\perp \mathbf{a} \mid \mathbf{b}$. Note that $\mathbf{b}$ can be constructed back from $\mathbf{c}, \mathbf{a}$ up to measure-zero as $\mathbf{b} = \mathbf{c} - \mathbf{a}$ if $\mathbf{c} > \mathbf{a}$ and $\mathbf{b} = \mathbf{c} - \mathbf{a} + 1$ if $\mathbf{c} \leq \mathbf{a}$; i.e., $\mathbf{c}$ is almost

[†] $P(\mathbf{c} < x) = P(\mathbf{a} + \mathbf{b} < x) + P(1 < \mathbf{a} + \mathbf{b} < 1 + x) = 0.5(x^2 - 1) + 1 - 0.5(1 - x)^2 = x.$

everywhere invertible for each fixed $\mathbf{a} = a$.

This construction with uniform random variables can be generalized to other continuous distributions by inverse transform sampling. Any marginal density of $\mathbf{a}, \mathbf{b}$ can be bijectively mapped to a uniform density over $[0, 1]$. Then $\mathbf{c}$ can be computed as above and then $\mathbf{a}, \mathbf{b}, \mathbf{c}$ can be bijectively mapped back; $\mathbf{c}$ could be mapped back with the CDF of $\mathbf{b}$. Conditional dependence is unaffected by bijective transformations and therefore the issue remains. Similar constructions exist with discrete random variables. In general, assumptions on the true data generating process will be needed to ensure joint independence.

D.1.5 From additive treatment processes to joint independence

Consider treatment processes of the form $\mathbf{t} = h(\mathrm{z}, \boldsymbol{\delta}) + g(\boldsymbol{\epsilon})$. Let the reconstruction map be additive:

$$\mathbf{t} = h'(\hat{\mathrm{z}}) + g'(\boldsymbol{\epsilon}).$$

Consider the random variable $\mathbf{t} - \mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}]$ which is sampled as follows: $\boldsymbol{\epsilon} \sim h(\boldsymbol{\epsilon}), \mathbf{z} \sim h(\mathrm{z}), \boldsymbol{\delta} \sim h(\boldsymbol{\delta})$ and $\mathbf{t} - \mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = h(\mathrm{z}, \boldsymbol{\delta}) + g(\boldsymbol{\epsilon}) - \mathbb{E}_{\mathrm{z},\boldsymbol{\delta}}[h(\mathrm{z}, \boldsymbol{\delta}) + g(\boldsymbol{\epsilon})]$. We show that $h'(\hat{\mathrm{z}})$ determines $h(\mathrm{z}, \boldsymbol{\delta})$ by expressing the random variable $\mathbf{t} - \mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}]$ in terms of z, $\boldsymbol{\delta}$ and $\hat{\mathrm{z}}$

$$h'(\hat{\mathrm{z}}) - \mathbb{E}_{\hat{\mathrm{z}}}[h'(\hat{\mathrm{z}})] = \mathbf{t} - \mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = h(\mathrm{z}, \boldsymbol{\delta}) - \mathbb{E}_{\mathrm{z},\boldsymbol{\delta}}[h(\mathrm{z}, \boldsymbol{\delta})].$$

Therefore for some constant c, $h'(\hat{\mathrm{z}}) = h(\mathrm{z}, \boldsymbol{\delta}) + c$. By the independence, $\hat{\mathrm{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$, we have

$$q(\hat{\mathrm{z}}, h(\mathrm{z}, \boldsymbol{\delta}) \mid \boldsymbol{\epsilon}) = q(\hat{\mathrm{z}}, h'(\hat{\mathrm{z}}) - c \mid \boldsymbol{\epsilon}) = q(\hat{\mathrm{z}}, h'(\hat{\mathrm{z}}) - c) = q(\hat{\mathrm{z}}, h(\mathrm{z}, \boldsymbol{\delta})).$$

Thus we have $(\hat{\mathbf{z}}, h(\mathbf{z}, \boldsymbol{\delta})) \perp\!\!\!\perp \boldsymbol{\epsilon}$. See lemma 14 for the proof that $(\hat{\mathbf{z}}, h(\mathbf{z}, \boldsymbol{\delta})) \perp\!\!\!\perp \boldsymbol{\epsilon}$ implies the joint independence $\boldsymbol{\epsilon} \perp\!\!\!\perp (\hat{\mathbf{z}}, \mathbf{z}, \boldsymbol{\delta})$ for any treatment process $\mathbf{t} = g(\boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta}))$, including $\mathbf{t} = g(\boldsymbol{\epsilon}) + h(\mathbf{z}, \boldsymbol{\delta})$.

D.1.6 Joint independence for treatments with additional noise

General control functions for treatments of the form $\mathbf{t} = g(\boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta}))$, unlike $\mathbf{t} = g(\boldsymbol{\epsilon}, \mathbf{z})$, require a stronger joint independence $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}}, \boldsymbol{\delta})$ to guarantee ignorability (A2, theorem 5). The structural assumptions — that helped guarantee $\boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}})$ above — can guarantee $\boldsymbol{\epsilon} \perp\!\!\!\perp (h(\mathbf{z}, \boldsymbol{\delta}), \hat{\mathbf{z}})$. Here, we show that $\boldsymbol{\epsilon} \perp\!\!\!\perp (h(\mathbf{z}, \boldsymbol{\delta}), \hat{\mathbf{z}}) \implies \boldsymbol{\epsilon} \perp\!\!\!\perp (\mathbf{z}, \hat{\mathbf{z}}, \boldsymbol{\delta})$ in such settings.

Lemma 14. *Consider treatment process* $\mathbf{t} = g(\boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta}))$ *and the joint independence* $(\hat{\mathbf{z}}, h(\mathbf{z}, \boldsymbol{\delta})) \perp\!\!\!\perp \boldsymbol{\epsilon}$ *holds. Then, if* $\hat{\mathbf{z}} = e(\mathbf{t}, \boldsymbol{\epsilon})$*, the joint independence* $(\hat{\mathbf{z}}, \mathbf{z}, \boldsymbol{\delta}) \perp\!\!\!\perp \boldsymbol{\epsilon}$ *holds.*

Proof. We begin by showing $q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\delta}) = q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta}))$:

$$\begin{aligned}
q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\delta}) &= \int q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\epsilon}, \mathbf{t} = t, \boldsymbol{\delta}) q(\mathbf{t} = t \mid \boldsymbol{\epsilon}, \mathbf{z}, \boldsymbol{\delta}) dt \quad \{\text{full joint expansion}\} \\
&= \int q(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, \mathbf{t} = t) q(\mathbf{t} = t \mid \boldsymbol{\epsilon}, \mathbf{z}, \boldsymbol{\delta}) dt \quad \{\hat{\mathbf{z}} \perp\!\!\!\perp (\mathbf{z}, \boldsymbol{\delta}) \mid \boldsymbol{\epsilon}, \mathbf{t} = t\} \\
&= \int q(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, \mathbf{t} = t) q(\mathbf{t} = t \mid \boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta})) dt \quad \{\mathbf{t} = g(\boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta}))\} \\
&= \int q(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, \mathbf{t} = t, h(\mathbf{z}, \boldsymbol{\delta})) q(\mathbf{t} = t \mid \boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta})) dt \quad \{\hat{\mathbf{z}} \perp\!\!\!\perp h(\mathbf{z}, \boldsymbol{\delta}) \mid \boldsymbol{\epsilon}, \mathbf{t} = t\} \\
&= q(\hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, h(\mathbf{z}, \boldsymbol{\delta})) \\
&= q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta})) \quad \{(\hat{\mathbf{z}}, h(\mathbf{z}, \boldsymbol{\delta})) \perp\!\!\!\perp \boldsymbol{\epsilon}\}
\end{aligned} \tag{D.4}$$

Integrating both sides with respect to $q(\boldsymbol{\epsilon} \mid \mathbf{z}, \boldsymbol{\delta})$ we get

$$\int q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta})) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z}, \boldsymbol{\delta}) d\epsilon = \int q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\epsilon} = \epsilon, \boldsymbol{\delta}) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z}, \boldsymbol{\delta}) d\epsilon = q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\delta}) \tag{D.5}$$

Now, the LHS in eq. (D.5) is

$$\int q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta}))q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{z}, \boldsymbol{\delta})d\epsilon = q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta})) \implies q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta})) = q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\delta}).$$

This means

$$q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\delta}) = q(\hat{\mathbf{z}} \mid h(\mathbf{z}, \boldsymbol{\delta})) = q(\hat{\mathbf{z}} \mid \mathbf{z}, \boldsymbol{\delta})$$

Thus $(\hat{\mathbf{z}}, h(\mathbf{z}, \boldsymbol{\delta}))\perp\!\!\!\perp\boldsymbol{\epsilon}$ implies the joint independence $(\hat{\mathbf{z}}, \mathbf{z}, \boldsymbol{\delta})\perp\!\!\!\perp\boldsymbol{\epsilon}$. □

Note. The proof above shows that we can recover a control function that satisfies ignorability. In this additive setting with finite support however, both the control function and the true confounder violate another fundamental assumption in causal estimation: *positivity*. To see this violation of positivity notice that $p(\mathbf{t} > a + \max_{\epsilon \in \mathrm{supp}(\boldsymbol{\epsilon})} g(\epsilon) \mid h(\mathbf{z}, \boldsymbol{\delta}) = a) = 0$ for any a such that $p(\mathbf{t} > a + \max_{\epsilon \in \mathrm{supp}(\boldsymbol{\epsilon})} g(\epsilon)) > 0$ and $p(h(\mathbf{z}, \boldsymbol{\delta}) = a) > 0$. When positivity is violated, further assumptions are needed to compute causal effects on the whole support of $\mathbf{t}$ in general. Without further assumptions, effects can only be computed on a compact subset of $B \subseteq \mathrm{supp}(\mathbf{t})$ within which positivity holds.

D.1.7 From monotonic treatment processes to joint independence

Imbens and Newey [143] explored identification for settings where the outcome process is non-separable but the treatment is a strictly monotonic function of the unobserved confounder. We show that if the reconstruction map $d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$ reflects this monotonicity condition and $\hat{\mathbf{z}}\perp\!\!\!\perp\boldsymbol{\epsilon}$, the control function is determined by the true confounder and therefore joint independence holds. In VDE, the decoder would be monotonic to reflect this assumption.

Lemma 15. *Let $\boldsymbol{\epsilon}$ and $\mathbf{z}$ be the true IV and confounder respectively. Let $\mathbf{z}$ be a continuous scalar.*

1. *Assume that $\mathbf{z}$ has a continuous strictly monotonic CDF. Let the true treatment process be $\mathbf{t} = g(\boldsymbol{\epsilon}, \mathbf{z})$*

where g is strictly monotonic in the second argument.

2. *Let the control function be $\hat{\mathbf{z}} = e(\boldsymbol{\epsilon}, \mathbf{t})$ and let $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$. Let reconstruction map be d where $\mathbf{t} = d(\boldsymbol{\epsilon}, \hat{\mathbf{z}})$. Let $e(\cdot, \cdot)$ and $d(\cdot, \cdot)$ be strictly monotonic in the second argument*[‡].
3. *Assume that the functions g, e, d are continuous in the second argument and exist for almost every value in the first argument.*

Then, the control function $\hat{\mathbf{z}}$ can be expressed as a deterministic function of the true confounder $\mathbf{z}$.

Proof. First, note that $\mathbf{t}$ can be written as a function of $\boldsymbol{\epsilon}$ and a uniform random variable $\mathbf{u}$ using the CDF-inverse trick. Let $H(z) = F(\mathbf{z} \leq z)$. By strict monotonicity and continuity of H, H^{-1} exists and $\mathbf{z} = H^{-1}(\mathbf{u})$ for a uniform random variable $\mathbf{u} \perp\!\!\!\perp \boldsymbol{\epsilon}$:

$$\mathbf{t} = g(\boldsymbol{\epsilon}, \mathbf{z}) = g(\boldsymbol{\epsilon}, H^{-1}(\mathbf{u})) = \hat{g}(\boldsymbol{\epsilon}, \mathbf{u}).$$

Note that H^{-1} is strictly monotonic. So, $\hat{g}$ is a strictly monotonic function in the second argument.

Second, due to $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$, the conditional CDF of $\hat{\mathbf{z}} \mid \boldsymbol{\epsilon} = \epsilon$ is the same as the marginal CDF as $\hat{\mathbf{z}}$ for almost every value $\epsilon \in \text{supp}(\boldsymbol{\epsilon})$; let's call this CDF $\hat{H}$. By the definition $\hat{\mathbf{z}} = e(\boldsymbol{\epsilon}, \mathbf{t})$ we can express $\hat{\mathbf{z}} = e(\boldsymbol{\epsilon}, \hat{g}(\boldsymbol{\epsilon}, \mathbf{u}))$. Now, $e(\cdot, \cdot), \hat{g}(\cdot, \cdot)$ are both continuous and strictly monotonic in the second argument. So, $\hat{\mathbf{z}}$'s CDF $\hat{H}$ is also strictly monotonic and $\hat{H}^{-1}$ exists and is again strictly monotonic. Therefore, for almost any $\epsilon \in \text{supp}(\boldsymbol{\epsilon})$, we can construct a new uniform random variable by applying $\hat{\mathbf{z}}$'s CDF $\hat{H}$ to $\hat{\mathbf{z}}$:

$$\mathbf{v} = \hat{H}(\hat{\mathbf{z}}) = \hat{H}(e(\boldsymbol{\epsilon}, \hat{g}(\boldsymbol{\epsilon}, \mathbf{u}))).$$

For simplicity, let $\mathbf{v} = J(\boldsymbol{\epsilon}, \mathbf{u})$. Note $J(\cdot, u)$ is strictly monotonic in u by strict monotonicity of $\hat{H}, \hat{g}$ in their second arguments. So, we can write $\mathbf{u}$'s CDF in terms of $\mathbf{v}$'s CDF:

$$a = P(\mathbf{u} < a) = P(\mathbf{v} < J(\epsilon, a)) = J(\epsilon, a).$$

[‡] Note that $e(\epsilon, \cdot) = d^{-1}(\epsilon, \cdot)$. Then, monotonicity of d in the second argument implies the same for e.

This means that $J(\epsilon, a)$ is an identity function for almost any $\epsilon \in \text{supp}(\boldsymbol{\epsilon})$.

Finally, we can write $\hat{\mathbf{z}}$ as a function of $\mathbf{z}$ for almost any $\epsilon \in \text{supp}(\boldsymbol{\epsilon})$, completing the proof:

$$\hat{\mathbf{z}} = \hat{H}^{-1}(J(\epsilon, H(\mathbf{z}))) = \hat{H}^{-1}(H(\mathbf{z}))$$

□

D.1.8 Comparion against other identification results

Imbens and Newey [143] consider non-separable outcome processes, i.e. $\mathbf{y} = f(\mathbf{t}, \mathbf{z})$ and construct control functions by assuming that 1) treatment is a strictly monotonic function of the confounder, 3) the confounder is continuous with a strictly monotonic CDF, and 2) positivity holds for $\mathbf{t}$ with respect to $\mathbf{z}$. These assumptions also lead to identification with general control functions due to the following: a) the positivity assumption is equivalent to the strong IV assumption and b) like additivity, the strict monotonicity assumption reflected in the reconstruction map $d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})$ as a function of $\hat{\mathbf{z}}$ helps guarantee joint independence; see appendix D.1.7.

2SLS requires the outcome process to be additive, $\mathbf{y} = f(\mathbf{t}) + \mathbf{z}$. Further, 2SLS needs a "completeness" property: the causal effect function and IV are correlated [134]. While joint independence may not be guaranteed by the completeness condition, it can be guaranteed in certain settings that violate completeness. An example is multiplicative treatment $\mathbf{t} = \mathbf{z} * \boldsymbol{\epsilon}$ with $\mathbf{z} \sim \mathcal{N}(0, 1)$ and a linear outcome; 2SLS fails because $\mathbb{E}[\mathbf{t}\boldsymbol{\epsilon}] = 0$. When joint independence can be guaranteed and the IV is strong, identification with general control functions does not require structural restrictions like additivity of the outcome process that both 2SLS and CFN rely on.

D.1.9 Estimation error bounds

We give an example of how violations in reconstruction and independence affect errors in effects.

D.1.9.1 GCFN's estimation error in additive treatment process

Theorem 2. *Assume an additive treatment process* $\mathbf{t} = \mathbf{z} + g(\boldsymbol{\epsilon})$ *where* g *is an* L_g*-Lipschitz function, and* $\mathbb{E}_{F(\mathbf{z})}\mathbf{z} = 0$*. Let* $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = t, \mathbf{z} = z] = f(t, z)$ *be an* L*-Lipschitz function in* z *for any* t*. Further,*

1. *let reconstruction error be non-zero but bounded* $\mathbb{E}_{q(\mathbf{t},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{t} - \hat{\mathbf{z}} - g'(\boldsymbol{\epsilon}))^2 \leq \delta$*. Assume that* g' *is also* L_g*-Lipschitz. Further, let* $\mathbb{E}_{q(\hat{\mathbf{z}})}\hat{\mathbf{z}} = 0$*, and* $\mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}| < \infty$*.*
2. *Assume* $\boldsymbol{\epsilon} \not\perp\!\!\!\perp \hat{\mathbf{z}}$ *and let the dependence be bounded:* $\max_{\hat{z}} \mathcal{W}_1\left(q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}} = \hat{z}) \ \| \ F(\boldsymbol{\epsilon})\right) \leq \gamma$*.*

With the estimated and true causal effects as $\hat{\tau}(t) = \mathbb{E}_{\hat{\mathbf{z}}} f(t, \hat{\mathbf{z}})$ *and* $\tau(t) = E_{\mathbf{z}} f(t, \mathbf{z})$ *respectively,*

$$\mathbb{E}_{F(\mathbf{t})}|\hat{\tau}(\mathbf{t}) - \tau(\mathbf{t})| \leq L\sqrt{\delta + 4\gamma L_g \mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}|}.$$

Proof. Recall the true data distribution is $F(\mathbf{t}, \mathbf{z}, \boldsymbol{\epsilon})$ such that $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon}$ and the implied joint $q(\hat{\mathbf{z}}, \mathbf{t}, \mathbf{z}, \boldsymbol{\epsilon}) = q(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon})F(\mathbf{t}, \mathbf{z}, \boldsymbol{\epsilon})$. For any L-Lipschitz function $\ell(\epsilon)$:

$$\begin{aligned}
|\mathbb{E}_{q(\boldsymbol{\epsilon},\hat{\mathbf{z}})}\ell(\boldsymbol{\epsilon})\hat{\mathbf{z}}| &= |\mathbb{E}_{q(\hat{\mathbf{z}})}\left(\hat{\mathbf{z}}\mathbb{E}_{q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}})}\ell(\boldsymbol{\epsilon})\right) - \left(\mathbb{E}_{q(\hat{\mathbf{z}})F(\boldsymbol{\epsilon})}\hat{\mathbf{z}}\ell(\boldsymbol{\epsilon})\right)| \quad \{\mathbb{E}_{q(\hat{\mathbf{z}})}\hat{\mathbf{z}} = 0\} \\
&= \left|\mathbb{E}_{q(\hat{\mathbf{z}})}\left(\hat{\mathbf{z}}\left(\mathbb{E}_{q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}})}\ell(\boldsymbol{\epsilon}) - \mathbb{E}_{F(\boldsymbol{\epsilon})}\ell(\boldsymbol{\epsilon})\right)\right)\right| \\
&\leq \mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}|\left|\mathbb{E}_{q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}})}\ell(\boldsymbol{\epsilon}) - \mathbb{E}_{F(\boldsymbol{\epsilon})}\ell(\boldsymbol{\epsilon})\right| \qquad \text{(D.6)} \\
&\leq L\mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}|\mathcal{W}_1\left(q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}) \ \| \ F(\boldsymbol{\epsilon})\right) \\
&\leq \gamma L\mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}|.
\end{aligned}$$

Using the definition of the additive treatment process and the reconstruction error bound, $\mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{z}+$

$g(\boldsymbol{\epsilon}) - \hat{\mathbf{z}} - g'(\boldsymbol{\epsilon}))^2 = \mathbb{E}_{q(\mathbf{t},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{t} - \hat{\mathbf{z}} - g'(\boldsymbol{\epsilon}))^2 \leq \delta$. Now, we can bound error in $\hat{\mathbf{z}}$ approximating $\mathbf{z}$

$$\begin{aligned}
\delta &\geq \mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{z} - \hat{\mathbf{z}} + g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon}))^2 \\
&= \mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}})}(\mathbf{z} - \hat{\mathbf{z}})^2 + \mathbb{E}_{F(\boldsymbol{\epsilon})}(g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon}))^2 + 2\mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{z} - \hat{\mathbf{z}})(g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon})) \\
&\geq \mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}})}(\mathbf{z} - \hat{\mathbf{z}})^2 + 2\mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}},\boldsymbol{\epsilon})}(\mathbf{z} - \hat{\mathbf{z}})(g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon})) \\
&= \mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}})}(\mathbf{z} - \hat{\mathbf{z}})^2 + 2\mathbb{E}_{F(\mathbf{z})F(\boldsymbol{\epsilon})}\mathbf{z}(g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon})) - 2\mathbb{E}_{q(\hat{\mathbf{z}},\boldsymbol{\epsilon})}\hat{\mathbf{z}}(g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon})) \qquad \{\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon}\} \\
&= \mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}})}(\mathbf{z} - \hat{\mathbf{z}})^2 + 0 - 2\mathbb{E}_{q(\hat{\mathbf{z}},\boldsymbol{\epsilon})}\hat{\mathbf{z}}(g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon})) \qquad \{\mathbb{E}_{F(\mathbf{z})}\mathbf{z} = 0\} \\
&\geq \mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}})}(\mathbf{z} - \hat{\mathbf{z}})^2 - 4\gamma L_g \mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}| \quad \{g(\boldsymbol{\epsilon}) - g'(\boldsymbol{\epsilon}) \text{ is } 2L_g\text{-Lipschitz}\}
\end{aligned}$$

Thus, $\mathbb{E}_{q(\mathbf{z},\hat{\mathbf{z}})}(\mathbf{z} - \hat{\mathbf{z}})^2 \leq \delta + 4\gamma L_g \mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}|$. We bound the absolute error in causal effect due to using $\hat{\mathbf{z}}$ instead of $\mathbf{z}$

$$\begin{aligned}
\mathbb{E}_{\mathbf{t}}|\hat{\tau}(\mathbf{t}) - \tau(\mathbf{t})| &= \mathbb{E}_{\mathbf{t}}|\mathbb{E}_{q(\hat{\mathbf{z}})}f(\mathbf{t}, \hat{\mathbf{z}}) - \mathbb{E}_{F(\mathbf{z})}f(\mathbf{t}, \mathbf{z})| \\
&= \mathbb{E}_{\mathbf{t}}|\mathbb{E}_{q(\hat{\mathbf{z}},\mathbf{z})}\left(f(\mathbf{t}, \hat{\mathbf{z}}) - f(\mathbf{t}, \mathbf{z})\right)| \\
&\leq \mathbb{E}_{\mathbf{t}}\mathbb{E}_{q(\hat{\mathbf{z}},\mathbf{z})}\left|f(\mathbf{t}, \hat{\mathbf{z}}) - f(\mathbf{t}, \mathbf{z})\right| \\
&\leq \mathbb{E}_{\mathbf{t}} L \mathbb{E}_{q(\hat{\mathbf{z}},\mathbf{z})}|\hat{\mathbf{z}} - \mathbf{z}| \\
&\leq L\mathbb{E}_{\mathbf{t}}\sqrt{\mathbb{E}_{q(\hat{\mathbf{z}},\mathbf{z})}\left(\hat{\mathbf{z}} - \mathbf{z}\right)^2} \quad \text{(Cauchy-Schwarz)} \\
&\leq L\sqrt{\delta + 4\gamma L_g \mathbb{E}_{q(\hat{\mathbf{z}})}|\hat{\mathbf{z}}|}
\end{aligned} \tag{D.7}$$

When sample size goes to ∞, we can guarantee that reconstruction becomes perfect, meaning that $\delta \longrightarrow 0$, and that $\hat{\mathbf{z}} \perp\!\!\!\perp \boldsymbol{\epsilon}$ holds, meaning that $\gamma \longrightarrow 0$. Then, this error bound on effects becomes 0.

□

D.1.9.2 Bounding effect estimation error

Here, we show that if positivity holds for $\mathbf{t}$ w.r.t. $\mathbf{z}$, and $\mathbf{t}$ w.r.t. $\hat{\mathbf{z}}$, the residual confounding given $\hat{\mathbf{z}}$, i.e. $\mathbf{I}(\mathbf{z};\mathbf{t} \mid \hat{\mathbf{z}})$, controls the expected absolute error in effects if $q(\mathbf{z} \mid \hat{\mathbf{z}})$ is sufficiently concentrated.

Theorem 3. *Let $F(\mathbf{y}, \mathbf{t}, \mathbf{z}, \boldsymbol{\epsilon})$ be the true data distribution. Let $q(\mathbf{y}, \mathbf{t}, \hat{\mathbf{z}}) = \int F(\mathbf{y}, \mathbf{t}, \mathbf{z} = z, \epsilon) q(\hat{\mathbf{z}} \mid \mathbf{t}, \epsilon) dz d\epsilon$. With $\tau(t^*)$ and $\hat{\tau}(t^*)$ as the true and estimated causal effect of $do(\mathbf{t} = t^*)$ respectively, let $\omega(t^*) = |\hat{\tau}(t^*) - \tau(t^*)|$ be the error. We assume the following.*

1. *Assume that $\mathbf{t}$ satisfies positivity with respect to $\mathbf{z}$, and $\mathbf{t}$ satisfies positivity with respect to $\hat{\mathbf{z}}$.*
2. *Let $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = t, \mathbf{z} = z]$ where $\mathbb{E}$ is w.r.t. F, be an L_t-Lipschitz function of z, for any t.*
3. *Let $L := \sup_t L_t$. Let $W := \sup_{t,\hat{z}} {}^{F(\mathbf{t}=t)}/_{q(\mathbf{t}=t \mid \hat{\mathbf{z}}=\hat{z})}$.*
4. *Assume $q(\mathbf{z} \mid \hat{\mathbf{z}})$ satisfies the transportation inequality $T_1(\sigma^2/2)$ [258].*

Then, the expected absolute error in effects is bounded as: $\quad \mathbb{E}_{F(\mathbf{t})}\omega(\mathbf{t}) \leq \sigma L \sqrt{W \mathbf{I}(\mathbf{z};\mathbf{t} \mid \hat{\mathbf{z}})}$.

Proof. (of theorem 3) Positivity of $\mathbf{t}$ w.r.t. $\mathbf{z}$ implies the conditional expectation $\mathbb{E}[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^*]$ exists for all $z \in \operatorname{supp}(F(\mathbf{z})), t^* \in \operatorname{supp}(\mathbf{t})$. Positivity of $\mathbf{t}$ w.r.t. $\hat{\mathbf{z}}$ implies the conditional expectation $\mathbb{E}[\mathbf{y} \mid \hat{\mathbf{z}} = \hat{z}, \mathbf{t} = t^*]$ exists for all $\hat{z} \in \operatorname{supp}(F(\hat{\mathbf{z}})), t^* \in \operatorname{supp}(\mathbf{t})$. We begin by expanding the expectation $\mathbb{E}[\mathbf{y} \mid \hat{\mathbf{z}} = \hat{z}, \mathbf{t} = t^*]$ as an integral over the conditional $F(\mathbf{y} \mid \mathbf{z}, \mathbf{t}, \hat{\mathbf{z}}) q(\mathbf{z} \mid \mathbf{t}, \hat{\mathbf{z}})$.

$$\begin{aligned}
\mathbb{E}[y \mid \hat{\mathbf{z}} = \hat{z}, \mathbf{t} = t^*] &= \int \mathbb{E}\left[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^*, \hat{\mathbf{z}} = \hat{z}\right] q(\mathbf{z} = z \mid \mathbf{t} = t^*, \hat{\mathbf{z}} = \hat{z}) dz \\
&= \int \mathbb{E}\left[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^*\right] q(\mathbf{z} = z \mid \mathbf{t} = t^*, \hat{\mathbf{z}} = \hat{z}) dz \quad \{\text{by } \mathbf{y} \perp\!\!\!\perp \hat{\mathbf{z}} \mid \mathbf{t}, \mathbf{z}\},
\end{aligned}$$

where the inner expectation is with respect to the conditional distribution $F(\mathbf{y} \mid \mathbf{t}, \mathbf{z})$. Now, we prove the bound on $\omega(t^*)$ by expanding the true and estimated effects as expectations over $\mathbf{z}$:

$$\omega(t^*) = |\tau(t^*) - \hat{\tau}(t^*)|$$

$$
\begin{aligned}
&= \left| \int \left[F(\mathbf{z} = z) - \mathbb{E}_{q(\hat{\mathbf{z}})} q(\mathbf{z} = z \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \right] \mathbb{E}\left[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^* \right] dz \right| \\
&= L_{t^*} \left| \int \left[F(\mathbf{z} = z) - \mathbb{E}_{q(\hat{\mathbf{z}})} q(\mathbf{z} = z \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \right] \frac{\mathbb{E}\left[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^* \right]}{L_t^*} dz \right| \\
&= L_{t^*} \left| \int \left[\mathbb{E}_{q(\hat{\mathbf{z}})} \left(q(\mathbf{z} = z \mid \hat{\mathbf{z}}) - q(\mathbf{z} = z \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \right) \right] \frac{\mathbb{E}\left[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^* \right]}{L_t^*} dz \right| \\
&\leq L_{t^*} \mathbb{E}_{q(\hat{\mathbf{z}})} \left| \int \left[\left(q(\mathbf{z} = z \mid \hat{\mathbf{z}}) - q(\mathbf{z} = z \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \right) \right] \frac{\mathbb{E}\left[\mathbf{y} \mid \mathbf{z} = z, \mathbf{t} = t^* \right]}{L_t^*} dz \right| \\
&\leq L_{t^*} \mathbb{E}_{q(\hat{\mathbf{z}})} \mathcal{W}_1 \left(q(\mathbf{z} \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \;\|\; q(\mathbf{z} \mid \hat{\mathbf{z}}) \right) \\
&\qquad \left\{ {}^{\mathbb{E}[\mathbf{y} \mid \mathbf{z}=z, \mathbf{t}=t^*]}/_{L_{t^*}} \text{ is 1-Lipschitz} \right\} \\
&\leq L_{t^*} \mathbb{E}_{q(\hat{\mathbf{z}})} \sigma \sqrt{\mathrm{KL}\left[q(\mathbf{z} \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \,\|\, q(\mathbf{z} \mid \hat{\mathbf{z}}) \right]} \\
&\leq L_{t^*} \sigma \sqrt{\mathbb{E}_{q(\hat{\mathbf{z}})} \mathrm{KL}\left[q(\mathbf{z} \mid \mathbf{t} = t^*, \hat{\mathbf{z}}) \,\|\, q(\mathbf{z} \mid \hat{\mathbf{z}}) \right]} \quad \{\text{by Cauchy Schwarz}\},
\end{aligned}
$$

where the $\mathcal{W}_1$ term was bounded by **KL** by the assumption that $q(\mathbf{z} \mid \hat{\mathbf{z}})$ satisfies the transportation inequality $T_1(\sigma^2/2)$ [258]. Using $L = \sup_t L_t$ and $W = {}^{F(\mathbf{t}=t)}/_{q(\mathbf{t}=t \mid \hat{\mathbf{z}}=\hat{z})}$, we can bound the average absolute error

$$
\begin{aligned}
\mathbb{E}_{F(\mathbf{t})} \omega(\mathbf{t}) &\leq \sigma \mathbb{E}_{F(\mathbf{t})} L_{\mathbf{t}} \sqrt{\mathbb{E}_{q(\hat{\mathbf{z}})} \mathrm{KL}\left[q(\mathbf{z} \mid \mathbf{t}, \hat{\mathbf{z}}) \,\|\, q(\mathbf{z} \mid \hat{\mathbf{z}}) \right]} \\
&\leq \sigma L \sqrt{\mathbb{E}_{q(\hat{\mathbf{z}})} \mathbb{E}_{F(\mathbf{t})} \mathrm{KL}\left[q(\mathbf{z} \mid \mathbf{t}, \hat{\mathbf{z}}) \,\|\, q(\mathbf{z} \mid \hat{\mathbf{z}}) \right]} \quad \{\text{by Cauchy Schwarz}\} \\
&= \sigma L \sqrt{\mathbb{E}_{q(\hat{\mathbf{z}})} \mathbb{E}_{q(\mathbf{t} \mid \hat{\mathbf{z}})} \frac{F(\mathbf{t})}{q(\mathbf{t} \mid \hat{\mathbf{z}})} \mathrm{KL}\left[q(\mathbf{z} \mid \mathbf{t}, \hat{\mathbf{z}}) \,\|\, q(\mathbf{z} \mid \hat{\mathbf{z}}) \right]} \\
&\leq \sigma L \sqrt{W} \sqrt{\mathbb{E}_{q(\hat{\mathbf{z}})} \mathbb{E}_{q(\mathbf{t} \mid \hat{\mathbf{z}})} \mathrm{KL}\left[q(\mathbf{z} \mid \mathbf{t}, \hat{\mathbf{z}}) \,\|\, q(\mathbf{z} \mid \hat{\mathbf{z}}) \right]} \\
&= \sigma L \sqrt{W} \sqrt{\mathbf{I}(\mathbf{t}; \mathbf{z} \mid \hat{\mathbf{z}})}
\end{aligned}
$$

□

D.1.10 Estimation with the Two-stage least-squares method

We first describe the general version of two-stage least-squares method (2SLS). Let the outcome, treatment and IV be $\mathbf{y}, \mathbf{t}', \boldsymbol{\epsilon}$ respectively and the true data distribution be $p(\mathbf{t}', \mathbf{y}, \boldsymbol{\epsilon})$.

1. In the first-stage, 2SLS learns the distribution $q(\mathbf{t} \mid \boldsymbol{\epsilon})$. Given some class of distributions Q, the first-stage can be framed as a maximum-likelihood problem:

   $$q = \arg\max_{q' \in Q} \mathbb{E}_{p(\mathbf{t}', \boldsymbol{\epsilon})} \log q'(\mathbf{t}' \mid \boldsymbol{\epsilon})$$

   In our setup, $\mathbf{t}$ is the *synthetic treatment* sampled from the conditional distribution q estimated in the first stage.

2. In the second-stage, 2SLS learns the conditional distribution of the outcome $\mathbf{y}$ given the *synthetic treatment* $\mathbf{t}$ sampled from the conditional $q(\mathbf{t} \mid \boldsymbol{\epsilon})$ from the first stage. Given some class of distributions G, 2SLS's second-stage can be framed as a maximum-likelihood problem:

   $$g = \arg\max_{g' \in G} \mathbb{E}_{p(\mathbf{y}, \boldsymbol{\epsilon})} \mathbb{E}_{q(\mathbf{t} \mid \boldsymbol{\epsilon})} \log g'(\mathbf{y} \mid \mathbf{t}).$$

   The causal effect estimate is then computed as: $f^*(t) = \mathbb{E}_{g(\mathbf{y} \mid \mathbf{t}=t)}[\mathbf{y}]$.

Typically in settings with continuous $\mathbf{y}, \mathbf{t}$, both stages of 2SLS are framed and implemented as least-squares regressions instead of maximum-likelihood problems. See Kelejian [121] for an overview of classical vs. Bayesian two-stage least-squares methods.

In this section, we derive an alternate expression for 2SLS's causal effect estimate $f^*(t)$. Recall that $\mathbf{t}$ is the *synthetic treatment* sampled from the conditional distribution q estimated in the first stage. We assume that both stages of 2SLS are perfectly solved. Note that $\mathbf{t}$ is independently sampled conditioned on $\boldsymbol{\epsilon}$. This imposes the following conditional independencies:

$$\mathbf{y} \perp\!\!\!\perp \mathbf{t} \mid \boldsymbol{\epsilon}, \mathbf{t}' \quad \textit{and} \quad \mathbf{t}' \perp\!\!\!\perp \mathbf{t} \mid \boldsymbol{\epsilon}.$$

We marginalize out $\mathbf{t}', \boldsymbol{\epsilon}$ from the joint $q(\mathbf{y}, \mathbf{t}, \mathbf{t}', \boldsymbol{\epsilon})$ to get the dependence of $\mathbf{y}$ on $\mathbf{t}$:

$$\begin{aligned}
f^*(t) &= \mathbb{E}[\mathbf{y} = y \mid \mathbf{t} = t] \\
&= \int_{t',\epsilon} yq(\mathbf{y} = y, \mathbf{t}' = t', \boldsymbol{\epsilon} = \epsilon \mid \mathbf{t} = t) d\epsilon dy dt' \\
&= \int_{t',\epsilon} yp(\mathbf{y} = y \mid \mathbf{t}' = t', \boldsymbol{\epsilon} = \epsilon, \mathbf{t} = t) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{t} = t) p(\mathbf{t} = t' \mid \boldsymbol{\epsilon} = \epsilon, \mathbf{t} = t) d\epsilon dy dt' \\
&= \int_{t',\epsilon} yp(\mathbf{y} = y \mid \mathbf{t}' = t', \boldsymbol{\epsilon} = \epsilon) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{t} = t) p(\mathbf{t}' = t' \mid \boldsymbol{\epsilon} = \epsilon) d\epsilon dy dt' \\
&\quad \{\text{by } \mathbf{t}' \perp\!\!\!\perp \mathbf{t} \mid \boldsymbol{\epsilon},\ \mathbf{t} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{t}, \boldsymbol{\epsilon}\},
\end{aligned} \tag{D.8}$$

which yields

$$\begin{aligned}
f^*(t) &= \int yp(\mathbf{y} = y \mid \mathbf{t}' = t', \boldsymbol{\epsilon} = \epsilon) q(\boldsymbol{\epsilon} = \epsilon \mid \mathbf{t} = t) p(\mathbf{t}' = t' \mid \boldsymbol{\epsilon} = \epsilon) d\epsilon dy dt' \\
&= \mathbb{E}_{q(\boldsymbol{\epsilon} \mid \mathbf{t} = t)} \mathbb{E}_{p(\mathbf{t}' \mid \boldsymbol{\epsilon})} \mathbb{E}[\mathbf{y} \mid \mathbf{t}', \boldsymbol{\epsilon}].
\end{aligned} \tag{D.9}$$

This shows that the effect estimated by 2SLS can be rewritten as

$$f^*(t) = \mathbb{E}[\mathbf{y} \mid \mathbf{t} = t] = \mathbb{E}_{q(\boldsymbol{\epsilon} \mid \mathbf{t} = t)} \mathbb{E}_{p(\mathbf{t}' \mid \boldsymbol{\epsilon})} \mathbb{E}[\mathbf{y} \mid \mathbf{t}', \boldsymbol{\epsilon}]$$

With this, we show that 2SLS's estimation is biased when the outcome process might have multiplicative interactions between treatment and confounders. Consider this data generation:

$$\boldsymbol{\epsilon}, \mathbf{z} \sim \mathcal{N}(0, 1), \ \ \mathbf{t} = \boldsymbol{\epsilon} + \mathbf{z}, \ \ \mathbf{y} = \mathbf{t} + \mathbf{t}^2 \mathbf{z}.$$

Let $p(\mathbf{t} \mid \boldsymbol{\epsilon})$ be the learned conditional treatment distribution from a perfectly solved first-stage.

We use the reverse conditional $p(\boldsymbol{\epsilon} \mid \mathbf{t})$. 2SLS's causal effect estimate can be rewritten as $f(t) = \mathbb{E}_{q(\boldsymbol{\epsilon} \mid \mathbf{t}=t)}\mathbb{E}_{p(\mathbf{t}' \mid \boldsymbol{\epsilon})}\mathbb{E}[\mathbf{y} \mid \mathbf{t}', \boldsymbol{\epsilon}]$. The true causal effect is $f(t) = \mathbb{E}_{p(\mathbf{z})}[\mathbf{t} + \mathbf{t}^2\mathbf{z} \mid \mathrm{do}(\mathbf{t} = t)] = t$. Note that $\mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{t} = t] = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,1)}[t - \mathbf{z}] = t$. The 2SLS-estimate is $3t \neq t = f(t)$:

$$\begin{aligned} f^*(t) &= \mathbb{E}_{q(\boldsymbol{\epsilon} \mid \mathbf{t}=t)}\mathbb{E}_{p(\mathbf{t}' \mid \boldsymbol{\epsilon})}\mathbb{E}[\mathbf{y} \mid \mathbf{t}', \boldsymbol{\epsilon}] \\ &= \mathbb{E}_{q(\boldsymbol{\epsilon} \mid \mathbf{t}=t)}\mathbb{E}_{p(z)}\mathbb{E}[\mathbf{y} \mid \mathbf{t}' = \mathbf{z} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon}] \\ &= \mathbb{E}_{q(\boldsymbol{\epsilon} \mid \mathbf{t}=t)}\mathbb{E}_{p(z)}[\boldsymbol{\epsilon} + \mathbf{z} + (\boldsymbol{\epsilon} + \mathbf{z})^2\mathbf{z}] = 3t \end{aligned}$$

This shows 2SLS needs to assume properties of the true outcome and treatment processes.

D.1.11 The DeepIV objective

DeepIV [131] extends the two-stage least-squares method to use neural networks in both stages of treatment and outcome estimation. For simplicity, we ignore the covariates $\mathbf{x}$. The first stage of DeepIV estimates the conditional density of treatment given the IV. Assuming the first-stage of DeepIV is solved and we have an estimate $p_\theta(\mathbf{t} \mid \boldsymbol{\epsilon})$, the outcome stage of DeepIV solves the following to obtain an estimate $f_\phi(\mathbf{t})$ for the true causal effect $f(t) = \mathbb{E}[\mathbf{y} \mid \mathrm{do}(\mathbf{t} = t)]$:

$$\min_{\phi} \mathbb{E}_{\mathbf{y},\boldsymbol{\epsilon}}[\mathbf{y} - \mathbb{E}_{p_\theta(\mathbf{t} \mid \boldsymbol{\epsilon})}f_\phi(\mathbf{t})]^2. \tag{D.10}$$

This optimization eq. (D.10) has a subtle issue. We will show that there exist different functions that solve the optimization problem, thereby resulting in different treatment-effect estimates. Assume that the first stage was solved with $\mathbf{t} \sim p(\mathbf{t} \mid \boldsymbol{\epsilon})$. The trouble lies in the fact that eq. (D.10) averages the function $f_\phi(\mathbf{t})$ over the distribution $p(\mathbf{t} \mid \boldsymbol{\epsilon})$. If there exists a function $f' \neq 0$ such that $\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}f'(\mathbf{t}) = 0$, both f and $f + f'$ solve the optimization problem in Equation (D.10). As there is no way to separate f from functions like $f + f'$, we face a non-identifiability issue.

We show that multiplicative interactions between $\boldsymbol{\epsilon}, \mathbf{z}$ in the true treatment process is a sufficient condition for such functions f' to exist. Consider the following data generation with no confounding:

$$\boldsymbol{\epsilon}, \mathbf{z} \sim \mathcal{N}(0,1), \;\; \mathbf{t} = \mathbf{z}\boldsymbol{\epsilon}, \;\; \mathbf{y} = \mathbf{t}^2.$$

Here the true causal effect is $f(t) = t^2$. We will show that $\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})} f(\mathbf{t}) = \mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}(f(\mathbf{t}) + \mathbf{t})$, meaning that both $f(t)$ and $f(t) + t$ solve the optimization problem eq. (D.10). Notice that $\mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = 0$ and therefore

$$\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}(f(\mathbf{t}) + \mathbf{t}) = \mathbb{E}[(\mathbf{t}^2 + \mathbf{t}) \mid \boldsymbol{\epsilon}] = \mathbb{E}[\mathbf{t}^2 \mid \boldsymbol{\epsilon}] + \mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = \mathbb{E}[\mathbf{t}^2 \mid \boldsymbol{\epsilon}] = \mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}[f(\mathbf{t})].$$

For any constant a, the function $t^2 + at$ also solves the optimization problem in eq. (D.10). This means that multiple solutions to the DeepIV objective exist that are not the true causal effect.

One potential reason that DeepIV may not run into this non-identifiability issue is that an upper bound of the original proposed objective is solved instead. To compute gradients for the original optimization, two independent expectations are needed, which is not sample-efficient; this is called the double-sample problem. So, [131] optimize an upper bound (via Jensen's):

$$\mathbb{E}_{F(\mathbf{y},\boldsymbol{\epsilon})}[\mathbf{y} - \mathbb{E}_{p_\theta(\mathbf{t} \mid \boldsymbol{\epsilon})} f_\phi(\mathbf{t})]^2 \leq \mathbb{E}_{F(\mathbf{y},\boldsymbol{\epsilon})} \mathbb{E}_{p_\theta(\mathbf{t} \mid \boldsymbol{\epsilon})}[\mathbf{y} - f_\phi(\mathbf{t})]^2. \tag{D.11}$$

The RHS above is a log-likelihood problem with a Gaussian likelihood. A general form of this is $\mathbb{E}_{F(\mathbf{y},\boldsymbol{\epsilon})} \mathbb{E}_{p_\theta(\mathbf{t} \mid \boldsymbol{\epsilon})} \log p_\phi(\mathbf{y} \mid \mathbf{t})$; where p_ϕ is supposed to model the distribution of the outcome under $\text{do}(\mathbf{t})$. Finally, as DeepIV is based on 2SLS, DeepIV assumes an additive outcome process to avoid the issues in the previous section.

DeepIV under multiplicative treatment processes We show here that the upper bound that DeepIV minimizes can also produce biased effect estimates when the true treatment process is multiplicative. The upper bound that DeepIV optimizes is:

$$\arg\min_{f^*} \mathbb{E}_{F(\mathbf{y},\boldsymbol{\epsilon})}\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}[\mathbf{y} - f^*(\mathbf{t})]^2 = \arg\min_{f^*} \mathbb{E}_{F(\boldsymbol{\epsilon})}\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}\mathbb{E}_{F(\mathbf{y} \mid \boldsymbol{\epsilon})}[\mathbf{y} - f^*(\mathbf{t})]^2$$

Note that we use $F(\mathbf{y} \mid \boldsymbol{\epsilon})$ and not $F(\mathbf{y} \mid \mathbf{t}, \boldsymbol{\epsilon})$ because here $\mathbf{t}$ refers to the synthetic treatment sampled from the conditional distribution $p(\mathbf{t} \mid \boldsymbol{\epsilon})$ learned in the first stage of DeepIV, which means $\mathbf{y} \perp\!\!\!\perp \mathbf{t} \mid \boldsymbol{\epsilon}$. We do a bias-variance decomposition of the expectation and refer to terms that do not depend on h as constants C with respect to the optimization.

$$\begin{aligned}\mathbb{E}_{F(\boldsymbol{\epsilon})}\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}\mathbb{E}_{F(\mathbf{y} \mid \boldsymbol{\epsilon})}[\mathbf{y} - f^*(\mathbf{t})]^2 &= \mathbb{E}_{F(\boldsymbol{\epsilon})}\mathbb{E}_{p(\mathbf{t} \mid \boldsymbol{\epsilon})}\mathbb{E}_{F(\mathbf{y} \mid \boldsymbol{\epsilon})}[\mathbb{E}[\mathbf{y} \mid \boldsymbol{\epsilon}] - f^*(\mathbf{t})]^2 + \mathbb{E}_{F(\boldsymbol{\epsilon})}[\sigma^2(\mathbf{y} \mid \boldsymbol{\epsilon})] \\ &= \mathbb{E}_{p(\mathbf{t})}\mathbb{E}_{p(\boldsymbol{\epsilon} \mid \mathbf{t})}\mathbb{E}_{F(\mathbf{y} \mid \boldsymbol{\epsilon})}[\mathbb{E}[\mathbf{y} \mid \boldsymbol{\epsilon}] - f^*(\mathbf{t})]^2 + C\end{aligned} \tag{D.12}$$

Now consider the generation process $\boldsymbol{\epsilon}, \mathbf{z} \sim \mathcal{N}(0, 1)$ with the true treatment and outcome generated as $\mathbf{t} = \boldsymbol{\epsilon}\mathbf{z}$ and $\mathbf{y} = \mathbf{t} + \mathbf{z}$. Note that $E[\mathbf{y} \mid \boldsymbol{\epsilon} = a] = E_{\mathbf{z}}[\mathbf{z} + a\mathbf{z}] = 0$. Therefore the optimization reduces to the following:

$$\arg\min_{f^*} \mathbb{E}_{p(\mathbf{t})}\mathbb{E}_{p(\boldsymbol{\epsilon} \mid \mathbf{t})}\mathbb{E}_{F(\mathbf{y} \mid \boldsymbol{\epsilon})}[0 - f^*(\mathbf{t})]^2 = 0 \neq f(t) = t$$

Thus, DeepIV's relaxed optimization problem also needs assumptions on the true treatment process.

D.1.12 Information preserving maps and additional utility constraints

A bijective map is one that maps each element in its domain to a unique element in its range. No information can be lost in this process, resulting in bijective transformations being called information-preserving maps. Information-preserving maps preserve computations that only involve conditioning and expectations; meaning that the causal effect estimate $\mathbb{E}_{\hat{\mathbf{z}}}\mathbb{E}[\mathbf{y} \mid \mathbf{t}, \hat{\mathbf{z}}]$ is preserved. Therefore we can impose additional distributional utility constraints satisfied by bijective transformations of the general control function $\hat{\mathbf{z}}$, without losing the properties of ignorability.

Coupled with flexible over-parametrized modelling, information-preserving maps give us the ability to enforce utility constraints on the latent space of $\hat{\mathbf{z}}$. If there is an outcome-model that works well with data drawn from a normal distribution, one can add an additional term to VDE's objective that is the KL divergence between the distribution of $\hat{\mathbf{z}}$ and a normal distribution. If we wanted information about continuity in $\mathbf{t}$ to be preserved in $\hat{\mathbf{z}}$, we could enforce linear interpolation. Similarly, we could force an constructed $\hat{\mathbf{z}}$ to have a monotonic relation with $\mathbf{t}$. One could enforce multiple constraints from a combination of distances, divergences, ordering and modality constraints. When used correctly, these constraints trade optimization complexity between outcome-stage and VDE.

D.2 Experimental Details

In this section, we expand on the details of experiments presented in section 5.4. In all experiments, the hidden layers in both encoder and decoder networks have 100 units and use ReLU activations. The outcome model is also a 2-hidden-layer neural network with ReLU activations unless specified otherwise. For the simulated data, the hidden layers in the outcome model have 50 hidden units. We optimize VDE and outcome-stage for 100 epochs with Adam; starting with a learning rate of 10^{-2} and halving it every 10 epochs if the training error goes up.

D.2.1 Selecting λ

We discuss here why good λ (equivalently κ) can be selected based on the resulting expected outcome likelihood, i.e. the outcome modelling objective, on a heldout validation set.

As VDE's control function is constructed as a function of $(\mathbf{t}, \boldsymbol{\epsilon})$, i.e. $\hat{\mathbf{z}} = e(\mathbf{t}, \boldsymbol{\epsilon})$, it holds that $\mathbf{y} \perp\!\!\!\perp \hat{\mathbf{z}} \mid \boldsymbol{\epsilon}, \mathbf{t}$. So, predicting $\mathbf{y}$ from $(\hat{\mathbf{z}}, \mathbf{t})$, as in GCFN, cannot be better than predicting $\mathbf{y}$ from $(\mathbf{t}, \boldsymbol{\epsilon})$:

$$\mathbf{H}(\mathbf{y} \mid \mathbf{t}, \hat{\mathbf{z}}) \geq \mathbf{H}(\mathbf{y} \mid \mathbf{t}, \boldsymbol{\epsilon}, \hat{\mathbf{z}}) = \mathbf{H}(\mathbf{y} \mid \mathbf{t}, \boldsymbol{\epsilon}).$$

The slack in the inequality is $\mathbf{H}(\mathbf{y} \mid \mathbf{t}, \hat{\mathbf{z}}) - \mathbf{H}(\mathbf{y} \mid \mathbf{t}, \boldsymbol{\epsilon}, \hat{\mathbf{z}}) = \mathbf{I}(\mathrm{y}, \boldsymbol{\epsilon} \mid \mathbf{t}, \hat{\mathbf{z}})$ and equality holds when $\mathbf{y} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \mathbf{t}, \hat{\mathbf{z}}$. This independence holds in general only if both $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}$ and perfect reconstruction hold; see appendix D.2.1.1. Thus, in general, the expected outcome likelihood achieves maximum only when both perfect reconstruction and conditional independence are satisfied.

In practice, instead of the unconstrained VDE, we optimize the lower-bound objective in eq. (5.6) on a finite dataset. Due to local minima or finite-sample error, this lower-bound optimized with a κ that is too large may give a $\hat{\mathbf{z}}$ that retains little information about $\mathbf{z}$ so as keep the $\mathbf{KL}$ small. Similarly, when κ is too small, $\hat{\mathbf{z}}$ may memorize $\mathbf{t}$ to keep the reconstruction error small without paying much in the $\kappa \times \mathbf{KL}$ term. In either case, the resulting $\hat{\mathbf{z}}$ fails to satisfy one of either perfect reconstruction or conditional independence, meaning that $\mathbf{y} \not\!\perp\!\!\!\perp \boldsymbol{\epsilon} \mid \mathbf{t}, \hat{\mathbf{z}}$ in general. Then, as discussed above, the outcome model cannot achieve the maximum possible expected outcome likelihood. This insight suggests the following procedure to select good κ based on validation outcome likelihood: [§]:

1. Solve VDE for a collection of κ and obtain the control function $\hat{\mathbf{z}}_\kappa$ for each.
2. Regress $\mathbf{y}$ on $\mathbf{t}, \hat{\mathbf{z}}_\kappa$ and evaluate expected outcome likelihood on a heldout validation set.

[§] At first glance, one failure case seems to be when $\hat{\mathbf{z}}$ memorizes $\boldsymbol{\epsilon}$ only, leading to $\mathbf{y} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \mathbf{t}, \hat{z}$. However, such a $\hat{\mathbf{z}}$ does not help reconstruct $\mathbf{t}$ along with $\boldsymbol{\epsilon}$ while resulting in a large KL $[q(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}) \parallel q(\hat{\mathbf{z}})]$. This leads to a very sub-optimal objective value in VDE. As we maximize to solve VDE, such failure cases do not occur.

(This heldout set should be different from the one used to tune all other hyperparameters)

3. Select the κ that led to the largest validation outcome likelihood; use the corresponding $\hat{\mathbf{z}}_\kappa$ in GCFN's second stage to estimate effects (retrain or use the model from step 2).

D.2.1.1 Conditional Independence of outcome and instrument given $\hat{\mathbf{z}}, \mathbf{t}$

By definition, the potential outcome $\mathbf{y}_\mathbf{t}$ depends only on $\mathbf{z}$ and for any observed $(\mathbf{t}, \mathbf{y})$, and by consistency, $\mathbf{y} = \mathbf{y}_\mathbf{t}$. Therefore $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t} \implies \mathbf{y}_\mathbf{t} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t} \iff \mathbf{y} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}$. Under the joint $q(\hat{\mathbf{z}}, \mathbf{t}, \boldsymbol{\epsilon}, \mathbf{z}) = q(\hat{\mathbf{z}} \mid \mathbf{t}, \boldsymbol{\epsilon}) F(\mathbf{t}, \boldsymbol{\epsilon}, \mathbf{z})$, it follows that $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}$ when the reconstruction property and the conditional independence $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}$ hold:

$$
\begin{aligned}
q(\mathbf{z}, \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}) &= q(\mathbf{z} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}, \mathbf{t}) q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}) \\
&= q(\mathbf{z} \mid \boldsymbol{\epsilon}, \hat{\mathbf{z}}) q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}) \quad \{\text{by reconstruction } \mathbf{t} = d(\hat{\mathbf{z}}, \boldsymbol{\epsilon})\} \\
&= q(\mathbf{z} \mid \hat{\mathbf{z}}) q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}) \quad \{\text{by joint independence } \mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}\} \\
&= q(\mathbf{z} \mid \hat{\mathbf{z}}, \mathbf{t}) q(\boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}) \quad \{\text{by joint independence and reconstruction } \mathbf{z} \perp\!\!\!\perp \mathbf{t} \mid \hat{\mathbf{z}}\},
\end{aligned}
\tag{D.13}
$$

where $\mathbf{z} \perp\!\!\!\perp \mathbf{t} \mid \hat{\mathbf{z}}$ is shown in the proof of theorem 5. If $\mathbf{y}_\mathbf{t}$ is an invertible function of $\mathbf{z}$, $\mathbf{y} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t} \implies \mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}$. Thus, in general, $\mathbf{z} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}$ is a necessary condition for $\mathbf{y} \perp\!\!\!\perp \boldsymbol{\epsilon} \mid \hat{\mathbf{z}}, \mathbf{t}$.

D.2.2 Simulations with Specific Decoder Structure

We used the python package *statsmodels* for 2SLS and our own implementation of CFN. We used the DeepIV package developed by Hartford et al. [131].

Multiplicative treatment + Additive outcome. We use the 2SLS function from statsmodels [259] which uses a linear model $\mathbf{t} = \beta \boldsymbol{\epsilon} + \boldsymbol{\eta}_\mathbf{t}$ that will correctly predict that $\mathbb{E}[\mathbf{t} \mid \boldsymbol{\epsilon}] = 0$. We optimized the treatment and the response models in DeepIV [131] for a 100 epochs each.

D.2.3 GCFN on high-dimensional covariates

Here, we give further details about section 5.4.3. We give Hartford et al. [131]'s simulation with our notation:

$$\mathbf{z}, \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), \quad \mathbf{t} = 25 + (\boldsymbol{\epsilon} + 3)\psi_s + \nu, \quad \mathbf{y} = \mathcal{N}\big(100 + (10+\mathbf{t})\ell(\mathbf{x})\psi_s - 2\mathbf{t} + 0.5\mathbf{z}, 0.75\big),$$

where ψ_s is a non-linear function of time s, and $\ell(\mathbf{x})$ is the label of the MNIST image. We optimized both VDE and outcome stage with Adam with batch size 500 for 200 epochs beginning at 10^{-2} and halving the learning rate when the average loss over 5 epochs increases. We use the outcome model architecture from DeepIV [131] where convolutional layers construct a representation which is concatenated with $\mathbf{t}$ and s, before being fed to the fully-connected layers. GCFN's outcome model differs only in that the fully-connected layers take as input the control function $\hat{\mathbf{z}}$, time s and treatment $\mathbf{t}$. The best outcome model was chosen based on validation outcome MSE.

D.2.4 GCFN on high-dimensional IV

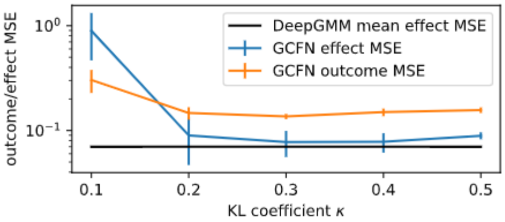


Figure D.2: GCFN performs on par with DeepGMM on high-dimensional IV experiment specified in DeepGMM [132].

Here, we give further details about section 5.4.4. The encoder and additive decoder in VDE are 2-layer networks like in the section 5.4.1. In this experiment we use a 3 layer outcome model with 50 units in each layer. We used 10,000 samples as in DeepGMM and optimized both VDE and outcome stage with Adam with a batch size 1000 for 100 epochs beginning at a learning rate of 10^{-2} and halving it when the average loss over 5 epochs increases. We plot outcome and effect MSE for GCFN for 5 different

$\kappa \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ in fig. D.2. Note that low outcome MSE corresponds to low effect MSE. The plot shows mean and standard deviation of effect MSE of the causal effect for 5 different α's and 10 random seeds. GCFN performs on par with DeepGMM [132].

D.2.5 Additional experiments

The following experiment is done with a structurally unrestricted decoder even though the true treatment process is additive. We compare against CFN to demonstrate that GCFN does not require structural restrictions on the outcome process. Let $\mathcal{N}$ be the normal distribution and α be a parameter to control the confounding strength. We generate

$$\mathbf{z}, \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), \quad \mathbf{t} = (\mathbf{z} + \boldsymbol{\epsilon})/\sqrt{2}, \quad \mathbf{y} \sim \mathcal{N}(\mathbf{t}^2 + \alpha \mathbf{z}^2, 0.1). \tag{D.14}$$

The larger the absolute values of α, the more the confounding. In economics terminology, the treatment noise and the outcome noise are $\boldsymbol{\eta}_t = \mathbf{z}$ and $\boldsymbol{\eta}_y = \alpha \mathbf{z}^2 + \textit{noise}$ respectively. The generation process in eq. (D.14) violates assumption A4 in Guo and Small [134] for CFN: $\mathbb{E}[\boldsymbol{\eta}_y | \boldsymbol{\eta}_t] \propto \boldsymbol{\eta}_t$. GCFN does not require this assumption. We use 5000 samples and a batch size of 500. We discretize the treatment to have 50 categories. Of the 50, 48 categories correspond to equally sized bins in $[-3.5, 3.5]$, with the remaining 2 correspond to values less than -3.5 and greater than 3.5 respectively. We compare against CFN with both stages correctly specified as functions of $\mathbf{t}$ and $\mathbf{z}$.

We find, as expected, that GCFN out-performs CFN. Over 5 runs, for $\alpha = 1$, we obtain an RMSE of $\mathbf{0.3 \pm 0.1}$ while the CFN only manages to obtain an RMSE of $\mathbf{1.5 \pm 0.1}$ despite having the correctly specified model for $\mathbf{t}^2$. For other $\alpha \in \{-2, -1, 2\}$, GCFN was similarly better.

E | Appendices for chapter 6

E.1 Theoretical details

E.1.1 A note about the assumptions

Note about the assumptions In theorem 6.1, assumption 1 consists of three parts that can all be validated on observed data: 1) that the gradient flow converges, 2) that the confounder value of the surrogate matches the confounder value whose effect is of interest, and 3) that the surrogate intervention lies in the support of the pre-outcome variables. Assumption 2 is required for expectations and their gradients to exist and be finite. In theorem 6.2, assumption 1 requires a consistent estimator of $\mathbb{E}[\mathbf{y} \mid \mathbf{t}]$, which can be provided with regression. Assumption 3 lists regularity conditions which help control how the surrogate estimation error propagates to the effect error.

E.1.2 Proof of Theorem 6.1

We restate the theorem for completeness:

Theorem E.1. *Assume* C-REDUNDANCY *holds. Assuming the following:*

1. *Let* $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ *be the limiting solution to the gradient flow equation* $\frac{d\tilde{\mathtt{t}}(s)}{ds} = -\nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}(s)) - h(\mathtt{t}_2^*))^2$, *initialized at* $\tilde{\mathtt{t}}(0) = \mathtt{t}^*$*; i.e.* $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \lim_{s\to\infty} \tilde{\mathtt{t}}(s)$.

Further, let $h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))) = h(\mathtt{t}_2^*)$ *and* $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)) \in \mathit{supp}(\mathbf{t})$.

2. $f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}), \eta)$ *and* $h(\tilde{\mathtt{t}})$ *as functions of* $\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}})$ *are continuous and differentiable and the derivatives exist for all* $\tilde{\mathtt{t}}, \eta$. *Let* $\nabla_{\tilde{\mathtt{t}}} f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}), \eta)$ *exist and be bounded and integrable w.r.t. the probability measure corresponding to* $p(\boldsymbol{\eta})$*, for all values of* $\tilde{\mathtt{t}}$ *and* $h(\tilde{\mathtt{t}})$.

Then the conditional effect (and therefore the average effect) is identified:

$$\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \phi\left(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))\right) = \mathbb{E}\left[\mathbf{y} \mid \mathbf{t} = \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))\right] \tag{E.1}$$

Proof. Recall definition of conditional effect $\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2)) = \mathbb{E}_{\boldsymbol{\eta}} f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2), \boldsymbol{\eta})$. Recall $\nabla_{\tilde{\mathtt{t}}}$ is the gradient with respect to the first argument of f, that is $\tilde{\mathtt{t}}$. First, by assumption 2, $\mathbb{E}$ and ∇ commute, under the dominated convergence theorem. Then, by C-REDUNDANCY

$$\nabla_{\tilde{\mathtt{t}}} \phi(\tilde{\mathtt{t}}, h(\mathtt{t}^*))^T \nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}}) = \nabla_{\tilde{\mathtt{t}}} \mathbb{E}_{\boldsymbol{\eta}} f(\tilde{\mathtt{t}}, h(\mathtt{t}^*), \boldsymbol{\eta})^T \nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}}) = \mathbb{E}_{\boldsymbol{\eta}}[\nabla_{\tilde{\mathtt{t}}} f(\tilde{\mathtt{t}}, h(\mathtt{t}^*), \boldsymbol{\eta})^T \nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}})] = 0.$$

Now consider the gradient flow equation ${d\tilde{\mathtt{t}}(s)}/{ds} = -\nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}) - h(\mathtt{t}_2^*))^2$. We refer to the gradient evaluated at $\tilde{\mathtt{t}}$ as $\Delta\tilde{\mathtt{t}} = -\nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}) - h(\mathtt{t}_2^*))^2 = -2(h(\tilde{\mathtt{t}}) - h(\mathtt{t}_2^*))\nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}})$. We will express $\phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}_2^*))$ as defined by the starting point $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$ and the gradient flow equation.

Let the solution path to the gradient flow equation be C with $\mathtt{t}^*, \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ being the starting and ending points respectively. By the Gradient Theorem [260], we have that $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$ and $\phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}_2^*))$ are related via the line integral over C:

$$\int_C \nabla_{\tilde{\mathtt{t}}} \phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*)) \cdot d\tilde{\mathtt{t}} = \phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}_2^*)) - \phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*))$$

Let $\tilde{\mathtt{t}}(s)$ be a parametrization of solution path C by the scalar time $s \in [0, \infty)$. Now, to obtain the value of $\phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*))$, we will compute the line integral over the vector field defined by

$\nabla_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*))$, which exists by assumption 2 in theorem 6.1, evaluated along the path C defined by $\Delta\tilde{\mathtt{t}}(s)$:

$$\begin{aligned}
\phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}_2^*)) &= \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) + \int_C \nabla_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}, h(\mathtt{t}_2^*)) \cdot d\tilde{\mathtt{t}} \\
&= \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) + \int_0^\infty \nabla_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}(s), h(\mathtt{t}_2^*))^T \frac{d\tilde{\mathtt{t}}(s)}{ds}\, ds \\
&= \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) + \int_0^\infty \nabla_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}(s), h(\mathtt{t}_2^*))^T \Delta\tilde{\mathtt{t}}(s)\, ds \\
&= \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) \\
&\qquad + \int_0^\infty -2((h(\tilde{\mathtt{t}}(s)) - h(\mathtt{t}_2^*)))\, \nabla_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}(s), h(\mathtt{t}_2^*))^T \nabla_{\tilde{\mathtt{t}}} h(\tilde{\mathtt{t}}(s))\, ds \\
&= \phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) + 0 \qquad \{\text{by C-REDUNDANCY}\}
\end{aligned} \tag{E.2}$$

Finally, by assumption 1 in theorem 6.1, $h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))) = h(\mathtt{t}_2^*)$, and so

$$\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}_2^*)) = \phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))) \tag{E.3}$$

For clarity, the same equation, but using $\mathtt{t}'$ and suppressing dependence on $\mathtt{t}^*, h(\mathtt{t}_2^*)$):

$$\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = \phi(\mathtt{t}', h(\mathtt{t}_2^*)) = \phi(\mathtt{t}', h(\mathtt{t}')) \tag{E.4}$$

Under the causal model for EFC, the outcome $\mathbf{y} = f(\mathbf{t}, h(\mathbf{t}), \boldsymbol{\eta})$. Then, $\forall \tilde{\mathtt{t}} \in \operatorname{supp}(p(\mathbf{t}))$,

$$\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}] = \mathbb{E}_{\boldsymbol{\eta}}[f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}), \boldsymbol{\eta})] = \phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}})). \tag{E.5}$$

Using that $\mathtt{t}'(\mathtt{t}^*, \mathtt{t}_2^*) \in \text{supp}(p(\mathbf{t}))$ and eqs. (E.4) and (E.5), the conditional effect is identified

$$\begin{aligned}\phi(\mathtt{t}^*, h(\mathtt{t}_2^*)) &= \phi(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)), h(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))) \\ &= \mathbb{E}[\mathbf{y} \mid \mathbf{t} = \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))]\end{aligned} \tag{E.6}$$

Thus, the conditional effect and the average effect are identified as $\mathbb{E}[\mathbf{y} \mid \mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))]$ and $\tau(\mathtt{t}^*) = \mathbb{E}_{h(\mathbf{t})}\mathbb{E}[\mathbf{y} \mid \mathtt{t}'(\mathtt{t}^*, h(\mathbf{t}))]$ respectively. □

Note about convergence of gradient flow Any ODE's solution, if it exists and converges, converges to an ω-limit set [261]. An ω-limit set is nonempty when the solution path lies entirely in a closed and bounded set and can consist of limit cycles, equilibrium points, or neither [261, 262]. A gradient flow equation $d\tilde{\mathtt{t}}(s)/ds = -\nabla h(\tilde{\mathtt{t}})$ (also called a gradient system) has the special property that its ω-limit set only consists of critical points of $h(\tilde{\mathtt{t}})$; critical points of $h(\tilde{\mathtt{t}})$ are also equilibrium points of the gradient flow equation [262]. Further, if $\nabla h(\tilde{\mathtt{t}})$ exists and is bounded and $h(\tilde{\mathtt{t}})$ has bounded sublevel sets ($\{\tilde{\mathtt{t}} : h(\tilde{\mathtt{t}}) \leq c\}$), then the solution to the gradient flow equation will entirely lie within a bounded set. This is because along the solution path, $h(\tilde{\mathtt{t}}(s))$ always decreases meaning that the solution will remain in any sublevel set it started in. Thus, if $h(\tilde{\mathtt{t}})$ has bounded sublevel sets, the solution of the gradient flow equation will converge only to critical points of $h(\tilde{\mathtt{t}})$.

E.1.3 Estimation error in lode

Theorem E.2. *Consider the conditional effect $\phi(\mathtt{t}^*, h(\mathtt{t}_2^*))$. Let $\hat{\mathtt{t}}(\mathtt{t}^*, h(\mathtt{t}_2^*))$ be the estimate of the surrogate intervention computed by lode, computed via Euler integration of the gradient flow $\frac{d\tilde{\mathtt{t}}(s)}{ds} = -\nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}(s)) - h(\mathtt{t}_2^*))^2$, initialized at $\tilde{\mathtt{t}}(0) = \mathtt{t}^*$. Assume the true surrogate $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ exists and is the limiting solution to the gradient flow equation.*

1. *Let the finite sample estimator of $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}]$ be $\hat{f}(\tilde{\mathtt{t}})$. Let the error for all $\tilde{\mathtt{t}}$ be bounded,*

$|\hat{f}(\tilde{\mathtt{t}}) - \mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}]| \leq c(N)$, *where* N *is the sample size and* $\lim_{N\to\infty} c(N) = 0$.

2. *Assume* K *Euler integrator steps were taken to find the surrogate estimate* $\hat{\mathtt{t}}(\mathtt{t}^*, h(\mathtt{t}_2^*))$, *each of size* ℓ. *Let the maximum confounder mismatch be* $\max_{i\leq K}(h(\tilde{\mathtt{t}}_i) - h(\mathtt{t}_2^*))^2 = M$.

3. *Let* $L_{z,\tilde{\mathtt{t}}}$ *be the Lipschitz-constant of* $\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2))$ *as a function of* $h(\tilde{\mathtt{t}}_2)$, *for fixed* $\tilde{\mathtt{t}}$.

   Let L_e *be the Lipschitz-constant of* $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}] = \phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}))$ *as a function of* $\tilde{\mathtt{t}}$.

   Assume h *has a gradient with bounded norm,* $\|\nabla h(\tilde{\mathtt{t}})\|_2 < L_h$.

   Assume f*'s Hessian has bounded eigenvalues:* $\forall \tilde{\mathtt{t}}, \tilde{\mathtt{t}}_2,\ \|\nabla^2_{\tilde{\mathtt{t}}}\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2))\|_2 \leq \sigma_{\mathrm{H}\phi}$.

The conditional effect estimate error, $\xi(\mathtt{t}^*, h(\mathtt{t}_2^*)) = |\hat{f}(\hat{\mathtt{t}}) - \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))|$, *is upper bounded by:*

$$c(N) + \min\left(L_e\|\mathtt{t}' - \hat{\mathtt{t}}\|_2,\ 2K\ell^2\left(O(\ell) + M\sigma_{\mathrm{H}\phi}L_h^2\right) + L_{z,\hat{\mathtt{t}}}\|h(\hat{\mathtt{t}}) - h(\mathtt{t}_2^*)\|_2\right) \tag{E.7}$$

Proof. (of Theorem 6.2) Recall the definition of conditional effect : $\phi(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2)) = \mathbb{E}_{\boldsymbol{\eta}} f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2), \boldsymbol{\eta})$.

LODE's estimate of the conditional effect is $\hat{f}(\hat{\mathtt{t}}(\mathtt{t}^*, h(\mathtt{t}_2^*)))$. We will suppress notation for dependence on $\mathtt{t}^*, h(\mathtt{t}_2^*)$, and use $\mathtt{t}'$ and $\hat{\mathtt{t}}$ to refer to the true surrogate intervention and the estimated surrogate interventions respectively. Note $\hat{f}$ is the estimate of the conditional expectation $\mathbb{E}[\mathbf{y} \mid \mathbf{t} = \tilde{\mathtt{t}}]$, learned from N samples. We first bound the error by splitting into two parts and bounding each separately:

$$\begin{aligned}
|\xi(\mathtt{t}^*, h(\mathtt{t}_2^*))| &= |\hat{f}(\hat{\mathtt{t}}) - \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))| \\
&\leq |\hat{f}(\hat{\mathtt{t}}) - \phi(\hat{\mathtt{t}}, h(\hat{\mathtt{t}}))| + |\phi(\hat{\mathtt{t}}, h(\hat{\mathtt{t}})) - \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))| \\
&\leq c(N) + |\phi(\hat{\mathtt{t}}, h(\hat{\mathtt{t}})) - \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))| \\
&\leq |\phi(\hat{\mathtt{t}}, h(\hat{\mathtt{t}})) - \phi(\hat{\mathtt{t}}, h(\mathtt{t}_2^*))| + |\phi(\hat{\mathtt{t}}, h(\mathtt{t}_2^*)) - \phi(\mathtt{t}^*, h(\mathtt{t}_2^*))| + c(N)
\end{aligned}$$

The first term is bounded via the Lipschitz-ness of ϕ as a function of $h(\tilde{\mathtt{t}})$ with fixed first argument

$\tilde{\mathrm{t}} = \hat{\mathrm{t}}$.

$$|\phi(\hat{\mathrm{t}}, h(\hat{\mathrm{t}})) - \phi(\hat{\mathrm{t}}, h(\mathrm{t}_2^*))| \leq L_{z,\hat{\mathrm{t}}} |h(\hat{\mathrm{t}}) - h(\mathrm{t}_2^*)|$$

We now bound the remaining term. Recall that LODE's computation of the surrogate intervention involved K gradient steps, each of size ℓ. We work with a constant step-size but the analysis can be generalized to a non-uniform step size. Indexing steps with i, let $d_i = h(\tilde{\mathrm{t}}_i) - h(\mathrm{t}_2^*)$ be the confounder mismatch error at the ith iterate. Then note that $\hat{\mathrm{t}} = \mathrm{t}^* - \ell \sum_{i=0}^{K-1} 2 d_i \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)$. We can use this to bound the error $\phi(\hat{\mathrm{t}}, h(\mathrm{t}_2^*)) - \phi(\mathrm{t}^*, h(\mathrm{t}_2^*))$. With $\tilde{\mathrm{t}}_K = \hat{\mathrm{t}}$ and $\tilde{\mathrm{t}}_0 = \mathrm{t}^*$, we proceed by expressing the error as a telescoping sum and using the Taylor expansion for $\phi(\tilde{\mathrm{t}}, h(\mathrm{t}_2^*))$ in terms of the the first argument $\tilde{\mathrm{t}}$.

$$\phi(\hat{\mathrm{t}}, h(\mathrm{t}_2^*)) - \phi(\mathrm{t}^*, h(\mathrm{t}_2^*)) = \sum_{i=0}^{K-1} \phi(\tilde{\mathrm{t}}_{i+1}, h(\mathrm{t}_2^*)) - \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) \tag{E.8}$$

$$= \sum_{i=0}^{K-1} \nabla_{\tilde{\mathrm{t}}} \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*))^\top (\tilde{\mathrm{t}}_{i+1} - \tilde{\mathrm{t}}_i) \tag{E.9}$$

$$+ \frac{1}{2} (\tilde{\mathrm{t}}_{i+1} - \tilde{\mathrm{t}}_i)^\top \nabla_{\tilde{\mathrm{t}}}^2 \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) (\tilde{\mathrm{t}}_{i+1} - \tilde{\mathrm{t}}_i) + \mathcal{O}(\|\tilde{\mathrm{t}}_{i+1} - \tilde{\mathrm{t}}_i\|_2^3) \tag{E.10}$$

$$= \sum_{i=0}^{K-1} 2\ell d_i \nabla_{\tilde{\mathrm{t}}} \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*))^\top \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i) + 2(\ell d_i)^2 \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)^\top \nabla_{\tilde{\mathrm{t}}}^2 \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i) + \mathcal{O}(\ell^3) \tag{E.11}$$

$$= \sum_{i=0}^{K-1} 0 + 2(\ell d_i)^2 \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)^\top \nabla_{\tilde{\mathrm{t}}}^2 \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i) + \mathcal{O}(\ell^3) \tag{E.12}$$

$$= \mathcal{O}(K\ell^3) + \sum_{i=0}^{K-1} 2(\ell d_i)^2 \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)^\top \nabla_{\tilde{\mathrm{t}}}^2 \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i) \tag{E.13}$$

$$\leq \mathcal{O}(K\ell^3) + \sum_{i=0}^{K-1} 2(\ell(h(\tilde{\mathrm{t}}_i) - h(\mathrm{t}_2^*)))^2 \left| \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)^\top \nabla_{\tilde{\mathrm{t}}}^2 \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i) \right| \tag{E.14}$$

$$\leq \mathcal{O}(K\ell^3) + \sum_{i=0}^{K-1} 2\ell^2 M \left| \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)^\top \nabla_{\tilde{\mathrm{t}}}^2 \phi(\tilde{\mathrm{t}}_i, h(\mathrm{t}_2^*)) \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i) \right| \tag{E.15}$$

$$\leq \mathcal{O}(K\ell^3) + \sum_{i=0}^{K-1} 2\ell^2 M \sigma_{\mathrm{H}\phi} \|\nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}_i)\|_2^2 \tag{E.16}$$

$$\leq \mathcal{O}(K\ell^3) + \sum_{i=0}^{K-1} 2\ell^2 M \sigma_{\mathrm{H}\phi} L_h^2 \tag{E.17}$$

$$= 2K\ell^2 \left(\mathcal{O}(\ell) + M\sigma_{\mathrm{H}\phi} L_h^2\right), \tag{E.18}$$

where the inequalities follow by the maximum value of $(h(\tilde{\mathrm{t}}_i) - h(\mathrm{t}_2^*))^2$, bounded eigenvalues of the Hessian of ϕ and the Lipschitz-ness of $h(\tilde{\mathrm{t}})$.

Another way we bound the error is via the Lipschitz constant of the conditional expectation as a function of $\tilde{\mathrm{t}}$. Recall this is L_e. An alternate bound on the error is as follows:

$$|\phi(\hat{\mathrm{t}}, h(\hat{\mathrm{t}})) - \phi(\mathrm{t}^*, h(\mathrm{t}_2^*))| = |\phi(\hat{\mathrm{t}}, h(\hat{\mathrm{t}})) - \phi(\mathrm{t}', h(\mathrm{t}'))| \leq L_e \|\mathrm{t}' - \hat{\mathrm{t}}\|_2$$

The bound follows:

$$|\xi(\tilde{\mathrm{t}}, h(\mathrm{t}_2^*))| \leq c(N) + \min\left(L_e\|\mathrm{t}' - \hat{\mathrm{t}}\|_2, \quad 2K\ell^2\left(\mathcal{O}(\ell) + M\sigma_{\mathrm{H}\phi}L_h^2\right) + L_{z,\hat{\mathrm{t}}}\|h(\hat{\mathrm{t}}) - h(\mathrm{t}_2^*)\|_2\right)$$

□

E.1.3.1 A note on linear confounder functions and lode

In the proof above, the error in Euler integration accumulates due to terms like this one:

$$\nabla_{\tilde{\mathrm{t}}}^\top h(\tilde{\mathrm{t}}) \nabla_{\tilde{\mathrm{t}}}^2 f(\tilde{\mathrm{t}}, h(\mathrm{t}^*), \eta) \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}).$$

For a linear confounder function that satisfies $\nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = \beta$, such terms can be expressed as $\beta^\top \nabla_{\tilde{\mathrm{t}}}(\nabla_{\tilde{\mathrm{t}}} f(\tilde{\mathrm{t}}, h(\mathrm{t}^*), \eta)^\top \beta) = \beta^\top \nabla_{\tilde{\mathrm{t}}}(0) = 0$ under c-redundancy. Thus, such error does not accumulate even with large step sizes.

Further, note that the gradient flow equation in lode for the causal model A in section 6.4 is a linear ODE whose solution has a closed form expression and one can estimate the surrogate without numerical integration [261].

E.1.4 Proof of sufficiency of Effect Connectivity

Theorem E.3. *Under Effect Connectivity, eq. (6.9), any surrogate intervention* $\mathrm{t}'(\mathrm{t}^*, h(\mathrm{t}_2^*)) \in supp(\mathbf{t})$.

Proof. Recall $\phi(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}})) = \mathbb{E}_{\boldsymbol{\eta}} f(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}), \boldsymbol{\eta})$. We have $\forall \mathrm{t}^* \in \mathrm{supp}(p(\mathbf{t}))$:

$$p(h(\mathbf{t}) = h(\mathrm{t}_2^*)) > 0 \implies p(\phi(\mathbf{t}, h(\mathbf{t})) = \phi(\mathrm{t}^*, h(\mathrm{t}_2^*)) \mid h(\mathbf{t}) = h(\mathrm{t}_2^*)) > 0.$$

This implies $\exists \mathrm{t}' \in \mathrm{supp}(\mathbf{t}), \phi(\mathrm{t}', h(\mathrm{t}_2^*)) = \phi(\mathrm{t}^*, h(\mathrm{t}_2^*)), \quad s.t. \quad h(\mathrm{t}') = h(\mathrm{t}_2^*)$.

Then, $\phi(\mathrm{t}^*, h(\mathrm{t}_2^*)) = \phi(\mathrm{t}', h(\mathrm{t}_2^*)) = \phi(\mathrm{t}', h(\mathrm{t}')) = \mathbb{E}[\mathbf{y} \mid \mathbf{t} = \mathrm{t}']$. □

E.1.5 Necessity of Effect Connectivity for Nonparametric effect estimation in efc

Theorem E.4. *Effect Connectivity is necessary for nonparametric effect estimation in* EFC.

Proof. (Proof of Theorem 6.4) Let the outcome be $\mathbf{y} = f(\mathbf{t}, h(\mathbf{t}))$. Recall the joint distribution $p(\mathbf{t}, \mathbf{y})$ and let $h(\mathbf{t})$ be the confounder. Let Effect Connectivity be violated, i.e. there exists a non-measure-zero subset $B \in \mathrm{supp}(\mathbf{t}) \times \mathrm{supp}(h(\mathbf{t}))$ such that [*]:

$$\forall\ \tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2) \in B, \qquad p(f(\mathbf{t}, h(\mathbf{t})) = f(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2)) \mid h(\mathbf{t}) = h(\tilde{\mathrm{t}}_2)) = 0.$$

Now, we construct a new outcome $\mathbf{y}_2 = f_2(\mathbf{t}, h(\mathbf{t}))$ and show the conditional effects for this new outcome are different from the one defined by f on $\forall(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2)) \in B$. Let

$$f_2(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2)) = f(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2)) + 10 * 1((\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2)) \in B)|.$$

[*] Non-zero w.r.t. the product measure over $\mathrm{supp}(\mathbf{t}) \times \mathrm{supp}(h(\mathbf{t}))$ due to p.

We have $f_2(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}})) = f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}))\, \forall \tilde{\mathtt{t}} \in \text{supp}(\mathbf{t})$, as the additional term in f_2 is only present for $(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2)) \in B$; this follows from the fact that $\forall \tilde{\mathtt{t}} \in \text{supp}(\mathbf{t})$, $(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}})) \notin B$ as

$$p[f(\mathbf{t}, h(\mathbf{t})) = f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}})) \mid h(\mathbf{t}) = h(\tilde{\mathtt{t}})] = p[f(\mathbf{t}, h(\mathbf{t})) = f(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}))] > 0.$$

Thus, $p(\mathbf{y}, \mathbf{t}) =^d p(\mathbf{y}_2, \mathbf{t})$ are equal in distribution since $B \cap \text{supp}(\mathbf{t}, h(\mathbf{t})) = \emptyset$. This means that the conditional effects are different for the outcomes $\mathbf{y}, \mathbf{y}_2$ for all $(\tilde{\mathtt{t}}, h(\tilde{\mathtt{t}}_2)) \in B$:

$$\mathbb{E}[\mathbf{y} \mid do(\mathbf{t} = \tilde{\mathtt{t}}), h(\mathbf{t}) = h(\tilde{\mathtt{t}}_2)] \neq \mathbb{E}[\mathbf{y}_2 \mid do(\mathbf{t} = \tilde{\mathtt{t}}), h(\mathbf{t}) = h(\tilde{\mathtt{t}}_2)]$$

Therefore, for causal models that violates Effect Connectivity, there exist observationally equivalent causal models with different causal effects. Thus, nonparametric effect estimation is impossible. Thus, Effect Connectivity is required for EFC. □

E.1.6 Algorithmic details

We give in algorithm 4 pseudocode for LODE.

Algorithm 4: LODE for $do(\mathbf{t} = \mathtt{t}^*)$

```
Input: Functional confounder $h(\mathbf{t})$; tolerance $\epsilon$
Output: Conditional effects of $\mathtt{t}^*, h(\mathtt{t}_2^*)$
Regress $\mathbf{y}$ on $\mathbf{t}$ and compute $\hat{f}() := \arg\min_{u \in \mathcal{F}} \mathbb{E}_{\mathbf{y},\mathbf{t}}(\mathbf{y} - u(\mathbf{t}))^2$.
To estimate effects of $\mathtt{t}^*, h(\mathtt{t}_2^*)$, compute the surrogate intervention $\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*))$ by Euler
  integrating the gradient flow equation, initialized at $\tilde{\mathtt{t}} = \mathtt{t}^*$, until $(h(\tilde{\mathtt{t}}_s) - h(\mathtt{t}_2^*))^2 < \epsilon$.

    $$\frac{d\tilde{\mathtt{t}}(s)}{ds} = \nabla_{\tilde{\mathtt{t}}}(h(\tilde{\mathtt{t}}_s) - h(\mathtt{t}_2^*))^2,$$

Return $\hat{f}(\mathtt{t}'(\mathtt{t}^*, h(\mathtt{t}_2^*)))$;
```

Extensions of lode Consider that we have access to $m(h(\mathbf{t}))$ for some bijective differentiable function $m(\cdot)$, instead of $h(\mathbf{t})$. The orthogonality in c-redundancy holds

$$\nabla_{\tilde{\mathrm{t}}} f(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2), \eta)^T \nabla_{\tilde{\mathrm{t}}} m(h(\tilde{\mathrm{t}})) = m'(h(\tilde{\mathrm{t}})) \nabla_{\tilde{\mathrm{t}}} f(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2), \eta)^T \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = 0.$$

Then, using $m(h(\tilde{\mathrm{t}}))$ to compute the surrogate $\mathrm{t}'(\mathrm{t}^*, h(\mathrm{t}_2^*))$, lode would estimate valid effects. Similarly, lode can estimate the effect on any differentiable transformation of the outcome $m(y)$, because $\nabla_{\tilde{\mathrm{t}}} m(y_{\tilde{\mathrm{t}}})^T \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = m'(y_{\tilde{\mathrm{t}}}) \nabla_{\tilde{\mathrm{t}}} f(\tilde{\mathrm{t}}, h(\tilde{\mathrm{t}}_2), \eta)^T \nabla_{\tilde{\mathrm{t}}} h(\tilde{\mathrm{t}}) = 0$ holds.

E.2 Experimental Details

E.2.1 Functional confounders in gwas

Here, we show how $h(\mathbf{t}) = At$ and A reflect the traditional pca based adjustment in gwas. Recall population structure acts as a confounder in gwas. Price et al. [147] demonstrated that using the principal components of the normalized genetic relationships matrix adjusts for confounding due to population structure in gwas. Let the genotype matrix be G with people as rows and snps as columns, such that each element is one of $0, 1/2, 1$, where $1/2$ and 1 refer to one and two copies of the allele respectively at the position of the snp. With p_s as the allele frequency at snp s [263], Φ is the genetic relationship matrix whose elements are defined as $\Phi_{i,j} = \frac{1}{S}\sum_{s=1}^{S} {}^{(G_{i,s}-p_s)(G_{j,s}-p_s)}/_{p_s(1-p_s)}$. Then, Price et al. [147] compute the top K (10 suggested) principal components of Φ to use as the axes of variation due to the population structure. The eigenvectors of Φ are the left eigenvectors of $\hat{G}$ such that $\Phi = \hat{G}\hat{G}^T$ which capture independent axes of variation of individuals.

Price et al. [147] exploit the idea that if a snp aligns with some of the axes of variation, this is due to the population structure. These axes of variation are the top K eigenvectors U of $\phi = \hat{G}\hat{G}^T \approx U\Lambda U^\top$, where $U \in \mathbb{R}^{N\times K}$, $\Phi \in \mathbb{R}^{N\times N}$ and $\Lambda \in \mathbb{R}^{K\times K}$. Here, U are also the left singular vectors of $\hat{G} \approx U\Sigma V^T$ where $\Sigma \in \mathbb{R}^{K\times K}$ is diagonal, and $V \in \mathbb{R}^{S\times K}$. We use $\approx$ to denote that the chosen K eigenvectors explain the variation due to population structure; what remains are random mutations.

Let the sth SNP be $\hat{G}_{\cdot,s} \in \mathbb{R}^N$, which is a column in $\hat{G}$. In Price et al. [147], population structure in the sth SNP is captured in $\hat{G}_{\cdot,s}^\top U$. In words, projecting the SNP $\hat{G}_{\cdot,s}$ onto the axes of variation in individuals gives the population structure between sth SNP and the outcome. This projection $\hat{G}_{\cdot,s}^\top U$ is a row of $\hat{G}^\top U \in \mathbb{R}^{S\times K}$. In turn, $\hat{G}^\top U \in \mathbb{R}^{S\times K}$ is the population structure in all SNPs. Projecting this population structure onto the genotype of an individual gives the confounding due to population structure amongst the SNPs present in the genotype. With $G_{j,\cdot} \in \{0, 1/2, 1\}^S$ as the genotype for an individual j, this projection is $\left((\hat{G}^\top U)^\top G_{j,\cdot}\right)$. However, $\hat{G} \approx U\Sigma V^T$ implies that $\hat{G}^\top U \approx V\Sigma$. Reflecting this, $h(\mathbf{t}) = \Sigma V^T \mathbf{t}$ is the functional confounder for an individual $\mathbf{t}$.

E.2.2 Expanded results

In table E.1, we list the 13 SNPs recovered by LODE, that have been previously reported as relevant to Celiac disease. In fig. E.1, we plot the true positive and false negative rate amongst SNPs deemed relevant by LODE. The ground truth here are the SNPs reported associated with celiac disease in prior literature.

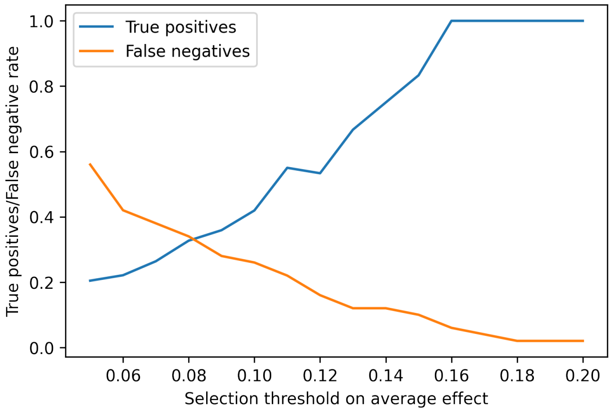


Figure E.1: True positive vs. False negative rate as we vary the threshold on average effects, that determines which SNPs LODE deems relevant to the outcome.

SNP	Effect	Lasso Coef.
rs3748816	0.12	0.20
rs10903122	0.10	0.17
rs2816316	0.11	0.20
rs13151961	0.17	0.32
rs2237236	0.17	0.00
rs12928822	0.14	0.29
rs2187668	−0.70	−2.37
rs2327832	−0.12	−0.20
rs1738074	−0.16	−0.23
rs11221332	−0.15	−0.24
rs653178	−0.13	−0.21
rs4899260	−0.12	−0.19
rs17810546	−0.12	−0.20

Table E.1: Full list of SNPs previously reported as relevant that were recovered by LODE, and their estimated effects and Lasso coefficients for SNPs. The effect threshold here is 0.1.

F | Appendices for chapter 7

F.1 Additional Figures and Tables

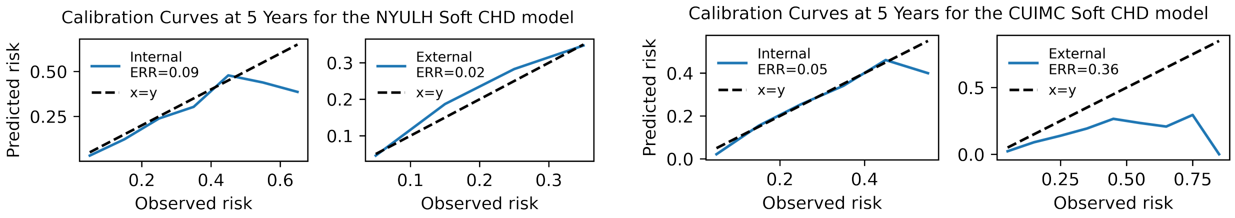
Lab and vital clipping ranges are reported in Table F.5.

F.1.1 Calibration curves

In figs. F.1 and F.2(b), we report calibration curves for 5-year risk where we adjust for censoring by weighting with the inverse-probability of censoring. For both outcomes, NYULH models calibration error under 0.1 both internally and externally; externally, this error seems to come from over-predicting risk. In contrast, CUIMC models internally have calibration error under 0.1 but dramatically under-predict risk when transported. We suspect that this loss of calibration stems NYULH data having earlier event times than CUIMC data: in the NYULH test data, nearly 90% of the uncensored patients have event times under 5 years but in the CUIMC test data, that fraction is only 66%.

F.1.2 Addressing censoring and label-leakage concerns.

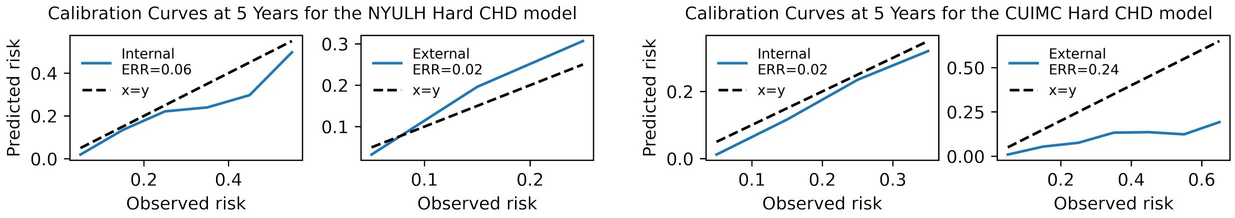
One concern in evaluating survival models is that censoring hides how well the model performs on patients that were censored. A separate concern with training and evaluating on EHR data is that features that co-occur with the outcome of interest may be recorded before the event itself



(a) NYULH models

(b) CUIMC models

Figure F.1: IPCW-Calibration curves for risk at 5 years and the associated root-mean-squared calibration error for the soft CHD outcome.



(a) NYULH models

(b) CUIMC models

Figure F.2: IPCW-Calibration curves for risk at 5 years and the associated root-mean-squared calibration error for the hard CHD outcome.

and may accidentally leak the label. To address these concerns, we turn to a variant of concordance that places more emphasis on discriminating short and long term events, which is called *truncated concordance*. Briefly, concordance truncated at 5 years measures how well the model discriminates uncensored event times under 5 years from larger event times either under or over 5 years. To adjust for the censoring mechanism, we evaluate the inverse-probability-of-censoring-weighted (IPCW) truncated concordance. All the models from table 7.1 so far achieve similar discriminative performance when evaluated with IPCW) concordance truncated at 5 years; see table F.1 and table F.2. To check if label-leakage occurs and inflates the performance, we restrict the evaluation to have event times larger than a year and evaluate unweighted concordance truncated at 5 years. All the models from table 7.1 retain similar discriminative performance when evaluated with concordance truncated at 5 years on the test set restricted to have times-to-event above a year; see table F.3 and table F.4.

Table F.1: Transportability in inverse-weighted concordance truncated at 5 years for **soft CHD**; ↑ implies higher. For a model trained at an institution (given in braces), we report the metric at the internal and external institutions and also the T-val at the external institution.

	Model (trained at)	Internal	External
IPCW Conc. ↑	DeepCAT (NYULH)	0.83 (0.825, 0.836)	0.78 (0.752, 0.822)
	DeepCAT (CUIMC)	0.87 (0.854, 0.895)	0.81 (0.805, 0.821)

Table F.2: Transportability in inverse-weighted concordance truncated at 5 years for **hard CHD**; ↑ implies higher. For a model trained at an institution (given in braces), we report the metric at the internal and external institutions and also the T-val at the external institution.

	Model (trained at)	Internal	External
IPCW Conc. ↑	DeepCAT (NYULH)	0.84 (0.825, 0.847)	0.85 (0.833, 0.886)
	DeepCAT (CUIMC)	0.90 (0.882, 0.935)	0.81 (0.800, 0.825)

Table F.3: Transportability in concordance truncated at 5 years for patients whose ***soft CHD*** *times to event are above* 1 *year*; ↑ implies higher is better. For a model trained at an institution (given in braces), we report the metric at the internal and external institutions and also the T-val at the external institution. DeepCAT stratifies patients whose events are at least 1 year away,

	Model (trained at)	Internal	External
Conc. at 5 *years.* ↑	DeepCAT (NYULH)	0.84 (0.830, 0.843)	0.81 (0.804, 0.823)
	DeepCAT (CUIMC)	0.86 (0.849, 0.863)	0.80 (0.797, 0.811)

Table F.4: Transportability in concordance truncated at 5 years for patients whose **hard CHD** *times to event are above* 1 *year*; ↑ implies higher is better. For a model trained at an institution (given in braces), we report the metric at the internal and external institutions and also the T-val at the external institution.

	Model (trained at)	Internal	External
Conc. at 5 *years.* ↑	DeepCAT (NYULH)	0.84 (0.827, 0.848)	0.84 (0.823, 0.851)
	DeepCAT (CUIMC)	0.87 (0.853, 0.880)	0.81 (0.798, 0.824)

Table F.5: Clipping Ranges for Labs and Vitals. We clip a subset of labs and vitals to clinically observable values to mitigate effects of possible data entry issues.

Measurement Name	concept_id	low	high
Alanine aminotransferase [Enzymatic activity/volume] in Serum or Plasma	3006923	1.0	10000
Albumin [Mass/volume] in Serum or Plasma	3024561	1.0	20
Albumin [Mass/volume] in Serum or Plasma by Electrophoresis	3028286	1.0	20
Alkaline phosphatase [Enzymatic activity/volume] in Serum or Plasma	3035995	5.0	10000
aPTT in Platelet poor plasma by Coagulation assay	3018677	10.0	180
Aspartate aminotransferase [Enzymatic activity/volume] in Serum or Plasma	3013721	1.0	10000
Bicarbonate [Moles/volume] in Arterial blood	3008152	2.0	100
Bicarbonate [Moles/volume] in Plasma	3015473	2.0	100
Bicarbonate [Moles/volume] in Venous blood	3027273	2.0	100
Body temperature	3020891	75.0	110
C reactive protein [Mass/volume] in Serum or Plasma	3020460	0.0	300
C reactive protein [Mass/volume] in Serum or Plasma by High sensitivity method	3010156	0.0	300
Calcium [Mass/volume] in Serum or Plasma	3006906	4.0	20
Carbon dioxide, total [Moles/volume] in Serum or Plasma	3015632	4.0	130
Chloride [Moles/volume] in Arterial blood	3031248	85.0	120
Chloride [Moles/volume] in Blood	3018572	85.0	120
Chloride [Moles/volume] in Serum or Plasma	3014576	85.0	120
Chloride [Moles/volume] in Venous blood	3035285	85.0	120
Cholesterol [Mass/volume] in Serum or Plasma	3027114	50.0	500
Cholesterol in HDL [Mass/volume] in Serum or Plasma	3007070	0.0	150
Cholesterol in LDL [Mass/volume] in Serum or Plasma	3028437	0.0	400
Creatinine [Mass/volume] in Blood	3051825	0.1	15
Creatinine [Mass/volume] in Serum or Plasma	3016723	0.1	15
Diastolic blood pressure	3012888	20.0	180
Erythrocytes [#/volume] in Blood by Automated count	3020416	1.0	8
Fibrin D-dimer FEU [Mass/volume] in Platelet poor plasma	3051714	0.0	10000
Fibrinogen [Mass/volume] in Platelet poor plasma by Coagulation assay	3016407	10.0	1000
Glucose [Mass/volume] in Blood	3000483	10.0	1500
Glucose [Mass/volume] in Serum or Plasma	3004501	10.0	1500
Heart rate	3027018	25.0	250
Hematocrit [Volume Fraction] of Blood	3009542	15.0	60
Hematocrit [Volume Fraction] of Blood by Automated count	3023314	15.0	60
Hemoglobin [Mass/volume] in Blood	3000963	2.5	19
Hemoglobin A1c/Hemoglobin.total in Blood	3004410	2.0	25
INR in Platelet poor plasma by Coagulation assay	3022217	0.1	15
Platelets [#/volume] in Blood by Automated count	3024929	0.0	2000
Potassium [Moles/volume] in Serum or Plasma	3023103	2.5	7
Protein [Mass/volume] in Serum or Plasma	3020630	2.0	10
Prothrombin time (PT)	3034426	5.0	50
Sodium [Moles/volume] in Serum or Plasma	3019550	100.0	180
Respiratory rate	3024171	5.0	100
Sodium [Moles/volume] in Arterial blood	3043706	100.0	180
Sodium [Moles/volume] in Blood	3000285	100.0	180
Sodium [Moles/volume] in Venous blood	3041473	100.0	180
Systolic blood pressure	3004249	40.0	300
Triglyceride [Mass/volume] in Serum or Plasma	3022192	15.0	1500
Urea nitrogen [Mass/volume] in Blood	3004295	2.0	130
Urea nitrogen [Mass/volume] in Serum or Plasma	3013682	2.0	130
Leukocytes [#/volume] in Blood by Automated count	3000905	0.0	500
Platelet mean volume [Entitic volume] in Blood	3001123	0.0	16
MCHC [Mass/volume] by Automated count	3009744	0.0	45
MCH [Entitic mass] by Automated count	3012030	10.0	50
Erythrocyte distribution width [Ratio] by Automated count	3019897	0.0	40
MCV [Entitic volume] by Automated count	3023599	50.0	140
Bilirubin.total [Mass/volume] in Serum or Plasma	3024128	0.0	50
Age		0.0	120

Table F.6: Table of the performance numbers from the results section.

Model	Data	Value	Notes
FRS	CUIMC	≤ .75	
FRS	NYULH	≤ .75	
PCE	CUIMC	≤ .75	
PCE	NYULH	≤ .75	
PREVENT	CUIMC	≤ .78	
PREVENT	NYULH	≤ .78	
CUIMC	CUIMC	.85	best external
NYULH	NYULH	.84	best external
CUIMC	NYULH	0.813 (0.807, 0.821)	best external
NYULH	CUIMC	0.809 (0.801, 0.816)	best external
NYULH	NYULH	0.839 (0.833, 0.846)	best internal
NYULH	CUIMC	0.782 (0.774, 0.789)	best internal
CUIMC	NYULH	0.811 (0.805, 0.817)	best internal
NYULH	NYULH	0.764 (0.757, 0.772)	Age, Smoking, Measurements
NYULH	CUIMC	0.746 (0.737, 0.752)	Age, Smoking, Measurements
CUIMC	CUIMC	0.798 (0.790, 0.806)	Age, Smoking, Measurements
CUIMC	CUIMC	0.808 (0.800, 0.815)	All except conditions
CUIMC	CUIMC	0.850 (0.843, 0.857)	All including conditions
CUIMC	NYULH	0.769 (0.764, 0.775)	All except conditions
CUIMC	NYULH	0.805 (0.799, 0.813)	All including conditions
NYULH	NYULH	0.796 (0.791, 0.804)	All except conditions
NYULH	NYULH	0.838 (0.832, 0.844)	All including conditions
NYULH	CUIMC	0.752 (0.746, 0.761)	All except conditions
NYULH	CUIMC	0.789 (0.782, 0.796)	All including conditions
NYULH	NYULH	0.846 (0.84, 0.855)	best external, on females
NYULH	NYULH	0.818 (0.808, 0.826)	best external, on males
CUIMC	CUIMC	0.855 (0.847, 0.864)	best external, on females
CUIMC	CUIMC	0.825 (0.818, 0.834)	best external, on males
CUIMC	CUIMC	0.958 (0.932, 0.99)	best external, on Asian females
CUIMC	CUIMC	0.818 (0.795, 0.843)	best external, on Hispanic males
CUIMC	CUIMC	0.851 (0.846, 0.856)	All without measurements
CUIMC	CUIMC	0.850 (0.845, 0.855)	All including measurements
CUIMC	NYULH	0.801 (0.795, 0.809)	All without measurements
CUIMC	NYULH	0.805 (0.799, 0.812)	All including measurements
NYULH	NYULH	0.837 (0.831, 0.843)	All without measurements
NYULH	NYULH	0.838 (0.832, 0.844)	All including measurements
NYULH	CUIMC	0.800 (0.794, 0.809)	All without measurements
NYULH	CUIMC	0.789 (0.782, 0.796)	All including measurements

F.2 Additional Cohort Information

We report patients statistics in Table F.7, data statistics in Table F.8. We give details about the smoking categories in Table F.10, about demographics in Table F.9, about labs in Table F.11, about vitals in Table F.12. about conditions in Table F.13, and about medications in Table F.14.

Table F.7: CAD Cohort Patient Characteristics. Patient demographics are reported in the full cohorts and just among the cases who are positive for a coronary artery disease event. The dataset sizes are quite similar, but the NYU cohort a higher proportion of White and Asian patients. Both cohorts have more Females than Males.

	CUIMC		NYUL	
	All	Positive Cases	All	Positive Cases
# Patients	1,326,380	71,169	1,743,006	84,150
# CAD Events	71,169 (5.4%)	–	84,150 (4.8%)	–
# Censored Events	1,255,211 (94.6%)	–	1,658,856 (95.2%)	–
# of Datapoints	19,432,807	1,555,688	25,227,328	1,540,442
Race				
Unknown/ Missing	720,413 (54.3%)	37,498 (52.7%)	379,587 (21.8%)	11,362 (13.5%)
White	445,864 (33.6%)	24,640 (34.6%)	1,048,894 (60.2%)	58,874 (70.0%)
Black	112,247 (8.5%)	7577 (10.6%)	190,631 (10.9%)	9024 (10.7%)
Asian	31,287 (2.4%)	919 (1.3%)	101453 (5.8%)	3668 (4.4%)
Native American/ Pacific Islander	16,569 (1.2%)	535 (0.8%)	22,441 (1.3%)	1222 (1.5%)
Ethnicity				
Unknown/ Missing	681,966 (51.4%)	32,502 (45.7%)	1,480,432 (84.9%)	59,673 (70.9%)
Non H/L	429,361 (32.4%)	22,093 (31.0%)	237,131 (13.6%)	22,531 (26.8%)
Hispanic/ Latino	215,053 (16.2%)	16,574 (23.3%)	25,443 (1.5%)	1946 (2.3%)
Sex				
Female	843,699 (63.6%)	37,748 (53.0%)	1,084,102 (62.2%)	41,641 (49.5%)
Male	482,323 (36.4%)	33,402 (46.9%)	658,486 (37.8%)	42,503 (50.5%)
Unknown/ Missing	358 (0.0%)	19 (0.0%)	418 (0.0%)	6 (0.0%)
Age	52.1 ± 18.9	60.8 ± 15.9	56.1 ± 17.7	68.5 ± 13.3
Age Groups				
[18, 30)	2,899,903 (14.9%)	58,889 (3.8%)	238,1527 (9.4%)	13,443 (0.9%)
[30, 40)	3,212,159 (16.5%)	114,886 (7.4%)	3,131,650 (12.4%)	30,966 (2.0%)
[40, 50)	2,892,979 (14.9%)	213,881 (13.7%)	3,513,713 (13.9%)	89,841 (5.8%)
[50, 60)	3,186,021 (16.4%)	316,175 (20.3%)	4,833,097 (19.2%)	242,996 (15.8%)
[60, 70)	3,264,696 (16.8%)	368,797 (23.7%)	5,317,300 (21.1%)	422,373 (27.4%)
[70, 80)	2,573,369 (13.2%)	313,129 (20.1%)	3,936,222 (15.6%)	432,717 (28.1%)
80+	1,403,680 (7.2%)	169,931 (10.9%)	2,113,819 (8.4%)	308,106 (20.0%)

Table F.8: CAD Cohort Dataset Characteristics. For each patient, whether they are positive for a CAD event or are censored, we define multiple data points; each data point is a unique interaction with the health system where some observations are recorded binned on a month level. We show the statistics across the entire dataset, where there are multiple data points for each patient.

	CUIMC		NYUL	
	All	Positive Cases	All	Positive Cases
# Patients	1,326,380	71,169	1,743,006	84,150
# of Datapoints	19,432,807	1,555,688	25,227,328	1,540,442
# Datapoints: CAD Cases	1,555,688 (8.0 %)	1,555,688	1,540,442 (6.1 %)	1,540,442
# Datapoints: Censored Cases	17877119 (92.0 %)	–	23,686,886 (93.9 %)	–
Race				
Unknown/Missing	9,340,132 (48.1 %)	857,209 (55.1 %)	4,479,356 (17.8 %)	186,612 (12.1 %)
White	7,541,770 (38.8 %)	488,850 (31.4 %)	16,238,504 (64.4 %)	1,103,663 (71.6 %)
Black	1,916,310 (9.9 %)	180,021 (11.6 %)	2,801,003 (11.1 %)	162,328 (10.5 %)
Asian	462,993 (2.4 %)	18,396 (1.2 %)	1,333,863 (5.3 %)	63,498 (4.1 %)
Native American/ Pacific Islander	171,602 (0.9 %)	11,212 (0.7 %)	374,602 (1.5 %)	24,341 (1.6 %)
Ethnicity				
Unknown/Missing	7,534,774 (38.8 %)	617,670 (39.7 %)	19,692,029 (78.1 %)	973,465 (63.2 %)
Non H/L	7,515,475 (38.7 %)	449,950 (28.9 %)	5,011,997 (19.9 %)	520,911 (33.8 %)
Hispanic/ Latino	4,382,558 (22.6 %)	488,068 (31.4 %)	523,302 (2.1 %)	46,066 (3.0 %)
Sex				
Female	1,320,0127 (67.9 %)	947,477 (60.9 %)	16,007,915 (63.5 %)	808,741 (52.5 %)
Male	6,228,936 (32.1 %)	607,834 (39.1 %)	9,214,274 (36.5 %)	731,513 (47.5 %)
Unknown/Missing	3744 (0.0 %)	377 (0.0 %)	5139 (0.0 %)	188 (0.0 %)
Age	52.1 ± 18.9	60.8 ± 15.9	56.1 ± 17.7	68.5 ± 13.3
Age Groups				
[18, 30)	2,899,903 (14.9 %)	58,889 (3.8 %)	238,1527 (9.4 %)	13,443 (0.9 %)
[30, 40)	3,212,159 (16.5 %)	114,886 (7.4 %)	3,131,650 (12.4 %)	30,966 (2.0 %)
[40, 50)	2,892,979 (14.9 %)	213,881 (13.7 %)	3,513,713 (13.9 %)	89,841 (5.8 %)
[50, 60)	3,186,021 (16.4 %)	316,175 (20.3 %)	4,833,097 (19.2 %)	242,996 (15.8 %)
[60, 70)	3,264,696 (16.8 %)	368,797 (23.7 %)	5,317,300 (21.1 %)	422,373 (27.4 %)
[70, 80)	2,573,369 (13.2 %)	313,129 (20.1 %)	3,936,222 (15.6 %)	432,717 (28.1 %)
80+	1,403,680 (7.2 %)	169,931 (10.9 %)	2,113,819 (8.4 %)	308,106 (20.0 %)

Table F.9: Demographic Mapping. Race, ethnicity and gender were encoded in the following way.

Race Encoding		
concept_id	concept name	encoding
8527	White	0
8516	Black or African American	1
8515	Asian	2
8557	Native Hawaiian or Other Pacific Islander	3
8657	American Indian or Alaska Native	3
38003610	Polynesian	3
38003613	Other Pacific Islander	3
8552	Unknown	4
0	No matching concept	4
44814653	Unknown	4
Ethnicity		
38003564	Not Hispanic or Latino	0
38003563	Hispanic or Latino	1
0	No Matching Concept	2
-1	Unknown	2
8552	Unknown	2
Sex		
8507	Male	0
8532	Female	1
0	No Matching Concept	2
-1	Unknown	2
8522	Unknown	2

Table F.10: Smoking Status Encoding. We identified shared higher-level categories to map existing concepts to categorical encodings.

concept_id	concept_name	encoding	smoking status
4184633	Passive smoker	0	non smoker
4144272	Never smoked tobacco	0	non smoker
4022662	Non-smoker for personal reasons	0	non smoker
37018706	At risk from passive smoking	0	non smoker
4222303	Non-smoker	0	non smoker
764104	Cigarette smoker (1-4 cigarettes/day)	1	light smoker
4042037	Light cigarette smoker	1	light smoker
4044775	Occasional cigarette smoker	1	light smoker
37395605	Occasional tobacco smoker	1	light smoker
4144273	Trivial cigarette smoker (≤ 1 cigarette/day)	1	light smoker
4052029	Light cigarette smoker (1-9 cigs/day)	1	light smoker
762498	Light tobacco smoker	1	light smoker
4209585	Moderate smoker (20 or less per day)	2	moderate smoker
4190573	Thinking about stopping smoking	2	moderate smoker
4246415	Cigar smoker	2	moderate smoker
4269997	Tobacco smoking consumption - finding	2	moderate smoker
4215409	Ready to stop smoking	2	moderate smoker
4275495	Tobacco smoking behavior - finding	2	moderate smoker
4276526	Cigarette smoker	2	moderate smoker
4298794	Smoker	2	moderate smoker
4218917	Pipe smoker	2	moderate smoker
4216174	Not interested in stopping smoking	2	moderate smoker
44784248	Hookah pipe smoker	2	moderate smoker
42709996	Smokes tobacco daily	2	moderate smoker
4052030	Moderate cigarette smoker (10-19 cigs/day)	2	moderate smoker
4046886	Smoking reduced	2	moderate smoker
4058137	Trying to give up smoking	2	moderate smoker
4044776	Moderate cigarette smoker	2	moderate smoker
762499	Heavy tobacco smoker	3	heavy smoker
4209006	Heavy smoker (over 20 per day)	3	heavy smoker
4041511	Heavy cigarette smoker	3	heavy smoker
4044777	Very heavy cigarette smoker	3	heavy smoker
4044778	Chain smoker	3	heavy smoker
4052947	Heavy cigarette smoker (20-39 cigs/day)	3	heavy smoker
4058136	Very heavy cigarette smoker (40+ cigs/day)	3	heavy smoker
42536346	Ex-smoker for less than 1 year	4	ex-smoker
40486721	Stopped smoking during pregnancy	4	ex-smoker
4310250	Ex-smoker	4	ex-smoker
4232375	Aggressive ex-smoker	4	ex-smoker
4145798	Ex-light cigarette smoker (1-9/day)	4	ex-smoker
4148416	Ex-cigarette smoker amount unknown	4	ex-smoker
762500	Former heavy tobacco smoker	4	ex-smoker
762501	Former light tobacco smoker	4	ex-smoker
4052032	Stopped smoking	4	ex-smoker
4052465	Ex-pipe smoker	4	ex-smoker
4052949	Ex-cigar smoker	4	ex-smoker
4207221	Tolerant ex-smoker	4	ex-smoker
4092281	Ex-cigarette smoker	4	ex-smoker
4141783	Ex-heavy cigarette smoker (20-39/day)	4	ex-smoker
4141784	Ex-very heavy cigarette smoker (40+/day)	4	ex-smoker
4148415	Ex-trivial cigarette smoker (<1/day)	4	ex-smoker
4141782	Ex-moderate cigarette smoker (10-19/day)	4	ex-smoker
46270534	Ex-smoker for more than 1 year	4	ex-smoker
4141786	Tobacco smoking consumption unknown	5	unknown
44814653	Unknown	5	unknown
4233486	Current non-smoker but history unknown	5	unknown
44814650	No information	5	unknown

Table F.11: Lab Measurements. We included the following key lab measurements for risk assessment.

Lab Test Name	concept_id
MCHC [Mass/volume] by Automated count	3009744
Diastolic blood pressure	3012888
Chloride [Moles/volume] in Blood	3018572
Hemoglobin A1c/Hemoglobin.total in Blood	3004410
Bilirubin.total [Mass/volume] in Serum or Plasma	3024128
Aspartate aminotransferase [Enzymatic activity/volume] in Serum or Plasma	3013721
Carbon dioxide, total [Moles/volume] in Serum or Plasma	3015632
Hemoglobin [Mass/volume] in Blood	3000963
Alkaline phosphatase [Enzymatic activity/volume] in Serum or Plasma	3035995
Erythrocyte distribution width [Ratio] by Automated count	3019897
Potassium [Moles/volume] in Serum or Plasma	3023103
C reactive protein [Mass/volume] in Serum or Plasma	3020460
Bicarbonate [Moles/volume] in Plasma	3015473
Sodium [Moles/volume] in Venous blood	3041473
Cholesterol in LDL [Mass/volume] in Serum or Plasma	3028437
Cholesterol in HDL [Mass/volume] in Serum or Plasma	3007070
MCV [Entitic volume] by Automated count	3023599
Glucose [Mass/volume] in Blood	3000483
Urea nitrogen [Mass/volume] in Blood	3004295
Platelet mean volume [Entitic volume] in Blood	3001123
Sodium [Moles/volume] in Blood	3000285
Chloride [Moles/volume] in Serum or Plasma	3014576
Sodium [Moles/volume] in Arterial blood	3043706
Glucose [Mass/volume] in Serum or Plasma	3004501
Platelets [#/volume] in Blood by Automated count	3024929
Respiratory rate	3024171
Heart rate	3027018
Albumin [Mass/volume] in Serum or Plasma by Electrophoresis	3028286
Cholesterol [Mass/volume] in Serum or Plasma	3027114
Prothrombin time (PT)	3034426
INR in Platelet poor plasma by Coagulation assay	3022217
Creatinine [Mass/volume] in Blood	3051825
MCH [Entitic mass] by Automated count	3012030
Triglyceride [Mass/volume] in Serum or Plasma	3022192
Urea nitrogen [Mass/volume] in Serum or Plasma	3013682
Bicarbonate [Moles/volume] in Arterial blood	3008152
aPTT in Platelet poor plasma by Coagulation assay	3018677
C reactive protein [Mass/volume] in Serum or Plasma by High sensitivity method	3010156
Fibrinogen [Mass/volume] in Platelet poor plasma by Coagulation assay	3016407
Erythrocytes [#/volume] in Blood by Automated count	3020416
Chloride [Moles/volume] in Arterial blood	3031248
Bicarbonate [Moles/volume] in Venous blood	3027273
Creatinine [Mass/volume] in Serum or Plasma	3016723
Albumin [Mass/volume] in Serum or Plasma	3024561
Hematocrit [Volume Fraction] of Blood by Automated count	3023314
Leukocytes [#/volume] in Blood by Automated count	3000905
Systolic blood pressure	3004249
Body temperature	3020891
Calcium [Mass/volume] in Serum or Plasma	3006906
Alanine aminotransferase [Enzymatic activity/volume] in Serum or Plasma	3006923
Chloride [Moles/volume] in Venous blood	3035285
Hematocrit [Volume Fraction] of Blood	3009542
Sodium [Moles/volume] in Serum or Plasma	3019550
Protein [Mass/volume] in Serum or Plasma	3020630

Table F.12: Vital Signs. We included the following key vital signs for risk assessment.

Vital Sign	concept_id
Diastolic blood pressure	3012888
Respiratory rate	3024171
Heart rate	3027018
Systolic blood pressure	3004249
Body temperature	3020891

Table F.13: Conditions. We included 669 conditions for risk assessment. This table includes the top 25 occurring conditions at NYULH. Both NYULH and CUIMC use the same total set of conditions.

Condition Name	concept_id
Essential hypertension	320128
Hyperlipidemia	432867
Vitamin D deficiency	436070
Dyspnea	312437
Preoperative state	4216244
Cough	254761
Chest pain	77670
Idiopathic osteoarthritis	4035439
Gastroesophageal reflux disease without esophag...	4144111
Obesity	433736
Inconclusive mammography finding	37108814
Electrocardiogram abnormal	320536
Postoperative state	438485
Chronic pain	436096
Fatigue	4223659
Pure hypercholesterolemia	437827
Low back pain	194133
Mixed hyperlipidemia	438720
Abdominal pain	200219
Dizziness and giddiness	433316
Anxiety disorder	442077
Hypothyroidism	140673
Anemia	439777
Postprocedural state finding	444239
Blood chemistry abnormal	436230

Table F.14: Medications (ingredients). We included 452 total drug ingredients for risk assessment, as well as additional drug brands. This table includes the top 25 occurring ingredients at NYULH. Both NYULH and CUIMC use the same total set of medications.

Medication name	concept_id
lidocaine	989878
acetaminophen	1125315
sodium chloride	967823
potassium chloride	19049105
propofol	753626
lactate	19011035
calcium chloride	19036781
ondansetron	1000560
fentanyl	1154029
oxycodone	1124957
midazolam	708298
polyethylene glycol 3350	986417
bupivacaine	732893
gadobutrol	19048493
ibuprofen	1177480
ketorolac	1136980
albuterol	1154343
epinephrine	1343916
famotidine	953076
atorvastatin	1545958
azithromycin	1734104
amoxicillin	1713332
aspirin	1112807
rocuronium	19003953
bisacodyl	924939

F.3 Baselines

Baseline, Framingham Model. We use the original Framingham score [177] as the baseline model to evaluate against. This score is the basis of all existing clinical risk scores which are used in practice for CVDs. The score considers age, sex, LDL cholesterol, HDL cholesterol, systolic and diastolic blood pressure, diabetes and smoking status. The risk score provides a simple means of translating these measurements to risk points which correspond to different percentages of 10-year risk. While the score has been shown to work well in the Framingham cohort, it is known to have lower performance when applied to EHRs [200].

Baseline, Pooled Cohort Equations. We use the AHA's 2013 version of the pooled cohort equations [176]. This model uses similar features to the Framingham risk score, including cholesterol, blood pressure, smoking status, diabetes status, age and sex. They develop separate models for Black and White patients as they are trained on more divers cohorts. We use the reported parameters from [176] to estimate 10-year risk.

Baseline, PREVENT. We use the CHD version of the PREVENT score from [181, 197]. This model uses similar features to the PCE, but without race, and adds eGFR, and the use of statins and hypertensives as features. We use the reported parameters from [197] to estimate 10-year risk.

F.4 Survival Modeling Evaluation Metrics

Inverse Probability Weighting Survival analysis methods are meant to address the challenge of censored time-to-event data. Data points are censored if the event time is unobserved. Often data is right censored, indicating that the event occurred after the censoring time, but the exact time of the event is not known. There are many well-known survival analysis metrics for assessing how well a model ranks event risk (i.e., discrimination) and how well the predicted risk reflects the true risk (i.e., calibration). The inherent challenge associated with censored data is that there are a number of unobserved event times. Given this, most prior studies typically only evaluate using the observed event times. However, there are a class of estimators which adjust for censored data points called inverse-probability-of-censoring weighting (IPCW) estimators [264]. The general idea behind this class of estimators is to adjust for censored data points by assigning a weight to each data point depending on how likely the data point is to be censored based on its features. Data points which are highly likely to be censored are given more weight, while those that are often observed are given less weight. These estimators effectively compensate for censored data points by assigning more weight to similar uncensored data [265].

Computing these metrics requires fitting a censoring model: $p(c \mid x)$ where c is the censoring time and x is the input features. When evaluating the transportability of model M_A on $\mathcal{D}_B$ using a weighted metric, we use the censoring model $p^B(c \mid x)$ fit at institution B.

Discrimination Metrics We outline both unweighted and IPCW versions of concordance as our primary evaluation of discriminative performance, or how well a model is ranking different data points based on predicted risk.

Unweighted Concordance using Expected Value of Predicted Event Distribution. The concordance is computed by taking the expected value of the predicted conditional distributions for each data point as the estimated time-to-event. All comparable pairs (i.e., pairs of data points where rank order is known) are then used to assess whether the model is correctly predicting that the estimated time-to-events are in the same order as the true time-to-events. The proportion that is correct is computed as a measure of concordance. For a time-to-event model p_θ, concordance is

$$C_{exp} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{1}(\mathbb{E}_{p_\theta}[t_i \mid x_i] < \mathbb{E}_{p_\theta}[t_j \mid x_j])\mathbf{1}(t_i < t_j)\delta_i}{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{1}(t_i < t_j)\delta_i} \tag{F.1}$$

Unweighted Concordance at Particular Cutoff τ. The concordance is computed by comparing the survival probabilities at a fixed time τ and evaluating all comparable pairs where at least one of the event times occurs prior to the cutoff time τ [266]. In this case $S_\theta(t)$ is defined as the survival function, or 1 minus the CDF of the predicted conditional distribution. This version of concordance evaluates a model's discriminative capability at a specific time τ and may be useful for risk models where a fixed time horizon is important (e.g., coronary artery disease risk at 10 years).

Inverse Weighted Concordance at Particular Cutoff τ. This version of concordance is similar to the prior one, but it uses inverse probability weighting to adjust for censored data points [266]. In this case, $G(t \mid x)$ is defined as the probability that the censoring time occurs after time t. Thus, by inverse weighting by this quantity, data points which are often censored prior to this time are assigned higher weights. The denominator of the equation renormalizes the estimator according to the sum of the weights [267]. With F_θ as the CDF of the time-to-event model

$$C_{wt}(\tau) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{1}(F_\theta(\tau \mid x_i) < F_\theta(\tau \mid x_j))\mathbf{1}(t_i < t_j)\mathbf{1}(t_i < \tau)\delta_i W_{ij}^{-1}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{1}(t_i < t_j)\mathbf{1}(t_i < \tau)\delta_i W_{ij}^{-1}} \tag{F.2}$$

$$W_{ij} = G(t_i \mid x_j)G(t_i \mid x_i) \tag{F.3}$$

Calibration Metrics We outline several metrics for assessment of calibration, a measure of how close predicted risk is to true risk.

Inverse Weighted Brier Score. The brier score in binary classification problems is defined simply as the average squared error between the predicted probabilities and the true outcome (0 or 1). In the case of survival analysis, the event times are separated into indicators using a particular time τ. All times prior to τ are assigned a 1 and all times after τ are assigned a 0. The predicted risk is the CDF of the conditional distribution. Since, there may be some censored times, we use IPCW to re-weight. We define $F_\theta(t \mid x)$ as the CDF of the predicted conditional distribution. In this case T_i is the true event time, which is sometimes not known due to censorship. Therefore, instead of computing equation F.4, we can compute equation F.5. For a full derivation see [216].

$$BS(\tau) = \frac{1}{N} = \sum_{i=1}^{N} (F_\theta(t_i \mid x_i) - \mathbf{1}(T_i < \tau))^2 \tag{F.4}$$

$$BS_{wt}(\tau) = \frac{1}{N} \sum_{i=1}^{N} \frac{(1 - F_\theta(\tau \mid x_i))^2 \delta_i \mathbf{1}(t_i < \tau)}{G(t_i \mid x_i)} + \frac{F_\theta(\tau \mid x_i)^2 \mathbf{1}(t_i > \tau)}{G(\tau \mid x_i)} \tag{F.5}$$

Inverse Weighted Binomial Log Likelihood The binomial log likelihood is similar to the brier score, but we compute negative log likelihood instead of squared error.

$$BLL_{wt}(\tau) = \frac{1}{N} \sum_{i=1}^{N} \frac{-\log(F_\theta(\tau \mid x_i))\delta_i \mathbf{1}(t_i < \tau)}{G(t_i \mid x_i)} + \frac{-\log(1 - F_\theta(\tau \mid x_i))\mathbf{1}(t_i > \tau)}{G(\tau \mid x_i)} \tag{F.6}$$

Inverse Weighted Calibration Curve. Calibration curves are commonly computed in binary classification tasks. This curve provides some understanding of how well the predicted risk corresponds to the true risk. The predicted risks are discretized into bins and within each bin, the frequency of data points which correspond to positive cases is computed. For each bin, this frequency should match the mean of the bin if the model is properly calibrated. In the case of survival analysis, we can use the CDF value evaluated at some τ as the risk of the event having already happened. Then, we can assign this risk to some bin and then count the number of uncensored data points in the bin where the event has happened before τ. This count is censor-weighted and then divided by the total number of bin members. The following estimates the risk inside the bin boundaries b_1 and b_2 at time τ.

$$CC_{wt}(\tau, b_1, b_2) = \frac{\sum_{i=1}^{N} \mathbf{1}(b_1 \leq F_\theta(\tau \mid x_i) \leq b_2)\delta_i \mathbf{1}(t_i < \tau)W_i^{-1}}{\sum_{i=1}^{N} \mathbf{1}(b_1 \leq F_\theta(\tau \mid x_i) \leq b_2)} \tag{F.7}$$

$$W_i = G(t_i \mid x_j) \tag{F.8}$$

Model Optimization Details

The models are trained using stochastic gradient descent with the AdamW optimizer and the following hyperparameters: (dropout rate: 0.1, weight decay: 1e-2, MDN mixture components: 5, a learning: 1e-4, embedding size: 64). The following bin boundaries in months are used for the categorical version of the model: [0, 1, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114, 117, 120, 132, 144, 156, 168, 180, 192, 204, 216, 228, 240]. The bins are more granular up until the tenth year, but then we specify larger intervals for each bin beyond ten years.

Bibliography

[1] Aaron Boussina, Supreeth P Shashikumar, Atul Malhotra, Robert L Owens, Robert El-Kareh, Christopher A Longhurst, Kimberly Quintero, Allison Donahue, Theodore C Chan, Shamim Nemati, et al. Impact of a deep learning sepsis prediction model on quality of care and survival. *npj Digital Medicine*, 7(1):14, 2024.

[2] Nathan Nunn and Leonard Wantchekon. The slave trade and the origins of mistrust in africa. *American Economic Review*, 101(7):3221–52, 2011.

[3] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[5] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks, 2020.

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[7] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of

why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[8] Zhao Wang and Aron Culotta. Identifying spurious correlations for robust text classification. *arXiv preprint arXiv:2010.02458*, 2020.

[9] 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.

[10] Marcus A Badgeley, John R Zech, Luke Oakden-Rayner, Benjamin S Glicksberg, Manway Liu, William Gale, Michael V McConnell, Bethany Percha, Thomas M Snyder, and Joel T Dudley. Deep learning predicts hip fracture using confounding patient and healthcare variables. *NPJ digital medicine*, 2(1):1–10, 2019.

[11] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[12] 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). *arXiv preprint arXiv:2003.00688*, 2020.

[13] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

[14] Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman. Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix. *arXiv preprint arXiv:2101.07732*, 2021.

[15] Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset

shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3118–3127. PMLR, 2019.

[16] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[17] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

[18] Maggie Makar, Ben Packer, Dan Moldovan, Davis Blalock, Yoni Halpern, and Alexander D'Amour. Causally-motivated shortcut removal using auxiliary labels. *arXiv preprint arXiv:2105.06422*, 2021.

[19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[20] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–331. Springer, 2019.

[21] Alexis Bellot and Mihaela van der Schaar. Accounting for unobserved confounding in domain generalization. *arXiv preprint arXiv:2007.10653*, 2020.

[22] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[23] Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the sub-

group performance gap with data augmentation. *arXiv preprint arXiv:2008.06775*, 2020.

[24] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *arXiv preprint arXiv:2102.10395*, 2021.

[25] 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.

[26] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. *arXiv preprint arXiv:1705.11122*, 2017.

[27] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

[28] Sandesh Ghimire, Satyananda Kashyap, Joy T Wu, Alexandros Karargyris, and Mehdi Moradi. Learning invariant feature representation to improve generalization across chest x-ray datasets. In *International Workshop on Machine Learning in Medical Imaging*, pages 644–653. Springer, 2020.

[29] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. Representation learning with statistical independence to mitigate bias. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2513–2523, 2021.

[30] Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021.

[31] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization:

A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

[32] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. Fairness without demographics through adversarially reweighted learning. *arXiv preprint arXiv:2006.13114*, 2020.

[33] 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.

[34] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[35] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

[36] Adarsh Subbaswamy, Bryant Chen, and Suchi Saria. A universal hierarchy of shift-stable distributions and the tradeoff between stability and performance. *arXiv preprint arXiv:1905.11374*, 2019.

[37] Erich L Lehmann and George Casella. *Theory of point estimation.* Springer Science & Business Media, 2006.

[38] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[39] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng.

Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[40] Eduardo HP Pooch, Pedro L Ballester, and Rodrigo C Barros. Can we trust deep learning models diagnosis? the impact of domain shift in chest radiograph classification. *arXiv preprint arXiv:1909.01940*, 2019.

[41] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019.

[42] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[43] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814. PMLR, 2018.

[44] Mark Goldstein, Xintian Han, Aahlad Puli, Adler Perotte, and Rajesh Ranganath. X-cal: Explicit calibration for survival analysis. *Advances in Neural Information Processing Systems*, 2020.

[45] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[46] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[47] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. *arXiv preprint arXiv:1909.06321*, 2019.

[48] Aahlad Manas Puli, Lily H Zhang, Eric Karl Oermann, and Rajesh Ranganath. Out-of-

distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=12RoR2o32T.

[49] 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*, 78(5):947–1012, 2016.

[50] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[51] Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-domain robustness via targeted augmentations. *arXiv preprint arXiv:2302.11861*, 2023.

[52] Amir Feder, Yoav Wald, Claudia Shi, Suchi Saria, and David Blei. Data augmentations for improved (large) language model generalization. 2023. URL https://api.semanticscholar.org/CorpusID:264305897.

[53] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[54] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR, 2022.

[55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[56] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from

failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

[57] Zongbo Han, Zhipeng Liang, Fan Yang, Liu Liu, Lanqing Li, Yatao Bian, Peilin Zhao, Bingzhe Wu, Changqing Zhang, and Jianhua Yao. Umix: Improving importance weighting for subpopulation shift via uncertainty-aware mixup. *Advances in Neural Information Processing Systems*, 35:37704–37718, 2022.

[58] He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. *arXiv preprint arXiv:1908.10763*, 2019.

[59] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[60] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[61] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

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

[63] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

[64] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[65] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

[66] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019.

[67] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR, 2020.

[68] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *ICML*, 2020.

[69] Yao Qin, Chiyuan Zhang, Ting Chen, Balaji Lakshminarayanan, Alex Beutel, and Xuezhi Wang. Understanding and improving robustness of vision transformers through patch-based negative augmentation. *arXiv preprint arXiv:2110.07858*, 2021.

[70] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[71] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019.

[72] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano

Ermon. Negative data augmentation. *arXiv preprint arXiv:2102.05113*, 2021.

[73] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[74] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 580–599. Springer, 2020.

[75] Amir Feder, Katherine A Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E Roberts, et al. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158, 2022.

[76] Divyansh Kaushik, Amrith Setlur, Eduard Hovy, and Zachary C Lipton. Explaining the efficacy of counterfactually augmented data. *arXiv preprint arXiv:2010.02114*, 2020.

[77] Jacob Eisenstein. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, July 2022. URL https://aclanthology.org/2022.naacl-main.321.

[78] Zhao Wang and Aron Culotta. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14024–14031, 2021.

[79] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M

Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/koh21a.html.

[80] Aahlad Manas Puli, Lily H Zhang, Eric Karl Oermann, and Rajesh Ranganath. Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=12RoR2o32T.

[81] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26484–26516. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhang22z.html.

[82] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[83] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang, editors, *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 336–351. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/idrissi22a.html.

[84] Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In *International Conference on Machine Learning*, pages 10849–10858. PMLR, 2021.

[85] Bohan Wang, Qi Meng, Huishuai Zhang, Ruoyu Sun, Wei Chen, Zhi-Ming Ma, and Tie-Yan Liu. Does momentum change the implicit regularization on separable data? *Advances in Neural Information Processing Systems*, 35:26764–26776, 2022.

[86] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

[87] Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. Malign overfitting: Interpolation and invariance are fundamentally at odds. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=dQNL7Zsta3.

[88] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[89] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[90] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[91] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348, 2021.

[92] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

[93] Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint arXiv:1907.10599*, 2019.

[94] Basri Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. *Advances in Neural Information Processing Systems*, 32, 2019.

[95] Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

[96] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018.

[97] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[98] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[99] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

[100] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

[101] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[102] Katherine L Hermann and Andrew K Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *arXiv preprint arXiv:2006.12433*, 2020.

[103] Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which shortcut cues will dnns choose? a study from the parameter-space perspective. *arXiv preprint arXiv:2110.03095*, 2021.

[104] 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*, 34:3438–3450, 2021.

[105] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[106] Mohammad Pezeshki, Amartya Mitra, Yoshua Bengio, and Guillaume Lajoie. Multi-scale feature learning dynamics: Insights for double descent. In *International Conference on Machine Learning*, pages 17669–17690. PMLR, 2022.

[107] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[108] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[109] Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. *Advances in Neural Information Processing Systems*, 34:18970–18983, 2021.

[110] Ziwei Ji, Miroslav Dudík, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *Conference on Learning Theory*, pages 2109–2136. PMLR, 2020.

[111] Saharon Rosset, Ji Zhu, and Trevor Hastie. Margin maximizing loss functions. *Advances in neural information processing systems*, 16, 2003.

[112] Alexandros Xenos, John Pavlopoulos, Ion Androutsopoulos, Lucas Dixon, Jeffrey Sorensen, and Léo Laugier. Toxicity detection sensitive to conversational context. *First Monday*, 2022.

[113] Michael Baker, Dwayne Benjamin, and Shuchita Stanger. The highs and lows of the minimum wage effect: A time-series cross-section study of the canadian law. *Journal of Labor Economics*, 17(2):318–350, 1999.

[114] Stanley Lieberson. *Making it count: The improvement of social research and theory*. Univ of California Press, 1987.

[115] Matt McGue, Merete Osler, and Kaare Christensen. Causal inference and observational research: The utility of twins. *Perspectives on psychological science*, 5(5):546–556, 2010.

[116] Kenneth J Rothman and Sander Greenland. Causation and causal inference in epidemiology. *American journal of public health*, 95(S1):S144–S150, 2005.

[117] Elizabeth A Stuart, Eva DuGoff, Michael Abrams, David Salkever, and Donald Steinwachs. Estimating causal effects in observational studies using electronic health data: challenges and (some) solutions. *Egems*, 1(3), 2013.

[118] Joshua D Angrist and Alan B Krueger. Instrumental variables and the search for identification: From supply and demand to natural experiments. *Journal of Economic perspectives*, 15(4):69–85, 2001.

[119] James J Heckman and Richard Robb Jr. Alternative methods for evaluating the impact of interventions: An overview. *Journal of econometrics*, 30(1-2):239–267, 1985.

[120] David Card. Using geographic variation in college proximity to estimate the return to schooling. Technical report, National Bureau of Economic Research, 1993.

[121] Harry H Kelejian. Two-stage least squares and econometric systems linear in parameters but nonlinear in the endogenous variables. *Journal of the American Statistical Association*, 66(334):373–374, 1971.

[122] Takeshi Amemiya. The nonlinear two-stage least-squares estimator. *Journal of econometrics*, 2(2):105–110, 1974.

[123] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, 91(434):444–455, 1996.

[124] Jeffrey M Wooldridge. Control function methods in applied econometrics. *Journal of Human Resources*, 50(2):420–445, 2015.

[125] Manuel Wiesenfarth, Carlos Matías Hisgen, Thomas Kneib, and Carmen Cadarso-Suarez. Bayesian nonparametric instrumental variables regression based on penalized splines and dirichlet process mixtures. *Journal of Business & Economic Statistics*, 32(3):468–482, 2014.

[126] Serge Darolles, Yanqin Fan, Jean-Pierre Florens, and Eric Renault. Nonparametric instrumental regression. *Econometrica*, 79(5):1541–1565, 2011.

[127] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[128] Rajeev H Dehejia and Sadek Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161, 2002.

[129] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.

[130] Michele Jonsson Funk, Daniel Westreich, Chris Wiesen, Til Stürmer, M Alan Brookhart, and Marie Davidian. Doubly robust estimation of causal effects. *American journal of epidemiology*, 173(7):761–767, 2011.

[131] Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Deep iv: A flexible approach for counterfactual prediction. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1414–1423. JMLR. org, 2017.

[132] Andrew Bennett, Nathan Kallus, and Tobias Schnabel. Deep generalized method of moments for instrumental variable analysis. *arXiv preprint arXiv:1905.12495*, 2019.

[133] Abraham Wald. The fitting of straight lines if both variables are subject to error. *The Annals of Mathematical Statistics*, 11(3):284–300, 1940.

[134] Zijian Guo and Dylan S Small. Control function instrumental variable estimation of nonlinear causal effect models. *The Journal of Machine Learning Research*, 17(1):3448–3482, 2016.

[135] Rahul Singh, Maneesh Sahani, and Arthur Gretton. Kernel instrumental variable regression. *arXiv preprint arXiv:1906.00232*, 2019.

[136] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054, 1982.

[137] Alexander Balke and Judea Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1171–1176, 1997.

[138] Charles F Manski. Nonparametric bounds on treatment effects. *The American Economic Review*, 80(2):319–323, 1990.

[139] Whitney K Newey. Nonparametric instrumental variables estimation. *American Economic Review*, 103(3):550–56, 2013.

[140] Denis Chetverikov and Daniel Wilhelm. Nonparametric instrumental variable estimation under monotonicity. *Econometrica*, 85(4):1303–1320, 2017.

[141] Flavio Cunha, James J Heckman, and Susanne M Schennach. Estimating the technology of cognitive and noncognitive skill formation. *Econometrica*, 78(3):883–931, 2010.

[142] Andrew Chesher. Identification in nonseparable models. *Econometrica*, 71(5):1405–1441, 2003.

[143] Guido W Imbens and Whitney K Newey. Identification and estimation of triangular simultaneous equations models without additivity. *Econometrica*, 77(5):1481–1512, 2009.

[144] Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[145] William Astle, David J Balding, et al. Population structure and cryptic relatedness in genetic association studies. *Statistical Science*, 24(4):451–471, 2009.

[146] Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. *Nature methods*, 8(10):833, 2011.

[147] Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick,

and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904, 2006.

[148] Jianming Yu, Gael Pressoir, William H Briggs, Irie Vroh Bi, Masanori Yamasaki, John F Doebley, Michael D McMullen, Brandon S Gaut, Dahlia M Nielsen, James B Holland, et al. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature genetics*, 38(2):203, 2006.

[149] Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, and Jian Yang. 10 years of gwas discovery: biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017.

[150] Rajesh Ranganath and Adler Perotte. Multiple causal inference with latent confounding. *arXiv preprint arXiv:1805.08273*, 2018.

[151] Yixin Wang and David M Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, (just-accepted):1–71, 2019.

[152] Miguel A Hernán and James M Robins. Causal inference: what if. *Boca Raton: Chapman & Hill/CRC*, 2020, 2020.

[153] Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.

[154] Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

[155] J. Correa and E. Bareinboim. A calculus for stochastic interventions: Causal effect identification and surrogate experiments. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020. AAAI Press.

[156] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[157] Marc Ratkovic. Balancing within the margin: Causal effect estimation with support vector machines. *Department of Politics, Princeton University, Princeton, NJ*, 2014.

[158] Jennifer L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011. doi: 10.1198/jcgs.2010.08162. URL https://doi.org/10.1198/jcgs.2010.08162.

[159] James M Robins. Robust estimation in sequentially ignorable missing data and causal inference models. In *Proceedings of the American Statistical Association*, volume 1999, pages 6–10. Indianapolis, IN, 2000.

[160] Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998.

[161] Wellcome Trust Case Control Consortium et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661, 2007.

[162] Patrick CA Dubois, Gosia Trynka, Lude Franke, Karen A Hunt, Jihane Romanos, Alessandra Curtotti, Alexandra Zhernakova, Graham AR Heap, Róza Ádány, Arpo Aromaa, et al. Multiple common variants for celiac disease influencing immune gene expression. *Nature genetics*, 42(4):295, 2010.

[163] Ludvig M Sollid. Coeliac disease: dissecting a complex inflammatory disorder. *Nature Reviews Immunology*, 2(9):647, 2002.

[164] Karen A Hunt, Alexandra Zhernakova, Graham Turner, Graham AR Heap, Lude Franke, Marcel Bruinenberg, Jihane Romanos, Lotte C Dinesen, Anthony W Ryan, Davinder Pane-

sar, et al. Novel celiac disease genetic determinants related to the immune response. *Nature genetics*, 40(4):395, 2008.

[165] Svetlana Adamovic, SS Amundsen, BA Lie, AH Gudjonsdottir, H Ascher, J Ek, DA Van Heel, S Nilsson, LM Sollid, and Å Torinsson Naluai. Association study of il2/il21 and fcgriia: significant association with the il2/il21 region in scandinavian coeliac disease families. *Genes and immunity*, 9(4):364, 2008.

[166] Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience*, 4(1):s13742–015, 2015.

[167] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct): 2825–2830, 2011.

[168] Lucia A Hindorff, Praveen Sethupathy, Heather A Junkins, Erin M Ramos, Jayashri P Mehta, Francis S Collins, and Teri A Manolio. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proceedings of the National Academy of Sciences*, 106(23):9362–9367, 2009.

[169] Carl A Anderson, Gabrielle Boucher, Charlie W Lees, Andre Franke, Mauro D'Amato, Kent D Taylor, James C Lee, Philippe Goyette, Marcin Imielinski, Anna Latiano, et al. Meta-analysis identifies 29 additional ulcerative colitis risk loci, increasing the number of confirmed associations to 47. *Nature genetics*, 43(3):246, 2011.

[170] Virginia Pascual, Romina Dieli-Crimi, Natalia López-Palacios, Andrés Bodas, Luz María Medrano, and Concepción Núñez. Inflammatory bowel disease and celiac disease: overlaps and differences. *World journal of gastroenterology: WJG*, 20(17):4846, 2014.

[171] World Health Organization et al. *Prevention of cardiovascular disease: guidelines for assessment and management of total cardiovascular risk.* World Health Organization, 2007.

[172] Donna K Arnett, Roger S Blumenthal, Michelle A Albert, Andrew B Buroker, Zachary D Goldberger, Ellen J Hahn, Cheryl Dennison Himmelfarb, Amit Khera, Donald Lloyd-Jones, J William McEvoy, et al. 2019 acc/aha guideline on the primary prevention of cardiovascular disease: a report of the american college of cardiology/american heart association task force on clinical practice guidelines. *Circulation*, 140(11):e596–e646, 2019.

[173] Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

[174] Ralph B D'Agostino Sr, Ramachandran S Vasan, Michael J Pencina, Philip A Wolf, Mark Cobain, Joseph M Massaro, and William B Kannel. General cardiovascular risk profile for use in primary care: the framingham heart study. *Circulation*, 117(6):743–753, 2008.

[175] Peter WF Wilson, Ralph B D'Agostino, Daniel Levy, Albert M Belanger, Halit Silbershatz, and William B Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

[176] David C Goff Jr, Donald M Lloyd-Jones, Glen Bennett, Sean Coady, Ralph B Dagostino, Raymond Gibbons, Philip Greenland, Daniel T Lackland, Daniel Levy, Christopher J Odonnell, et al. 2013 acc/aha guideline on the assessment of cardiovascular risk: a report of the american college of cardiology/american heart association task force on practice guidelines. *Circulation*, 129(25_suppl_2):S49–S73, 2014.

[177] P. W. Wilson, R. B. D'Agostino, D. Levy, A. M. Belanger, H. Silbershatz, and W. B. Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998. ISSN 0009-7322. doi: 10.1161/01.cir.97.18.1837.

[178] Christine Wallisch, Georg Heinze, Christoph Rinner, Gerald Mundigler, Wolfgang C Winkelmayer, and Daniela Dunkler. External validation of two framingham cardiovascular risk equations and the pooled cohort equations: a nationwide registry analysis. *International journal of cardiology*, 283:165–170, 2019.

[179] Benjamin S Wessler, Jason Nelson, Jinny G Park, Hannah McGinnes, Gaurav Gulati, Riley Brazil, Ben Van Calster, David van Klaveren, Esmee Venema, Ewout Steyerberg, et al. External validations of cardiovascular clinical prediction models: a large-scale review of the literature. *Circulation: Cardiovascular Quality and Outcomes*, 14(8):e007858, 2021.

[180] Gaurav Gulati, Jenica Upshaw, Benjamin S Wessler, Riley J Brazil, Jason Nelson, David van Klaveren, Christine M Lundquist, Jinny G Park, Hannah McGinnes, Ewout W Steyerberg, et al. Generalizability of cardiovascular disease clinical prediction models: 158 independent external validations of 104 unique models. *Circulation: Cardiovascular Quality and Outcomes*, 15(4):e008487, 2022.

[181] Sadiya S Khan, Josef Coresh, Michael J Pencina, Chiadi E Ndumele, Janani Rangaswami, Sheryl L Chow, Latha P Palaniappan, Laurence S Sperling, Salim S Virani, Jennifer E Ho, et al. Novel prediction equations for absolute risk assessment of total cardiovascular disease incorporating cardiovascular-kidney-metabolic health: a scientific statement from the american heart association. *Circulation*, 2023.

[182] Julia Hippisley-Cox, Carol Coupland, and Peter Brindle. Development and validation of qrisk3 risk prediction algorithms to estimate future risk of cardiovascular disease: prospective cohort study. *bmj*, 357, 2017.

[183] Stephen D Persell, Alexis P Dunne, Donald M Lloyd-Jones, and David W Baker. Electronic health record-based cardiac risk assessment and identification of unmet preventive needs. *Medical care*, pages 418–424, 2009.

[184] Sheng-Feng Sung, Kuan-Lin Sung, Ru-Chiou Pan, Pei-Ju Lee, and Ya-Han Hu. Automated risk assessment of newly detected atrial fibrillation poststroke from electronic health record data using machine learning and natural language processing. *Frontiers in Cardiovascular Medicine*, 9, 2022.

[185] Edward H Kennedy, Wyndy L Wiitala, Rodney A Hayward, and Jeremy B Sussman. Improved cardiovascular risk prediction using nonparametric regression and electronic health record data. *Medical care*, 51(3):251, 2013.

[186] Qianyu Yuan, Tianrun Cai, Chuan Hong, Mulong Du, Bruce E Johnson, Michael Lanuti, Tianxi Cai, and David C Christiani. Performance of a machine learning algorithm using electronic health record data to identify and estimate survival in a longitudinal cohort of patients with lung cancer. *JAMA Network Open*, 4(7):e2114723–e2114723, 2021.

[187] Yolanda Hagar, David Albers, Rimma Pivovarov, Herbert Chase, Vanja Dukic, and Noémie Elhadad. Survival analysis with electronic health record data: Experiments with chronic kidney disease. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(5): 385–403, 2014.

[188] Julia Hippisley-Cox, Carol Coupland, Yana Vinogradova, John Robson, Rubin Minhas, Aziz Sheikh, and Peter Brindle. Predicting cardiovascular risk in england and wales: prospective derivation and validation of qrisk2. *Bmj*, 336(7659):1475–1482, 2008.

[189] Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. In *Machine Learning for Healthcare Conference*, pages 101–114. PMLR, 2016.

[190] Xenia Miscouridou, Adler Perotte, Noémie Elhadad, and Rajesh Ranganath. Deep survival analysis: Nonparametrics and missingness. In *Machine Learning for Healthcare Conference*, pages 244–256. PMLR, 2018.

[191] Xing Song, Alan SL Yu, John A Kellum, Lemuel R Waitman, Michael E Matheny, Steven Q

Simpson, Yong Hu, and Mei Liu. Cross-site transportability of an explainable artificial intelligence model for acute kidney injury prediction. *Nature communications*, 11(1):5668, 2020.

[192] Erkin Otles, Jeeheh Oh, Benjamin Li, Michelle Bochinski, Hyeon Joo, Justin Ortwine, Erica Shenoy, Laraine Washer, Vincent B Young, Krishna Rao, et al. Mind the performance gap: examining dataset shift during prospective validation. In *Machine Learning for Healthcare Conference*, pages 506–534. PMLR, 2021.

[193] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L Beam, Irene Y Chen, and Rajesh Ranganath. A review of challenges and opportunities in machine learning for health. *AMIA Summits on Translational Science Proceedings*, 2020:191, 2020.

[194] Sharon E Davis, Michael E Matheny, Suresh Balu, and Mark P Sendak. A framework for understanding label leakage in machine learning for health care. *Journal of the American Medical Informatics Association*, 31(1):274–280, 2024.

[195] Fabrizio D'Ascenzo, Ovidio De Filippo, Guglielmo Gallone, Gianluca Mittone, Marco Agostino Deriu, Mario Iannaccone, Albert Ariza-Solé, Christoph Liebetrau, Sergio Manzano-Fernández, Giorgio Quadri, et al. Machine learning-based prediction of adverse events following an acute coronary syndrome (praise): a modelling study of pooled datasets. *The Lancet*, 397(10270):199–207, 2021.

[196] Chenxi Huang, Shu-Xia Li, César Caraballo, Frederick A Masoudi, John S Rumsfeld, John A Spertus, Sharon-Lise T Normand, Bobak J Mortazavi, and Harlan M Krumholz. Performance metrics for the comparative analysis of clinical risk prediction models employing machine learning. *Circulation: Cardiovascular Quality and Outcomes*, 14(10):e007526, 2021.

[197] Sadiya S Khan, Kunihiro Matsushita, Yingying Sang, Shoshana H Ballew, Morgan E Grams, Aditya Surapaneni, Michael J Blaha, April P Carson, Alexander R Chang, Elizabeth

Ciemins, et al. Development and validation of the american heart association's prevent equations. *Circulation*, 149(6):430–449, 2024.

[198] Lily Zhang, Veronica Tozzo, John Higgins, and Rajesh Ranganath. Set norm and equivariant skip connections: Putting the deep in deep sets. In *International Conference on Machine Learning*, pages 26559–26574. PMLR, 2022.

[199] J Weston Hughes, James Tooley, Jessica Torres Soto, Anna Ostropolets, Tim Poterucha, Matthew Kai Christensen, Neal Yuan, Ben Ehlert, Dhamanpreet Kaur, Guson Kang, et al. A deep learning-based electrocardiogram risk score for long term cardiovascular death and disease. *npj Digital Medicine*, 6(1):169, 2023.

[200] Rajesh Ranganath, Adler Perotte, Noemie Elhadad, and David Blei. Deep survival analysis. pages 101–114, 2016. URL https://arxiv.org/abs/1608.02158v2.

[201] Mindy M Pike, Paul A Decker, Nicholas B Larson, Jennifer L St. Sauver, Paul Y Takahashi, Véronique L Roger, Walter A Rocca, Virginia M Miller, Janet E Olson, Jyotishman Pathak, et al. Improvement in cardiovascular risk prediction with electronic health records. *Journal of cardiovascular translational research*, 9:214–222, 2016.

[202] Rine Nakanishi, Piotr J Slomka, Richard Rios, Julian Betancur, Michael J Blaha, Khurram Nasir, Michael D Miedema, John A Rumberger, Heidi Gransar, Leslee J Shaw, et al. Machine learning adds to clinical and cac assessments in predicting 10-year chd and cvd deaths. *Cardiovascular Imaging*, 14(3):615–625, 2021.

[203] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024. doi: 10.1109/TNNLS.2022.3229161.

[204] Yuxuan Hu, Albert Lui, Mark Goldstein, Mukund Sudarshan, Andrea Tinsay, Cindy Tsui,

Samuel D Maidman, John Medamana, Neil Jethani, Aahlad Puli, et al. Development and external validation of a dynamic risk score for early prediction of cardiogenic shock in cardiac intensive care units using machine learning. *European Heart Journal: Acute Cardiovascular Care*, page zuae037, 2024.

[205] Seema Pursnani and Maqdooda Merchant. South asian ethnicity as a risk factor for coronary heart disease. *Atherosclerosis*, 315:126–130, 2020.

[206] Mingkai Peng, Cathy Eastwood, Alicia Boxill, Rachel Joy Jolley, Laura Rutherford, Karen Carlson, Stafford Dean, and Hude Quan. Coding reliability and agreement of international classification of disease, 10th revision (icd-10) codes in emergency department data. *International Journal of Population Data Science*, 3(1), 2018.

[207] Mark Woodward, Peter Brindle, and Hugh Tunstall-Pedoe. Adding social deprivation and family history to cardiovascular risk assessment: the assign score from the scottish heart health extended cohort (shhec). *Heart*, 93(2):172–176, 2007.

[208] Andrea Rotnitzky, Quanhong Lei, Mariela Sued, and James M Robins. Improved double-robust estimation in missing data and causal inference models. *Biometrika*, 99(2):439–456, 2012.

[209] Angela Zhang, Lei Xing, James Zou, and Joseph C Wu. Shifting machine learning for healthcare from development to deployment and from models to data. *Nature Biomedical Engineering*, 6(12):1330–1345, 2022.

[210] Benjamin Smith, Senne Van Steelandt, and Anahita Khojandi. Evaluating the impact of health care data completeness for deep generative models. *Methods of Information in Medicine*, 62(01/02):031–039, 2023.

[211] Yook Chin Chia, Sarah Yu Weng Gray, Siew Mooi Ching, Hooi Min Lim, and Karuthan Chinna. Validation of the framingham general cardiovascular risk score in a multiethnic

asian population: a retrospective cohort study. 5(5):e007324. Publisher: British Medical Journal Publishing Group.

[212] Ramachandran S Vasan and Edwin van den Heuvel. Differences in estimates for 10-year risk of cardiovascular disease in black versus white individuals with identical risk factor profiles using pooled cohort equations: an in silico cohort study. *The Lancet Digital Health*, 4(1):e55–e63, 2022.

[213] CDC. National center for health statistics. paradata file description. *Accessed February*, 2, 2023.

[214] Thomas A Gerds and Martin Schumacher. Consistent estimation of the expected brier score in general survival models with right-censored event times. *Biometrical Journal*, 48 (6):1029–1040, 2006.

[215] Håvard Kvamme and Ørnulf Borgan. The brier score under administrative censoring: Problems and solutions. *arXiv preprint arXiv:1912.08581*, 2019.

[216] Xintian Han, Mark Goldstein, Aahlad Puli, Thomas Wies, Adler Perotte, and Rajesh Ranganath. Inverse-weighted survival games. *Advances in neural information processing systems*, 34:2160–2172, 2021.

[217] Paul E. Stang, Patrick B. Ryan, Judith A. Racoosin, J. Marc Overhage, Abraham G. Hartzema, Christian Reich, Emily Welebob, Thomas Scarnecchia, and Janet Woodcock. Advancing the science for active surveillance: Rationale and design for the observational medical outcomes partnership. 153(9):600–606. ISSN 0003-4819. doi: 10.7326/0003-4819-153-9-201011020-00010. URL https://www.acpjournals.org/doi/full/10.7326/0003-4819-153-9-201011020-00010. Publisher: American College of Physicians.

[218] Jeffrey S Brown, Michael Kahn, and Sengwee Toh. Data quality assessment for comparative effectiveness research in distributed data networks. *Medical care*, 51:S22–S29, 2013.

[219] Nicole Gray Weiskopf and Chunhua Weng. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association*, 20(1):144–151, 2013.

[220] Michael G Kahn, Tiffany J Callahan, Juliana Barnard, Alan E Bauck, Jeff Brown, Bruce N Davidson, Hossein Estiri, Carsten Goerg, Erin Holve, Steven G Johnson, et al. A harmonized data quality assessment terminology and framework for the secondary use of electronic health record data. *Egems*, 4(1), 2016.

[221] Hossein Estiri, Jeffrey G Klann, and Shawn N Murphy. A clustering approach for detecting implausible observation values in electronic health records data. *BMC medical informatics and decision making*, 19:1–16, 2019.

[222] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[223] Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. *Advances in neural information processing systems*, 24, 2011.

[224] Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[225] Per K Andersen, Ornulf Borgan, Richard D Gill, and Niels Keiding. *Statistical models based on counting processes.* Springer Science & Business Media, 2012.

[226] Donald M Lloyd-Jones. Cardiovascular risk prediction: basic concepts, current status, and future directions. *Circulation*, 121(15):1768–1777, 2010.

[227] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay

Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, et al. Wilds: A benchmark of in-the-wild distribution shifts 2021. *arXiv preprint arXiv:2012.07421*, 2020.

[228] Qingyao Sun, Kevin Murphy, Sayna Ebrahimi, and Alexander D'Amour. Beyond invariance: Test-time label-shift adaptation for distributions with" spurious" correlations. *arXiv preprint arXiv:2211.15646*, 2022.

[229] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[230] David Brandfonbrener, Hanlin Zhang, Andreas Kirsch, Jonathan Richard Schwarz, and Sham Kakade. Color-filter: Conditional loss reduction filtering for targeted language model pre-training. *arXiv preprint arXiv:2406.10670*, 2024.

[231] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

[232] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

[233] Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa. Domain generalization with optimal transport and metric learning. *arXiv preprint arXiv:2007.10573*, 2020.

[234] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[235] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of repre-

sentations for domain adaptation. *Advances in neural information processing systems*, 19: 137, 2007.

[236] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in Data Science and Information Engineering*, pages 877–894, 2021.

[237] Zhao Wang and Aron Culotta. Robustness to spurious correlations in text classification via automatically generated counterfactuals. *arXiv preprint arXiv:2012.10040*, 2020.

[238] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[239] Yongqiang Chen, Kaiwen Zhou, Yatao Bian, Binghui Xie, Bingzhe Wu, Yonggang Zhang, MA KAILI, Han Yang, Peilin Zhao, Bo Han, et al. Pareto invariant risk minimization: Towards mitigating the optimization dilemma in out-of-distribution generalization. In *The Eleventh International Conference on Learning Representations*, 2022.

[240] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, pages 26397–26411. PMLR, 2022.

[241] Yongqiang Chen, Wei Huang, Kaiwen Zhou, Yatao Bian, Bo Han, and James Cheng. Understanding and improving feature learning for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

[242] Aahlad Manas Puli, Lily Zhang, Yoav Wald, and Rajesh Ranganath. Don't blame dataset shift! shortcut learning due to gradients and cross entropy. *Advances in Neural Information Processing Systems*, 36:71874–71910, 2023.

[243] LIN Yong, Lu Tan, HAO Yifan, Ho Nam Wong, Hanze Dong, WEIZHONG ZHANG, Yu-

jiu Yang, and Tong Zhang. Spurious feature diversification improves out-of-distribution generalization. In *The Twelfth International Conference on Learning Representations*, 2023.

[244] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

[245] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

[246] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[247] Xinlong Feng and Zhinan Zhang. The rank of a random matrix. *Applied mathematics and computation*, 185(1):689–694, 2007.

[248] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[249] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

[250] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[251] Felix V Agakov and David Barber. An auxiliary variational method. In *International Conference on Neural Information Processing*, pages 561–566. Springer, 2004.

[252] Tim Salimans, Diederik P Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. *arXiv preprint arXiv:1410.6460*, 2014.

[253] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[254] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

[255] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[256] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[257] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[258] Jonathan Niles-Weed and Philippe Rigollet. Estimation of wasserstein distances in the spiked transport model. *arXiv preprint arXiv:1909.07513*, 2019.

[259] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[260] Michael Spivak. *Calculus on manifolds: a modern approach to classical theorems of advanced calculus*. CRC press, 2018.

[261] Gerald Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

[262] Morris W Hirsch, Robert L Devaney, and Stephen Smale. *Differential equations, dynamical systems, and linear algebra*, volume 60. Academic press, 1974.

[263] Timothy Thornton and Michael Wu. Summer institute in statistical genetics 2015.

[264] James M Robins et al. Information recovery and bias adjustment in proportional hazards regression analysis of randomized trials using surrogate markers. In *Proceedings of the*

biopharmaceutical section, American statistical association, volume 24, page 3. San Francisco CA, 1993.

[265] Gaohong Dong, Lu Mao, Bo Huang, Margaret Gamalo-Siebers, Jiuzhou Wang, GuangLei Yu, and David C Hoaglin. The inverse-probability-of-censoring weighting (ipcw) adjusted win ratio statistic: An unbiased estimator in the presence of independent censoring. *Journal of biopharmaceutical statistics*, 30(5):882–899, 2020.

[266] Thomas A Gerds, Michael W Kattan, Martin Schumacher, and Changhong Yu. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in medicine*, 32(13):2173–2184, 2013.

[267] Samir Khan and Johan Ugander. Adaptive normalization for ipw estimation. *Journal of Causal Inference*, 11(1):20220019, 2023.