PREDICTIVE MODELING IN THE PRESENCE OF NUISANCE-INDUCED SPURIOUS CORRELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

In many prediction problems, spurious correlations 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 the 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 first define the *nuisance-varying family*, a set of distributions that differ only in the nuisance-label relationship. We then introduce the nuisance-randomized distribution, a distribution where the nuisance and 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 within the set on every distribution in the nuisance-varying family. 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.

1 INTRODUCTION

Spurious correlations are relationships between the label and the covariates prone to change between training and test distributions (Gulrajani and Lopez-Paz, 2020). Predictive models that exploit spurious correlations perform worse than even predicting without covariates on the test distribution (Arjovsky et al., 2019). Detecting spurious correlations requires more than the training distribution because any fixed dataset has a fixed label-covariate relationship. Often, spurious correlations are discovered by noticing different relationships across multiple distributions between the label and 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 due to natural habitats (Arjovsky et al., 2019; Geirhos et al., 2020), but photos of these animals can be taken outside their habitats. In classifying hair color from celebrity faces on CelebA (Liu et al., 2015), gender is correlated with the hair color. This relationship may not hold in different countries (Sagawa et al., 2020). 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 (Wang and Culotta, 2020a). 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 (Zech et al., 2018). Such factors are rarely recorded in datasets but produce subtle differences in X-ray images that convolutional networks easily learn (Badgeley et al., 2019).

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 may 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. We develop NURD to use data collected under one nuisance-label

relationship to build predictive models that do well on every member of the family regardless of the nuisance-label relationship in that member. Before presenting NURD, we introduce the theoretical elements that help guarantee performance on every member of the family in section 2. We define the distribution where nuisance and label are independent, called the nuisance-randomized distribution. We show that the distribution of the label given the covariates under the nuisance-randomized distribution is minimax optimal for the nuisance-varying family, if the label and the nuisance are independent given the covariates.

The assumption that conditioning on covariates does not correlate the label and the nuisance does not hold in all nuisance-varying families. Instead, we focus on a set of *representations* of the covariates that help build good predictors for all members of the family. This set of representations, where conditioning on any member does not correlate the label and the nuisance, is called the *uncorrelating set*; we refer to its members as *uncorrelating representations.* Under the nuisance-randomized distribution, the conditional distribution of the label given a representation in the uncorrelating set only depends on properties that are shared across all members of the nuisance-varying family. This fact helps guarantee such a representation performs at least as well as predicting without covariates on any member of the nuisance varying family. Within the uncorrelating set, we show a representation achieves the highest performance on every member of the nuisance-varying family if conditioning on this representation and the nuisance renders the label independent of every uncorrelating representation. Further, we show that this performance can be realized by uncorrelating representations that are most informative of the label under the nuisance-randomized distribution.

To find the representation with the highest performance by maximizing the information between an uncorrelating representation and the label under the nuisance-randomized distribution, we develop Nuisance-Randomized Distillation (NURD). 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*, learns an uncorrelating representation with the maximum information with the label on the nuisance-randomized data. In deriving the distillation step, we show that optimization over the entire uncorrelating set by penalizing conditional information can have many local optima that are bad predictors. Distillation, instead, focuses on a subset of the uncorrelated set by penalizing the joint information that the label and representation have with the nuisance. We evaluate NURD on class-conditional Gaussians, labeling colored MNIST images (Arjovsky et al., 2019), detecting waterbirds, 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 NUISANCE-RANDOMIZING AND UNCORRELATING SETS

We develop nuisance-induced spurious correlations through a family of data generating processes. Let $\mathbf{y}$ be the label, $\mathbf{z}$ be the nuisance, and $\mathbf{x}$ be the covariates. 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 form

$$p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y}) p_D(\mathbf{z} \mid \mathbf{y}) p(\mathbf{x} \mid \mathbf{z}, \mathbf{y}), \tag{1}$$

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 $S_{\mathcal{F}}$. 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, called the training distribution, can perform worse than predicting without covariates on another member of the family, called a test distribution. We define performance of a predictor $\hat{p}(\mathbf{y} \mid \mathbf{x})$ on a distribution p_{te} as the negative expected $\mathbf{KL}$ 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. As an example, consider a family $\{q_a\}_{a \in \mathbb{R}}$:

$$\mathbf{y} \sim \mathcal{N}(0, 1) \quad \mathbf{z} \sim \mathcal{N}(a\mathbf{y}, 0.5) \quad \mathbf{x} = \left[\mathbf{x}_1 \sim \mathcal{N}(\mathbf{y} - \mathbf{z}, 1.5), \mathbf{x}_2 \sim \mathcal{N}(\mathbf{y} + \mathbf{z}, 0.5)\right]. \tag{2}$$

Given the training distribution is $p_{tr} = q_1$ and a test distribution is $p_{te} = q_{-1}$, $p_{tr}(\mathbf{y} \mid \mathbf{x})$ 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.8 for the proof. For $p_{te}, p_{tr} \in \mathcal{F}$, $p_{tr}(\mathbf{y}) = p_{te}(\mathbf{y}) = p(\mathbf{y})$, so predicting without covariates only requires estimating the marginal distribution $p_{tr}(\mathbf{y})$. When the changing nuisance-label relationship makes the conditional $p_D(\mathbf{y} \mid \mathbf{x})$ of one member unsuitable for other members $p'_D \in \mathcal{F}$ the family exhibits *nuisance-induced spurious correlations*.

The training conditional $p_{tr}(\mathbf{y} \mid \mathbf{x})$ predicts worse than the marginal $p_{tr}(\mathbf{y})$ because it exploits the nuisance-label relationship in the training distribution, which can be different in the test distribution p_{te}. If the nuisance and label are independent, the nuisance does not predict the label by itself. Such a distribution, denoted by $p_{\perp\!\!\!\perp}$, is defined as follows:

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

By construction in the nuisance-randomized distribution, the label is independent of the nuisance. The hope in constructing the nuisance-randomized distribution $p_{\perp\!\!\!\perp}$ is that since the nuisance does not predict the label by itself, the conditional distribution $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ should be a good predictor for other test distributions in the family $\mathcal{F}$. This hope bears fruit; when conditioning on the covariates does not make the label and nuisance dependent in the nuisance-randomized distribution, and the nuisance-varying family has a diverse set of nuisance-label relationships, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ is minimax optimal. Let $\perp\!\!\!\perp_q$ to denote independence under a distribution q.

Proposition 1. *Consider a nuisance-varying family $\mathcal{F}$ (eq. (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*

$$\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z} \implies \exists p'_D \in \mathcal{F} s.t. \left[\mathbb{E}_{p'_D(\mathbf{x})} KL\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y}) \right] > 0. \tag{3}$$

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 :

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

The information criterion in eq. (3) ensures that the nuisance-label relationship $p_D(\mathbf{z} \mid \mathbf{y})$ changes enough within the family that any reliance on $\mathbf{z}$ degrades performance under some test distribution. The information criterion eq. (3) holds in common families of distributions like jointly Gaussian families. See appendix A.4 for proposition 1's proof and see appendix A.7 for the jointly Gaussian example. If the family is not sufficiently diverse in that eq. (3) does not hold, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ may not be minimax optimal. For example, if $\mathcal{F} = \{p_D, p_{\perp\!\!\!\perp}\}$, then $p_D(\mathbf{y} \mid \mathbf{x})$ can be minimax optimal; using the data generating process for $q_a(\mathbf{y}, \mathbf{z}, \mathbf{x})$ in appendix A.7, this holds for $\mathcal{F} = \{p_D = q_2, p_{\perp\!\!\!\perp} = q_0\}$.

Uncorrelating sets of representations. The minimax optimality relies on the assumption that $\mathbf{z}$ does not provide information about $\mathbf{y}$ after having conditioned in $\mathbf{x}$, i.e. $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid \mathbf{x}$. This condition does not hold in all nuisance-varying families. Consider the example in eq. (2). Conditioning on $\mathbf{x}$ makes the label and nuisance dependent. However, there exists a function $r(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2 = \mathbf{y} + noise$ that, being a pure function of the label and noise from $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$, would have the same statistical dependence with the label in any member of $\mathcal{F}$. Since the dependence does not change, intuitively, the predictor $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ should perform well on any member of the nuisance-varying family. On the other hand, the predictor $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ is worse than predicting without covariates under arbitrary test distributions without further assumptions. See appendix A.8 for details. These functions do worse because when the label depends on the nuisance given a function of the covariates, properties of the nuisance that appear in this function become predictive of the label. When a predictor uses properties of the nuisance, the performance across members of a nuisance-varying family may change arbitrarily. This insight leads to the definition of a set of such *uncorrelating* representations we call the *uncorrelating set*:

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

The uncorrelating set consists of representations of the covariates $r(\mathbf{x})$ such that knowing the nuisance $\mathbf{z}$ given the representation does not improve the prediction of the label $\mathbf{y}$. Under the nuisance-randomized distribution, the conditional distribution of the label given any representation in the uncorrelating set only depends on the 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 $\mathcal{F}$. Let $\mathbf{z}'$ be such that $p_{\perp\!\!\!\perp}(\mathbf{z}' \mid r(\mathbf{x})) > 0$,

$$p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x})) = p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}') = \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{z}') p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}')}{p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z}')} = \frac{p(\mathbf{y}) p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}')}{\mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}')}. \tag{4}$$

[1] 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 Appendix A.2.

This fact allows the performance of $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ to be characterized for any member within the family and helps show that such conditionals do at least as well as predicting without covariates for any test distribution:

Lemma 1. *Let $\mathcal{F}$ be a nuisance-varying family (eq. (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*

$$\texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) = \texttt{Perf}_{p_{te}}(p(\mathbf{y})) + \mathop{\mathbb{E}}_{p_{te}(\mathbf{y},\mathbf{z})} KL\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right]. \quad (5)$$

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

See appendix A.2 for the proof. Lemma 1 lower bounds the performance of the conditional distribution under $p_{\perp\!\!\!\perp}$ of the label given any representation in the uncorrelating set on *any* $p_{te} \in \mathcal{F}$. However, it does not specify which of these representations leads to the best conditional distribution. Next, we characterize representations in the uncorrelating set that are *simultaneously* optimal for all test distributions $p_{te} \in \mathcal{F}$.

Optimal uncorrelating representations. 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 representations $r, r_2 \in \mathcal{R}(p_{\perp\!\!\!\perp})$. How can $r_2(\mathbf{x})$ dominate $r(\mathbf{x})$ in performance on every test distribution in the nuisance-varying family? Equation (4) shows that for any $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$, the conditional $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ relies only on those properties that are shared across all the distributions in the nuisance-varying family; i.e. for any $\mathbf{z} \sim p_{\perp\!\!\!\perp}(\mathbf{z} \mid r(\mathbf{x}))$, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x})) \propto p(\mathbf{y})p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z} = \mathbf{z})$. Intuitively, if $r_2(\mathbf{x})$ blocks this dependence between the label and $r(\mathbf{x})$, i.e. $p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r_2(\mathbf{x})) = p(r(\mathbf{x}) \mid \mathbf{z}, r_2(\mathbf{x}))$, then $r_2(\mathbf{x})$ should perform as well as or better than $r(\mathbf{x})$ on any test distribution. Theorem 1 confirms this intuition, and shows optimality of the *maximally blocking* uncorrelating representation

Theorem 1. *Consider a pair $r, r_2 \in \mathcal{R}(p_{\perp\!\!\!\perp})$ such that r_2 **blocks** r: $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r_2(\mathbf{x})$. Then r_2 dominates r in performance on every test distribution in $\mathcal{F}$:*

$$\forall p_{te} \in \mathcal{F}, \qquad \texttt{Perf}_{p_{te}}(r_2(\mathbf{x}))\,) \geq \texttt{Perf}_{p_{te}}(r(\mathbf{x}))).$$

*Let $\exists r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})$ that is **maximally blocking**: $\forall r \in \mathcal{R}(p_{\perp\!\!\!\perp})$, $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r^*(\mathbf{x})$. Then, $r^*(\mathbf{x})$ is **optimal** in that $r^*(\mathbf{x})$ dominates every uncorrelating $r(\mathbf{x})$ in performance on every $p_{te} \in \mathcal{F}$:*

$$\forall p_{te} \in \mathcal{F}, \forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}) \qquad \texttt{Perf}_{p_{te}}(r^*(\mathbf{x}))\,) \geq \texttt{Perf}_{p_{te}}(r(\mathbf{x}))).$$

See appendix A.3 for the proof. Next, we develop an algorithm to find optimal uncorrelating representations that help guarantee the highest performance (within the uncorrelating set) on any test distribution in the family $\mathcal{F}$.

3 Nuisance-Randomized Distillation (NuRD)

We start off by showing the optimal representation in theorem 1 has maximum information with the label under the nuisance-randomized distribution and representations in $\mathcal{R}(p_{\perp\!\!\!\perp})$ with the same information as the optimal representation are also optimal.

Theorem 2. *Assume $r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})$ is maximally blocking, then 1) for all $r'(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$, $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) \geq \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r'(\mathbf{x}))$, 2) for $r'(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$ such that $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r'(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x}))$, r' matches the performance of r^*, $\texttt{Perf}_{p_{te}}(r^*(\mathbf{x})) = \texttt{Perf}_{p_{te}}(r'(\mathbf{x}))$, and is optimal.*

See appendix A.3 for the proof. Theorem 2 says a maximal information representation under the nuisance-randomized distribution is best. We develop a representation learning algorithm to maximize the mutual information between the label and the 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 builds 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 $\mathbf{z} \sim p_{tr}(\mathbf{z}) \quad \mathbf{y} \sim p(\mathbf{y}) \quad \mathbf{x} \sim \hat{p}_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$. 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 original 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 the estimate, denoted $\hat{p}_{\perp\!\!\!\perp}$, of the nuisance-randomized distribution $p_{\perp\!\!\!\perp}$. Maximizing the information translates to maximum likelihood because the entropy $\mathbf{H}_{\hat{p}_{\perp\!\!\!\perp}}(\mathbf{y})$ is constant with respect to the representation r_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 2 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 $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{z} \mid r_\gamma(\mathbf{x})) = 0$. With a scalar parameter λ, the likelihood maximization objective with a penalty to encourage conditional independence is

$$\max_{\theta, \gamma} \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x})} \log p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x})) - \lambda \mathbf{I}_{\hat{p}_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{z} \mid r_\gamma(\mathbf{x})). \tag{6}$$

The objective in eq. (6) 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. The exogenous noise and nuisance are jointly independent of the label under $p_{\perp\!\!\!\perp}$. The intuition behind the local optima is that the value of introducing information that predicts the label does not exceed the cost of the introduced conditional dependence. See appendix A.5 for a formal discussion and an a Gaussian example with local minima. Annealing the λ, which controls the cost of conditional dependence, would 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. 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*, the distillation step in NURD is

$$\max_{\theta, \gamma} \mathbb{E}_{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x})} \log p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x})) - \lambda \mathbf{I}_{\hat{p}_{\perp\!\!\!\perp}}([\mathbf{y}, r_\gamma(\mathbf{x})]; \mathbf{z}). \tag{7}$$

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. NURD makes use of the classification-based density-ratio estimation trick (Sugiyama et al., 2012). We use a *critic model* p_ϕ to estimate said density ratio. We describe this technique in appendix A.1 for completeness.

Algorithm. We give the full algorithms for both reweighting-NURD and generative-NURD in appendix A.1. The setup of nuisance-induced spurious correlations in eq. (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 for a procedure to build models when $p_{tr}(\mathbf{y}) \neq p_{te}(\mathbf{y})$.

4 Related Work

Nuisance versus Environment. In general, an environment is a distribution with a specific spurious correlation (Sagawa et al., 2019; Chang et al., 2020). In nuisance-induced spurious correlations, 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 collecting data from sufficiently different sources, nuisances can be specified from a single source of data using domain knowledge.

Domain Generalization. Domain generalization methods aim to build models with the goal of generalizing to unseen test data different from the training data (Gulrajani and Lopez-Paz, 2020). Recent work uses multiple *sufficiently different* environments to generalize to unseen test data that lies in the support of the given environments or subgroups (Li et al., 2018a; Arjovsky et al., 2019; Akuzawa et al., 2019; Mahajan et al., 2020; Krueger et al., 2020; Bellot and van der Schaar, 2020; Li et al., 2018b; Goel et al., 2020; Wald et al., 2021). Chang et al. (2020) develop a multi-environment objective to interpret neural network predictions that are robust to spurious correlations. Other work builds representations that are independent of the domain (or some special variable)(Ganin et al., 2016; Xie et al., 2017; Zhang et al., 2018; Ghimire et al., 2020; Adeli et al., 2021). Due to its focus on nuisances, NuRD works with data from a single environment.

Spurious correlations. Taking a distributional robustness (Duchi et al., 2021) approach, Sagawa et al. (2019) applied group-DRO to training data where the relative size of certain groups 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 (Lahoti et al., 2020; Martinez et al., 2021); as subgroups are unknown, they only find an approximate minimizer of the worst subgroup error in general even with infinite data. In contrast, NuRD builds predictive models with performance guarantees across all test distributions (that factorize as eq. (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. Wang and Culotta (2020b) 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 $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$.

Nuisance as the environment variable. Groups of samples defined by nuisance values can be treated as environments. Using nuisance itself as the environment variable will produce non-overlapping supports over images between environments in some of our experiments. 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. (2019); Krueger et al. (2020) will not produce invariant representations because the predictor can segment out the covariate space and learn separate predictors for each environment. Methods based on conditional distribution matching (Mahajan et al., 2020; Guo et al., 2021) 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 changes 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 (Schölkopf et al., 2012) falls in the class of the family of distributions in eq. (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 (Subbaswamy et al., 2019a;b). Subbaswamy et al. (2019a) perform *graph surgery* to build predictors (they learn $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$) that are stable across environments assuming access to $\mathbf{z}$ during test time. In nuisance-induced spurious correlations, the nuisance $\mathbf{z}$ is unavailable at test time in general, rendering shift-stable predictors inapplicable. However, if the nuisance is available at test time, the combined covariate-set $[\mathbf{x}, \mathbf{z}]$ is uncorrelating because $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{z} \mid [\mathbf{x}, \mathbf{z}]$ and maximally blocking because $r([\mathbf{x}, \mathbf{z}]) \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}, [\mathbf{x}, \mathbf{z}]$; then, theorem 1 shows $p_{\perp\!\!\!\perp}(\mathbf{y} \mid [\mathbf{x}, \mathbf{z}])$ is optimal. Two concurrent works also build models using the idea of a nuisance variable. Makar et al. (2021) 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. (2021) 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, while NuRD and its theory work with general high-dimensional nuisances. While counterfactual invariance promises that a representation will not vary with the

Table 1: Accuracy of NURD versus ERM on class conditional Gaussians. Oracle accuracy is 63%.

Method	Heldout p_{tr}	Heldout $\hat{p}_{\perp}$	p_{te}
ERM	$80 \pm 0\%$	–	$40 \pm 0\%$
generative-NURD	$62 \pm 1\%$	$61 \pm 1\%$	$59 \pm 1\%$
reweighting-NURD	$60 \pm 2\%$	$60 \pm 1\%$	$60 \pm 1\%$

nuisance, it does not produce optimal predictors in general because it rejects using parts of the covariates that are mixed functions of the label and the nuisance. On the other hand, uncorrelating representations allow using the nuisance to extract the information about the label from such mixed functions; this leads to better performance in some nuisance-varying families as we show using the theory of minimal sufficient statistics (Lehmann and Casella, 2006) in appendix A.6.

5 EXPERIMENTS

We evaluate the implementations of NURD on class-conditional Gaussians, Colored MNIST, Waterbirds (Sagawa et al., 2019), and chest X-rays (Irvin et al., 2019; Johnson et al., 2019). See appendix B for details about models, optimization, and other experimental specifics.

Model selection, baselines, and metrics. Models in both steps of NURD are selected using held-out subsets of the training data. For nuisance-randomization, this selection uses standard measures 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 Empirical Risk minimization (ERM) because, as discussed in section 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 that make use of additional labels in both training and at test time. For every method, we report the average accuracy and standard error of mean 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 ($\hat{p}_{\perp}$).

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

$$\mathbf{y} \sim \mathcal{B}(0.5) \quad \mathbf{z} \sim \mathcal{N}(a(2\mathbf{y} - 1), 1) \quad \mathbf{x} = [\mathbf{x}_1 \sim \mathcal{N}(\mathbf{y} - \mathbf{z}, 9), \mathbf{x}_2 \sim \mathcal{N}(\mathbf{y} + \mathbf{z}, 0.01)]. \tag{8}$$

The training and validation sets consist of 10000 and 2000 samples from $p_{tr} = q_{0.5}$ respectively, and the test set consists of 2000 samples from $p_{te} = q_{-0.9}$. We parameterize all models in reweighting and generative-NURD using neural networks.

Results. Mean and standard error of the accuracy of NURD and ERM averaged over 10 seeds. The oracle accuracy is the analytical accuracy of predicting with $p_{\perp}(\mathbf{y} \mid r^*(\mathbf{x}))$ where $r^*(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$ is the optimal uncorrelating representation; see appendix B for details. Both generative-NURD and reweighting-NURD outperform ERM and achieve close to the oracle accuracy.

5.2 COLORED-MNIST

We construct a colored MNIST dataset (Arjovsky et al., 2019; Gulrajani and Lopez-Paz, 2020) with images of 0s and 1s where 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 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, validation, and test data consist of $3941, 986$ and 4939 samples. The nuisance is the color which is given by the most intense pixel.

Results. See table 2 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%.

Table 2: 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	$90 \pm 0\%$	–	$10 \pm 0\%$
generative-NURD	$79 \pm 1\%$	$78 \pm 1\%$	$72 \pm 2\%$
reweighting-NURD	$75 \pm 0\%$	$76 \pm 0\%$	$75 \pm 0\%$

5.3 LEARNING TO CLASSIFY WATERBIRDS AND LANDBIRDS

Sagawa et al. (2019) consider the task of detecting the type of bird (water or land) from images where the background is a nuisance. Unlike Sagawa et al. (2019), we do not assume access to validation and test sets without background-label relationships. So, we filter from their dataset 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, validation, and test sets consist of $1936, 484$ and 500 samples respectively. We ensure that $p(\mathbf{y} = 1) = 0.5$ in train, validation, and test data. Thus, predicting the most-frequent-class achieves an accuracy of 0.5. The nuisance is the background of the image, which we represent as the pixels outside the central patch of 196×196 pixels. This is a high-dimensional nuisance which impacts many existing methods negatively; see section 4. The covariates are the whole image; see fig. 1.

Method	p_{tr}	$\hat{p}_{\perp\!\!\!\perp}$	p_{te}
ERM	$88 \pm 1\%$	–	$58 \pm 3\%$
reweighting-NURD	$83 \pm 1\%$	$83 \pm 1\%$	$78 \pm 1\%$

Figure 1: Results table 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 82%. On the right is an example of background patches in Waterbirds data. Models in NURD have both images only during training. Only the left image is given at test time.

Results. Figure 1 reports mean accuracies and standard error of mean across 10 random seeds. We construct a gold standard with additional labels of what the background is in the image in both the training and the test data. Using these labels, we build separate models for each background and predict on the corresponding subsets of the test data, achieving a gold standard accuracy of 82%. ERM uses the background to predict the label, as evidenced by its test accuracy of 58%. Reweighting-NURD uses the background patches to adjust for the spurious correlation and achieves an average accuracy of 78%, which is close to the gold standard. We do not report generative-NURD's performance because training on the generated images resulted in classifiers that predict as poorly as random chance on training images. This may be due to the small training dataset.

5.4 LEARNING TO LABEL PNEUMONIA FROM X-RAYS

In many problems such as classifying cows versus penguins in natural images, the background predicts the label; the background is the nuisance. Medical imaging datasets have this same property, where factors like lighting and device used to take the measurement leave a signature on the background (Zech et al., 2018; Badgeley et al., 2019). For this experiment, we construct a dataset by mixing two chest x-ray datasets, CheXpert and MIMIC, that have different styles of backgrounds. 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. Pneumonia cases are upsampled to make sure that in the training, validation, and test sets, healthy and pneumonia cases are equally probable. Thus, predicting the most-frequent-class achieves an accuracy of 0.5. The training, validation, and test datasets consist of $34200, 3804$, and 5602 samples respectively. The nuisance is the background of the chest X-ray, which we represent as the pixels outside the central patch of 24×24 pixels. This is a high-dimensional nuisance which impacts existing methods negatively; see section 4. The covariates are the whole image; see fig. 2 for an example.

Results. See fig. 2 for results aggregated across 10 random seeds. Building an oracle model in this experiment requires knowledge of all factors that correlate the background and the label. 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 background

Method	p_{tr}	$\hat{p}_{\perp}$	p_{te}
ERM	$87 \pm 0\%$	$-$	$30 \pm 1\%$
generative-NuRD	$60 \pm 3\%$	67 ± 2	$48 \pm 4\%$
reweighting-NuRD	$70 \pm 1\%$	$61 \pm 1\%$	$59 \pm 2\%$

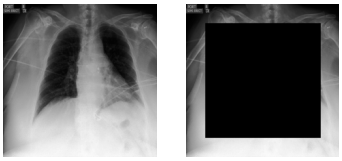

Figure 2: Results table and figure showing an example of the nuisance for chest X-rays. On the left are accuracies of NuRD versus ERM on X-ray data. On the right is an example of non-lung patches in Chest X-rays (right). Models in NuRD have both images only during training. Only the left image is given at test time.

with the label (Zech et al., 2018). ERM uses the nuisance to predict pneumonia, as evidenced by its test accuracy of 30%. Reweighting-NuRD uses the non-lung patches to adjust for the spurious correlation and achieves an average accuracy of 59%, a significant 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. We see a few seeds do not produce good test accuracies which may be due to a couple of ways generative nuisance-randomization may be imperfect: 1) incorrect estimate of the reliance of $\mathbf{x}$ on $\mathbf{z}$ with $\mathbf{y}$ fixed, 2) insufficient quality of generation which leads to bad generalization from generated images to real images.

6 Discussion

We formulate the nuisance-varying family. Changing nuisance-label relationships in this family make predictive models built on top of the covariates from samples of one member of the family unsuitable for other members. We say such a family manifests nuisance-induced spurious correlations. In the presence of nuisance-induced spurious correlations, we show that the best predictive model built under a broken nuisance-label relationship, i.e. the nuisance-randomized distribution $p_{\perp}$, is minimax optimal, when conditioning on the covariates does not make the label and the nuisance dependent. As nuisance-varying families do not always satisfy this condition, we introduce uncorrelating representations and characterize one that is *simultaneously* optimal for all members in the family. 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 it outperforms ERM on synthetic and real data by adjusting for nuisance-induced spurious correlations.

Limitations and the future. The nuisance-randomization steps in generative-NuRD and reweighting-NuRD model different distributions in the nuisance-varying family eq. (1). 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 (Guo et al., 2017), one must take care to build calibrated models for $p(\mathbf{y} \mid \mathbf{z})$. Adapting either calibration-focused losses (Kumar et al., 2018; Goldstein et al., 2020) or ensembling multiple models (Lakshminarayanan et al., 2016) may help produce calibrated probabilities. Recent work has empirically shown that weighting samples does not change the decision boundary that deep neural networks reach (Byrd and Lipton, 2019; Sagawa et al., 2019; 2020). Given groups based on pairs of nuisance-label values, Sagawa et al. (2020) suggest that subsampling equally from each group produces models more robust to spurious correlations than reweighting. However, both subsampling and reweighting are ineffective when the nuisance is high-dimensional because they produce nearly the same distribution as the original training set as groups may have at most 1 sample. 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. In our experiments, the training data contains a single environment. Methods for invariant representation learning 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 generative or reweighting nuisance-randomization methods to produce samples from other different environments using data from only a single environment would be a fruitful direction.

REFERENCES

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks, 2020.

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.

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.

Zhao Wang and Aron Culotta. Identifying spurious correlations for robust text classification. *arXiv preprint arXiv:2010.02458*, 2020a.

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

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.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

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.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

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, 2018a.

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.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

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.

Alexis Bellot and Mihaela van der Schaar. Accounting for unobserved confounding in domain generalization. *arXiv preprint arXiv:2007.10653*, 2020.

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, 2018b.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. *arXiv preprint arXiv:2008.06775*, 2020.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *arXiv preprint arXiv:2102.10395*, 2021.

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.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. *arXiv preprint arXiv:1705.11122*, 2017.

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.

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.

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.

John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

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.

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.

Zhao Wang and Aron Culotta. Robustness to spurious correlations in text classification via automatically generated counterfactuals. *arXiv preprint arXiv:2012.10040*, 2020b.

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.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

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, 2019a.

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*, 2019b.

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.

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.

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

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.

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.

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.

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.

Mark Goldstein, Xintian Han, Aahlad Puli, Adler Perotte, and Rajesh Ranganath. X-cal: Explicit calibration for survival analysis. *Advances in Neural Information Processing Systems*, 33, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

A THEORETICAL DETAILS

A.1 THE FULL NURD ALGORITHM

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 λ, 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.

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 the label.

Nuisance Randomization step;
Initialize $p_\alpha(\mathbf{y} \mid \mathbf{z})$;
for N_w *iterations* **do**
  Sample training batch $B \sim D$;
  Compute likelihood $\sum_{(\mathbf{y}_i, \mathbf{z}_i) \in B} \log p_\alpha(\mathbf{y}_i \mid \mathbf{z}_i)$;
  Update α to maximize this likelihood (via Adam for example);
end
Estimate the marginal distribution over the label $\hat{p}(\mathbf{y})$;
Produce weights $w_i = \hat{p}(\mathbf{y}_i) / p_\alpha(\mathbf{y}_i \mid \mathbf{z}_i)$ for each $(\mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i) \in D$;
Distillation step;
Initialize $r_\gamma, p_\theta, p_\phi$;
for N_p *iterations* **do**
  for N_c *iterations* **do**
    Sample training batch $B \sim D$;
    Sample independent copies of the nuisance 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 ϕ to maximize likelihood (via Adam for example);
  end
  Sample training batch $B \sim D$;
  Compute distillation objective

$$\frac{1}{|B|} \sum_{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i) \in B} w_i \left[\log p_\theta(\mathbf{y}_i \mid r_\gamma(\mathbf{x}_i)) - \lambda \log \frac{p_\phi(\ell = 1 \mid \mathbf{y}_i, \mathbf{z}_i, r_\gamma(\mathbf{x}_i))}{1 - p_\phi(\ell = 1 \mid \mathbf{y}_i, \mathbf{z}_i, r_\gamma(\mathbf{x}_i))} \right];$$

  Update θ, γ to maximize objective (via Adam for example).
end
Return $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$.

Density-ratio trick in Distillation. The density-ratio trick for estimating mutual information (Sugiyama et al., 2012) 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 model. We refer to p_ϕ as the critic model.

Accounting for shifts in the marginal label distribution NURD relies on the assumption in eq. (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 $\mathcal{F}' = \{p_{te}(\mathbf{y}) / p_{tr}(\mathbf{y}) p_D(\mathbf{y}, \mathbf{z}, \mathbf{x})\}$ 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 lower-bound on any test distribution $p_{te} \in \mathcal{F}'$.

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 λ, 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 the label.

Nuisance Randomization step;
for N_w *iterations* **do** nuisance
  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 β 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 batch from generated training data $B \sim \hat{D}$;
    Sample an independent copies of the nuisance 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 ϕ to maximize likelihood (via Adam for example);
  end
  Sample batch from generated training data $B \sim \tilde{D}$;
  Compute distillation objective

$$\frac{1}{|B|} \sum_{(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k) \in B} \left[\log p_\theta(\mathbf{y}_k \mid r_\gamma(\mathbf{x}_k)) - \lambda \log \frac{p_\phi(\ell = 1 \mid \mathbf{y}_k, \mathbf{z}_k, r_\gamma(\mathbf{x}_k))}{1 - p_\phi(\ell = 1 \mid \mathbf{y}_k, \mathbf{z}_k, r_\gamma(\mathbf{x}_k))} \right];$$

  Update θ, γ to maximize objective (via Adam for example).
end
Return $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$.

A.2 KEY LEMMAS FOR UNCORRELATING REPRESENTATIONS $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$

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. (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*

$$\texttt{Perf}_{p_{te}}(p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))) = \texttt{Perf}_{p_{te}}(p(\mathbf{y})) + \mathop{\mathbb{E}}_{p_{te}(\mathbf{y}, \mathbf{z})} KL \left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \right]. \quad (9)$$

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 $\mathbf{KL}$ terms and expectations of log-ratios are all well-defined:

$$\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_{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}) \parallel 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}))} \\
&= \mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{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})} \mathrm{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})} \mathrm{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})} \mathrm{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})} \mathrm{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})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] - \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{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 $\mathbf{KL}$ term:

$$\begin{aligned}
\mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] - &\mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] \\
&= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel p_{\perp\!\!\!\perp}(r(\mathbf{x}) \mid \mathbf{z})\right] \qquad (10) \\
&= \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z})} \mathrm{KL}\left[p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})\right].
\end{aligned}$$

Rearranging these terms completes the proof. □

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 $r \in \mathcal{R}(p_{\perp\!\!\!\perp,1})$,

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

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

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

$$
\mathbb{E}_{p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x}))} p_{\perp\!\!\!\perp,1}(\mathbf{y} \mid r(\mathbf{x})) = \mathbb{E}_{p_{\perp\!\!\!\perp,2}(\mathbf{z} \mid r(\mathbf{x}))} p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) = p_{\perp\!\!\!\perp,2}(\mathbf{y} \mid r(\mathbf{x})). \tag{11}
$$

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, $r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp,1}) \iff r(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp,2})$.

□

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 $\frac{p'_D(\mathbf{z} \mid \mathbf{y})}{p_D(\mathbf{z} \mid \mathbf{y})} > 0$,

$$
p'_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) p'_D(\mathbf{z} \mid \mathbf{y}) p(\mathbf{y}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) p_D(\mathbf{z} \mid \mathbf{y}) p(\mathbf{y}) \frac{p'_D(\mathbf{z} \mid \mathbf{y})}{p_D(\mathbf{z} \mid \mathbf{y})} = p_D(\mathbf{y}, \mathbf{z}, \mathbf{x}) \frac{p'_D(\mathbf{z} \mid \mathbf{y})}{p_D(\mathbf{z} \mid \mathbf{y})}. \tag{12}
$$

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

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

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.3 Optimal uncorrelating representations

Theorem 1. *Consider a pair $r, r_2 \in \mathcal{R}(p_{\perp\!\!\!\perp})$ such that r_2 **blocks** r: $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r_2(\mathbf{x})$. Then r_2 dominates r in performance on every test distribution in $\mathcal{F}$:*

$$
\forall p_{te} \in \mathcal{F}, \qquad \texttt{Perf}_{p_{te}}(r_2(\mathbf{x}))\,) \geq \texttt{Perf}_{p_{te}}(r(\mathbf{x}))).
$$

*Let $\exists r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})$ that is **maximally blocking**: $\forall r \in \mathcal{R}(p_{\perp\!\!\!\perp})$, $\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r^*(\mathbf{x})$. Then, $r^*(\mathbf{x})$ is **optimal** in that $r^*(\mathbf{x})$ dominates every uncorrelating $r(\mathbf{x})$ in performance on every $p_{te} \in \mathcal{F}$:*

$$
\forall p_{te} \in \mathcal{F}, \forall r \in \mathcal{R}(p_{\perp\!\!\!\perp}) \qquad \texttt{Perf}_{p_{te}}(r^*(\mathbf{x}))\,) \geq \texttt{Perf}_{p_{te}}(r(\mathbf{x}))).
$$

Proof. (proof for theorem 1) For readability, let $\ell(r_2) = \mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))\right]$. For the first part, 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. (1):

$$
p_D(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) = \frac{p_D(r_2(\mathbf{x}), r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}{p_D(r(\mathbf{x}) \mid \mathbf{y}, \mathbf{z})}
$$

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

Next, we will show that

$$\mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] \tag{13}$$
$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})\right] \tag{14}$$

The steps are similar to lemma 1's proof

$$\mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right] = \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{te}(\mathbf{y} \mid \mathbf{x})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))} + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{x})} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \tag{15}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))} \tag{16}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z})} \qquad \{\text{as } r, r_2 \in \mathcal{R}(p_{\perp\!\!\!\perp})\} \tag{17}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}), r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z})} \qquad \{\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} r(\mathbf{x}) \mid \mathbf{z}, r_2(\mathbf{x})\} \tag{18}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(\mathbf{y}, r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}), \mathbf{z}) p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})} \tag{19}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}),r_2(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})} \tag{20}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))} \mathbb{E}_{p_{te}(r_2(\mathbf{x}) \mid \mathbf{y},\mathbf{z},r(\mathbf{x}))} \log \frac{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, r(\mathbf{x}), \mathbf{z})}{p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})} \tag{21}$$

$$= \ell(r_2) + \mathbb{E}_{p_{te}(\mathbf{y},\mathbf{z},r(\mathbf{x}))} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}, r(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(r_2(\mathbf{x}) \mid r(\mathbf{x}), \mathbf{z})\right] \tag{22}$$

Noting that $\mathbf{KL}$ is non-negative and that `Perf` is negative-$\mathbf{KL}$ proves the theorem:

$$\mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r_2(\mathbf{x}))\right] \leq \mathbb{E}_{p_{te}(\mathbf{x})} \mathrm{KL}\left[p_{te}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))\right]. \tag{23}$$

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 $\mathbf{KL}$, the proof follows that r^* dominates r in performance. □

Theorem 2. *Assume $r^* \in \mathcal{R}(p_{\perp\!\!\!\perp})$ is maximally blocking, then 1) for all $r'(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$, $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) \geq \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r'(\mathbf{x}))$, 2) for $r'(\mathbf{x}) \in \mathcal{R}(p_{\perp\!\!\!\perp})$ such that $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r'(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x}))$, r' matches the performance of r^*, $\mathtt{Perf}_{p_{te}}(r^*(\mathbf{x})) = \mathtt{Perf}_{p_{te}}(r'(\mathbf{x}))$, and is optimal.*

Proof. (of theorem 2) Theorem 1 implies 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 a 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.

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

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

This difference is non-negative for any $r \in \mathcal{R}(p_{\perp\!\!\!\perp})$:

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

Any information-equivalent representation r' which satisfies $\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; r^*(\mathbf{x})) = \mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}, r'(\mathbf{x}))$ implies

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

Under this condition, theorem 1 implies

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

However, as $r' \in \mathcal{R}(p_{\perp\!\!\!\perp})$ and that $r^*(\mathbf{x})$ is maximally blocking, theorem 1 implies

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

The only way both these conditions hold is if

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

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.4 Minimax optimality

Proposition 1. *Consider a nuisance-varying family $\mathcal{F}$ (eq. (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*

$$\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z} \implies \exists p'_D \in \mathcal{F} s.t. \left[\mathbb{E}_{p'_D(\mathbf{x})} KL\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_D(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y})\right] > 0. \tag{24}$$

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, note that lemma 1 with $p_{te} = p'_D$ with $r(\mathbf{x}) = \mathbf{x}$ gives

$$\begin{aligned}
&\mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y}) - \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] \\
&\qquad = \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y})\right] - \mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] \\
&\qquad = \mathbb{E}_{p'_D(\mathbf{y}, \mathbf{z})} \mathrm{KL}\left[p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{y})} p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})\right]. \\
&\qquad = \mathbb{E}_{p'_D(\mathbf{y}, \mathbf{z})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) \parallel p_{\perp\!\!\!\perp}(\mathbf{x} \mid \mathbf{z})\right] \geq 0.
\end{aligned} \tag{25}$$

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

$$\max_{p'_D \in \mathcal{F}} \left[\mathbb{E}_{p'_D(\mathbf{x})} \mathrm{KL}\left[p'_D(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{p'_D}(\mathbf{x}; \mathbf{y})\right] \leq 0. \tag{26}$$

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. (26) 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 $\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z}$,

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

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

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

□

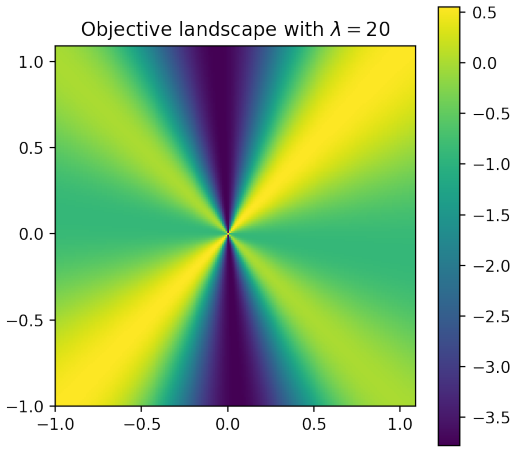

Figure 3: Landscape of values of the objective in eq. (6) for the example in eq. (2) for linear representations $r_{u,v}(\mathbf{x}) = u\mathbf{x}_1 + v\mathbf{x}_2$. Local maxima correspond to representations $r_{-u,u}$ and global maxima to representations $r_{u,u}$.

A.5 Distillation details and a local optima example for eq. (6)

The objective in eq. (6) 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 $(\boldsymbol{\epsilon}, \mathbf{z}) \perp\!\!\!\perp_{p_\perp} \mathbf{y}$. Then,

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

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 $p_\perp$ meaning it does not predict the label.

Local minima example for conditional information regularization eq. (6). Figure 3 plots the value of the objective in eq. (6) 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). 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. 3 shows, are local maxima on the landscape of the maximization objective in eq. (6). 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. $[r(\mathbf{x}), \mathbf{y}] \perp\!\!\!\perp_{p_\perp} \mathbf{z}$,*

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

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

NuRD *maximizes the information term $\mathbf{I}_{p_\perp}(\mathbf{y}; r_\gamma(\mathbf{x}))$ 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 $r \in \mathcal{R}(p_\perp)$ and $\forall p_{te} \in \mathcal{F}$,

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

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

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

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

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

□

Performance gaps between joint independence and the uncorrelating property 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.6.1, we use the theory of minimal sufficient statistics (Lehmann and Casella, 2006) 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.6 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 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 Veitch et al. (2021). 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.6.1 by constructing a nuisance-varying family where the optimal performance is achieved by an uncorrelating representation that is *dependent* on the nuisance.

A.6.1 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 3. *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*

$$
\forall p_{te} \in \mathcal{F} \qquad \max_{r \in \mathcal{R}_J} \texttt{Perf}_{p_{te}}(r(\mathbf{x})) \leq \max_{r \in \mathcal{R}_C} \texttt{Perf}_{p_{te}}(r(\mathbf{x})), \tag{27}
$$

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

$$
\max_{r \in \mathcal{R}_J} \texttt{Perf}_{p_{te}}(r(\mathbf{x})) < \max_{r \in \mathcal{R}_C} \texttt{Perf}_{p_{te}}(r(\mathbf{x})), \tag{28}
$$

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 $\texttt{Perf}_{p_{te}}(f(\mathbf{x})) = \texttt{Perf}_{p_{te}}(\mathbf{x})$). We know however that $\mathbf{x} \in \mathcal{R}_C$, completing the proof.

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

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

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 $p_{\perp\!\!\!\perp} \in \mathcal{F}$,

$$\begin{aligned}
0 = \texttt{Perf}_{p_{\perp\!\!\!\perp}}(f(\mathbf{x})) &= \texttt{Perf}_{p_{\perp\!\!\!\perp}}(T(\mathbf{x})) && (29)\\
&= -\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid T(\mathbf{x})\right] && (30)\\
&= -\mathbb{E}_{p_{\perp\!\!\!\perp}(\mathbf{x})} \mathrm{KL}\left[p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x}, T(\mathbf{x})) \parallel p_{\perp\!\!\!\perp}(\mathbf{y} \mid T(\mathbf{x})\right] && (31)\\
&= -\mathbf{I}_{p_{\perp\!\!\!\perp}}(\mathbf{y}; \mathbf{x} \mid T(\mathbf{x})) && (32)\\
&\implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x} \mid T(\mathbf{x}). && (33)
\end{aligned}$$

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 $p_{\perp\!\!\!\perp} \in \mathcal{F}$. We define $p_{\perp\!\!\!\perp}$ with binary $\mathbf{y}$ and a normal $\mathbf{z}$ be generated as $p(\mathbf{y})p(\mathbf{z})p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$: with $\rho : \{0, 1\} \times \{0, 1\} \to (0, 1)$, 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 $\mathbf{y}$, 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 $\mathbf{x} = [b, a]$, we let $f(\mathbf{x}) = f(b, a) = p(\mathbf{y} = 1 \mid \mathbf{x} = [b, a])$. We write $f(b, a)$ for different values of b here,

$$\begin{aligned}
f(1, a) = p(\mathbf{y} = 1 \mid \mathbf{x} = [1, a]) &= p(\mathbf{y} = 1 \mid \mathbf{b} = 1, \mathbf{a} = a) \\
&= \frac{p(\mathbf{b} = 1, \mathbf{y} = 1 \mid \mathbf{a} = a)}{p(\mathbf{b} = 1 \mid \mathbf{a} = a)} \\
&= \frac{p(\mathbf{y} = 1) p(\mathbf{b} = 1 \mid \mathbf{y} = 1, \mathbf{a} = a)}{\sum_{y \in \{0,1\}} p(\mathbf{y} = y) p(\mathbf{b} = 1 \mid \mathbf{y} = y, \mathbf{a} = a)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{0.5p(\mathbf{b}=1 \mid \mathbf{y}=1, \mathbf{a}=a)}{0.5\sum_{y\in\{0,1\}}\left(p(\mathbf{b}=1 \mid \mathbf{y}=y, \mathbf{a}=a)\right)} \\
&= \frac{\rho(1,a)}{\rho(0,a)+\rho(1,a)}. \\
f(0,a) &= p(\mathbf{y}=1 \mid \mathbf{x}=[0,a]) = p(\mathbf{y}=1 \mid \mathbf{b}=0, \mathbf{a}=a) \\
&= \frac{p(\mathbf{b}=0, \mathbf{y}=1 \mid \mathbf{a}=a)}{p(\mathbf{b}=0 \mid \mathbf{a}=a)} \\
&= \frac{p(\mathbf{y}=1)p(\mathbf{b}=0 \mid \mathbf{y}=1, \mathbf{a}=a)}{\sum_{y\in\{0,1\}} p(\mathbf{y}=y)p(\mathbf{b}=0 \mid \mathbf{y}=y, \mathbf{a}=a)} \\
&= \frac{0.5(1-p(\mathbf{b}=1 \mid \mathbf{y}=1, \mathbf{a}=a))}{0.5\left(\sum_{y\in\{0,1\}} 1-p(\mathbf{b}=1 \mid \mathbf{y}=y, \mathbf{a}=a)\right)} \\
&= \frac{1-\rho(1,a)}{2-\rho(0,a)-\rho(1,a)}.
\end{aligned}
$$

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 $a=1$,

$$
f(1,a) = \frac{\rho(1,1)}{\rho(0,1)+\rho(1,1)} = \frac{0.5}{0.5+0.5} = 0.5, \tag{34}
$$

$$
f(0,a) = \frac{1-\rho(1,1)}{2-\rho(0,1)-\rho(1,a)} = \frac{1-0.5}{2-0.5-0.5} = 0.5, \tag{35}
$$

and with $a=0$,

$$
f(1,a) = \frac{\rho(1,0)}{\rho(0,0)+\rho(1,0)} = \frac{0.9}{0.1+0.9} = 0.9, \tag{36}
$$

$$
f(0,a) = \frac{1-\rho(1,0)}{2-\rho(0,0)-\rho(1,0)} = \frac{1-0.9}{2-0.1-0.9} = 0.1. \tag{37}
$$

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 Lehmann and Casella (2006) which constructs a minimal sufficient statistic for a finite family of distributions $\mathcal{P} = \{p_i; i \in \{0, K-1\}\}$. The minimal sufficient statistic is

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

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,

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

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 $p(\mathbf{y} = 1 \mid \mathbf{x})$ 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.7 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 $\mathbf{y} \not\perp\!\!\!\perp_{p_D} \mathbf{z}$, 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 $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2$):

$$\Sigma = \begin{pmatrix} 1 & 1 & a \\ 1 & 2 & a \\ a & a & a^2+1 \end{pmatrix} \qquad \implies \Sigma_{1,2} = [1, a], \quad \Sigma_{2,2,}^{-1} = \frac{1}{a^2+2} \begin{pmatrix} a^2+1 & -a \\ -a & 2 \end{pmatrix}, \tag{38}$$

The conditional mean and variance are:

$$\implies \mathbb{E}_{q_a}[\mathbf{y} \mid \mathbf{x} = \mathbf{x}] = \Sigma_{1,2}\Sigma_{2,2}^{-1}\mathbf{x} = \frac{1}{a^2+2}[1, a]\mathbf{x} \tag{39}$$

$$\sigma^2_{q_a}(\mathbf{y} \mid \mathbf{x}) = \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1} = 1 - \frac{a^2+1}{a^2+2} = \frac{1}{a^2+2}. \tag{40}$$

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

$$\begin{aligned} &\mathbb{E}_{q_a(\mathbf{x})} \mathrm{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q_b(\mathbf{y} \mid \mathbf{x})\right] - \mathbf{I}_{q_a}(\mathbf{x}; \mathbf{y}) \\ &\quad = \mathbb{E}_{q_a(\mathbf{x})} \mathrm{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q_b(\mathbf{y} \mid \mathbf{x})\right] - \mathbb{E}_{q_a(\mathbf{x})} \mathrm{KL}\left[q_a(\mathbf{y} \mid \mathbf{x}) \parallel q(\mathbf{y})\right]. \\ &\quad = \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})} \log \frac{q_a(\mathbf{y} \mid \mathbf{x})}{q_b(\mathbf{y} \mid \mathbf{x})} - \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})} \log \frac{q_a(\mathbf{y} \mid \mathbf{x})}{q(\mathbf{y})}. \\ &\quad = \mathbb{E}_{q_a(\mathbf{x},\mathbf{y})} \left(\log q(\mathbf{y}) - \log q_b(\mathbf{y} \mid \mathbf{x})\right). \end{aligned} \tag{41}$$

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:

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

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

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

The proof follows for any a such that

$$\frac{(b^2+1-ab)^2 - 1}{2(b^2+2)} + \log\sqrt{1/b^2+2} = \frac{(b^2+1-ab)^2 - 1}{2(b^2+2)} - \frac{1}{2}\log\left(b^2+2\right) > 0 \tag{42}$$

Let $a = b + \frac{1+\nu}{b}$ for some scalar ν. Then,

$$\frac{(b^2+1-ab)^2 - 1}{2(b^2+2)} - \frac{1}{2}\log\left(b^2+2\right) = \frac{\nu^2 - 1}{2(b^2+2)} - \frac{1}{2}\log\left(b^2+2\right) > 0 \tag{43}$$

for any $|\nu| > 1 + (b^2+2)\log(b^2+2)$.

□

A.8 Gaussian example where $p_{tr}(\mathbf{y} \mid \mathbf{x})$ and $p_{\perp}(\mathbf{y} \mid \mathbf{x})$ perform worse than $p(\mathbf{y})$ under p_{te}

Consider the following data generating process with q_a, indexed by $a \in \mathbb{R}$ and fixed σ^2:

$$\mathbf{y} \sim \mathcal{N}(0, 1) \quad \mathbf{z} \sim \mathcal{N}(a\mathbf{y}, 0.5) \quad \mathbf{x} = \left[\mathbf{x}_1 \sim \mathcal{N}(\mathbf{y} - \mathbf{z}, \sigma^2 - 0.5), \mathbf{x}_2 \sim \mathcal{N}(\mathbf{y} + \mathbf{z}, 0.5)\right]. \tag{44}$$

Letting $\sigma^2 = 2$ recovers the example in eq. (2). We keep σ^2 for ease of readability. First, we derive the relative performance of $q(\mathbf{y})$ w.r.t. $q_b(\mathbf{y} \mid \mathbf{x})$ under q_a.

Relative performance of q_b and $q(\mathbf{y})$ under q_a We first derive the performance of some $q_b(\mathbf{y} \mid \mathbf{x})$ relative to $q(\mathbf{y})$under some q_a. Rewrite $\mathbf{x} = \left[(1-a) * \mathbf{y} + \sqrt{\sigma^2}\epsilon_1, (1+a) * \mathbf{y} + \epsilon_2)\right]$, where $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0,1)$. 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, with the following covariance matrix (over $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2$):

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

$$\implies \Sigma_{1,2} = [1-a, 1+a], \quad \Sigma_{2,2,}^{-1} = \frac{1}{\sigma^2(1+a)^2 + (1-a)^2 + \sigma^2} \begin{pmatrix} (1+a)^2 + 1 & -(1-a^2) \\ -(1-a^2) & (1-a)^2 + \sigma^2 \end{pmatrix},$$

$$\implies \mathbb{E}_{q_a}[\mathbf{y} \mid \mathbf{x} = \mathbf{x}] = \Sigma_{1,2}\Sigma_{2,2}^{-1}\mathbf{x} = \frac{1}{\sigma^2(1+a)^2 + (1-a)^2 + \sigma^2}[(1-a), \sigma^2(1+a)]\mathbf{x}$$

$$\implies \sigma^2_{q_a}(\mathbf{y} \mid \mathbf{x}) = \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1} = 1 - \frac{\sigma^2(1+a)^2 + (1-a)^2}{\sigma^2(1+a)^2 + (1-a)^2 + \sigma^2} = \frac{\sigma^2}{\sigma^2(1+a)^2 + (1-a)^2 + \sigma^2}.$$

The relative performance of $q(\mathbf{y})$ (recall `Perf` is negative $\mathbf{KL}$) w.r.t $q_b(\mathbf{y} \mid \mathbf{x})$ 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:

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

As $\sigma^2_{q_b}(\mathbf{y} \mid \mathbf{x})$ does not vary with $\mathbf{x}$, only the first two terms vary with $\mathbf{y}, \mathbf{x}$ and we will compute the expectation over those:

$$\begin{aligned}
&\mathbb{E}_{q_a(\mathbf{x},\mathbf{y})}\left(\log q(\mathbf{y}) - \log q_b(\mathbf{y} \mid \mathbf{x})\right) \\
&= \mathbb{E}_{q_a(\mathbf{x})q_a(\mathbf{y} \mid \mathbf{x})}\left(-\frac{\mathbf{y}^2}{2} + (\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)\frac{\left(\mathbf{y} - \frac{1}{\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2}[(1-b), \sigma^2(1+b)]\mathbf{x}\right)^2}{2\sigma^2}\right) \\
&= \mathbb{E}_{q(\mathbf{y})}\left(-\frac{\mathbf{y}^2}{2}\right) + (\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)\mathbb{E}_{q(\mathbf{y})q_a(\mathbf{x} \mid \mathbf{y})}\frac{\left(\mathbf{y} - \frac{1}{\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2}[(1-b), \sigma^2(1+b)]\mathbf{x}\right)^2}{2\sigma^2} \\
&= -\frac{1}{2} + (\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)\mathbb{E}_{q(\mathbf{y})q_a(\mathbf{x} \mid \mathbf{y})}\frac{\left((\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)\mathbf{y} - [(1-b), \sigma^2(1+b)]\mathbf{x}\right)^2}{2\sigma^2(\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)^2} \\
&= -\frac{1}{2} + \mathbb{E}_{q(\mathbf{y})q_a(\mathbf{x} \mid \mathbf{y})}\frac{\left((\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)\mathbf{y} - [(1-b), \sigma^2(1+b)]\mathbf{x}\right)^2}{2\sigma^2(\sigma^2(1+b)^2 + (1-b)^2 + \sigma^2)} \\
&\qquad \left(\mathbf{x} = [(1-a)\mathbf{y} + \sqrt{\sigma^2}\epsilon_1, (1+a)\mathbf{y} + \epsilon_2]\right) \\
&= -\frac{1}{2} +
\end{aligned}$$

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{y})q(\epsilon_y)q(\epsilon)} \frac{\left((\sigma^2(1+b)^2+(1-b)^2+\sigma^2)\mathbf{y}-(1-b)\sqrt{\sigma^2}\epsilon_1-(1-a)(1-b)\mathbf{y}-\sigma^2(1+a)(1+b)\mathbf{y}-\sigma^2(1+b)\epsilon_2\right)^2}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} \\
&= -\frac{1}{2} + \mathbb{E}_{q(\mathbf{y})q(\epsilon_y)q(\epsilon)} \frac{\left(\left(\sigma^2(1+b)^2+(1-b)^2-\sigma^2(a+b+ab)-(1-a)(1-b)\right)\mathbf{y}-(1-b)\sqrt{\sigma^2}\epsilon_1-\sigma^2(1+b)\epsilon_2\right)^2}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} \\
&\qquad \{\text{let } C(a,b) = \left(\sigma^2(1+b)^2+(1-b)^2-\sigma^2(a+b+ab)-(1-a)(1-b)\right)\} \\
&= -\frac{1}{2} + \frac{\text{var}\left(C(a,b)\mathbf{y}\right) + \text{var}((1-b)\sqrt{\sigma^2}\epsilon_1) + \text{var}(\sigma^2(1+b)\epsilon_2)}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} \\
&= -\frac{1}{2} + \frac{(C(a,b))^2\,\text{var}\left(\mathbf{y}\right) + \sigma^2(1-b)^2\text{var}(\epsilon_1) + (\sigma^2)^2(1+b)^2\text{var}(\epsilon_2)}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} \\
&= -\frac{1}{2} + \frac{(C(a,b))^2 + \sigma^2(1-b)^2 + (\sigma^2)^2(1+b)^2}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} \\
&= \frac{C(a,b)^2-(\sigma^2)^2}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)}
\end{aligned}
$$

Note that $\log\sqrt{\sigma^2_{q_b}(\mathbf{y}\mid\mathbf{x})} - \log\sqrt{\frac{2}{\sigma^2(1+b)^2+(1-b)^2+\sigma^2}} = \frac{1}{2}\log\frac{\left(\sigma^2(1+b)^2+(1-b)^2+\sigma^2\right)}{\sigma^2}$, With this, we have the ability to bound the performance of $q_b(\mathbf{y}\mid\mathbf{x})$ under q_a relative to $q(\mathbf{y})$.

The conditional $q_1(\mathbf{y}\mid\mathbf{x})$ performs worse than $q_b(\mathbf{y})$ on q_{-1} We show here that $p_{tr}(\mathbf{y}\mid\mathbf{x}) = q_1(\mathbf{y}\mid\mathbf{x})$ does worse than $q_1(\mathbf{y}) = p(\mathbf{y})$ on $p_{te} = q_{-1}$. Letting $a = -1 \implies (1+a) = 0, (1-a) = -1$ and $b = 1 \implies (1+b) = 2, (1-b) = 0$ then

$$C(a,b) = \sigma^2(1+b)^2+(1-b)^2-\sigma^2(a+b+ab)-(1-a)(1-b) = \sigma^2 * 2^2+\sigma^2-\sigma^2 = 4\sigma^2.$$

Then

$$
\begin{aligned}
&\frac{C(a,b)^2-(\sigma^2)^2}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} - \frac{1}{2}\log\frac{\left(\sigma^2(1+b)^2+(1-b)^2+\sigma^2\right)}{\sigma^2} \\
&\quad = \frac{16(\sigma^2)^2-(\sigma^2)^2}{2*\sigma^2(5*\sigma^2)} - \frac{1}{2}\log\frac{5*\sigma^2}{\sigma^2} \\
&\quad = \frac{3}{2} - \frac{1}{2}\log 5 > 0.
\end{aligned}
$$

Predictive failure when $\mathbf{x}$ is not uncorrelating Note that $q_0(\mathbf{y}\mid\mathbf{x}) = p_\perp(\mathbf{y}\mid\mathbf{x})$ corresponds to the nuisance randomized conditional of the label given the covariates. We show that $p_\perp(\mathbf{y}\mid\mathbf{x})$ performs worse than $p(\mathbf{y})$ on infinitely many q_a. Letting $b = 0$ in $C(a,b)$ gives

$$C(a,b) = \sigma^2(1+b)^2+(1-b)^2-\sigma^2(a+b+ab)-(1-a)(1-b) = \sigma^2+1+a\sigma^2-(1-a) = \sigma^2+a(1-\sigma^2).$$

Then, the performance of $p(\mathbf{y})$ relative to $p_\perp(\mathbf{y}\mid\mathbf{x})$ is

$$
\begin{aligned}
&\frac{C(a,b)^2-(\sigma^2)^2}{2\sigma^2(\sigma^2(1+b)^2+(1-b)^2+\sigma^2)} - \frac{1}{2}\log\frac{\left(\sigma^2(1+b)^2+(1-b)^2+\sigma^2\right)}{\sigma^2} \\
&\quad = \frac{\left(a^2(1-\sigma^2)^2+(\sigma^2)^2+2\sigma^2(1-\sigma^2)a\right)-(\sigma^2)^2}{2*\sigma^2(2*\sigma^2+1)} - \frac{1}{2}\log\frac{\left(2*\sigma^2+1\right)}{2} \\
&\qquad \{\text{letting } \sigma^2 = 2.\} \\
&\quad = \frac{a^2-4a}{20} - \frac{1}{2}\log\frac{5}{2}.
\end{aligned}
$$

This performance difference between $q(\mathbf{y})$ and $p_\perp(\mathbf{y}\mid\mathbf{x})$ is positive for any $a > 5$ or $a < -1$. Thus, for every test distribution q_a such that $a > 5$ or $a < -1$, $p_\perp(\mathbf{y}\mid\mathbf{x})$ performs worse than predicting without covariates.

B FURTHER EXPERIMENTAL DETAILS

Implementation details In section 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 $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}) \neq p_{tr}(\mathbf{y})$. When such mis-match occurs, $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ may perform even worse than predicting without covariates. Given an estimate of the nuisance-randomized distribution, $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x})$, we weight it again, $\frac{p_{tr}(\mathbf{y})}{\hat{p}_{\perp\!\!\!\perp}(\mathbf{y})} \hat{p}_{\perp\!\!\!\perp}$, to ensure that $p_{tr}(\mathbf{y}) = \hat{p}_{\perp\!\!\!\perp}(\mathbf{y})$.

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 representation $s_\psi(\mathbf{z})$ is different in the three experiments and we give these details below. In all experiments, we reported results with the distillation step optimized with a fixed $\lambda = 1$ and one epoch's worth of gradient steps for the critic model for each gradient step of the predictive model and the representation.

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. Specifically, model selection uses 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. Similarly, in reweighting-NURD, the model for $p_{tr}(\mathbf{y} \mid \mathbf{z})$ is evaluated on a heldout subset of the training data.

Next, we give details about each individual experiment.

Theoretical details for Class Conditional Gaussians. Here, we show that $r^*(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$ satisfies $\mathbf{x} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z}$, which in turn implies r^* is maximally blocking. : $r \in \mathcal{R}(p_{\perp\!\!\!\perp}), r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z}$. First let the Gaussian noises in the two coordinates of $\mathbf{x}$, $\mathbf{x}_1$ and $\mathbf{x}_2$ (given $\mathbf{y}, \mathbf{z}$) be $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ respectively. By the definition of $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$, a few independencies hold and will be key: $\boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_2)$ and $\boldsymbol{\epsilon}_2 \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_1)$, and $(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2) \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z})$. Next, we derive a key conditional independence: $\boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} \mathbf{y} \mid (2\mathbf{y} + \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2, \mathbf{z})$. As $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2$ are Gaussian,

$$\boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} \boldsymbol{\epsilon}_2 \mid \mathbf{y}, \mathbf{z} \implies \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2 \mid \mathbf{y}, \mathbf{z}.$$

Then, as $\boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z})$, it holds that $\boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2)$ and

$$\boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2) \implies \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} (\mathbf{y}, \mathbf{z}, 2\mathbf{y} + \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2) \implies \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_D} \mathbf{y} \mid (2\mathbf{y} + \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2, \mathbf{z}).$$

The proof that $r^*(\mathbf{x})$ satisfies $\mathbf{x} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z}$ follows from noting that $r^*(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2 = 2\mathbf{y} + \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2$ which means

$$\boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1 \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z} \implies [r^*(\mathbf{x}), 2\mathbf{z} + \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1] \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z} \implies \mathbf{x} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z},$$

where the last step follows from noting $\mathbf{x}$ bijectively transforms into $[\mathbf{x}_1 + \mathbf{x}_2, \mathbf{x}_2 - \mathbf{x}_1] = [r^*(\mathbf{x}), 2\mathbf{z} + \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_1]$. Finally, by the laws of independence $\mathbf{x} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z} \implies r(\mathbf{x}) \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r^*(\mathbf{x}), \mathbf{z}$ for every function r (which is a superset of $\mathcal{R}(p_{\perp\!\!\!\perp})$). This means that theorem 1 holds and $r^*(\mathbf{x})$ is the optimal uncorrelating representation.

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 1 hidden layer with with 16 units and ReLU activations.

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 (Kingma and Ba, 2014) optimizer with a learning rate of 10^{-3}. To train $p_{tr}(\mathbf{y} \mid \mathbf{z})$ we use batch size of 64 and a batch size of 256 for the distillation step to train $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$. We optimized for a 100 epochs and chose the model based on the value of the objective in both stages of NURD on a heldout subset of the training data. We run the distillation step with a fixed $\lambda = 1$ and one epoch's worth of gradient steps (40) for the critic model for each gradient step of the predictive model and the representation. We saw performance similar to what was reported with other hyperparameter values like 20 critic steps and $\lambda = 100$.

Implementation details for Colored-MNIST. For reweighting-NURD, to use the same architecture for the representation and for in learning $p_{tr}(\mathbf{y} \mid \mathbf{z})$, we construct the nuisance as a 28×28 image with each pixel being equal to the most intense pixel in the original image in the training data. In generative-NURD, we use a PixelCNN model for $p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ used 10 masked convolutional layers each with 64 filters. The model was trained 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.

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 take 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 (Kingma and Ba, 2014) optimizer with a learning rate of 10^{-3}. To train $p_{tr}(\mathbf{y} \mid \mathbf{z})$ we use batch size of 64 and a batch size of 256 for the distillation step to train $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$. We optimized for 10 epochs and chose the model based on the value of the objective in both stages of NURD on a heldout subset of the training data set. We run the distillation step with a fixed $\lambda = 1$ and one 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 randomly re-initialize p_ϕ (which includes s_ψ) after every gradient step of the predictive model.

Further details for the Waterbirds. For generative-NURD, we use VQ-VAE 2 (Razavi et al., 2019) 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 1936 samples. The model for $\hat{p}_{\perp\!\!\!\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is two feedforward layers stacked on top of representation $r_\gamma(\mathbf{x})$. The model for $\hat{p}_{tr}(\mathbf{y} \mid \mathbf{z})$ 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 take 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 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 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 (Kingma and Ba, 2014) optimizer with a learning rate of 10^{-3}. To train $p_{tr}(\mathbf{y} \mid \mathbf{z})$ we use batch size of 64 and a batch size of 256 for the distillation step to train $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$. We run the distillation step with a fixed $\lambda = 1$ and one epoch's worth of gradient steps (12) for the critic model for each gradient step of the predictive model and the representation.

Further details for the chest X-ray experiments. We use a VQ-VAE2 (Razavi et al., 2019) 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 (Kingma and Ba, 2014) 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 $\hat{p}_{\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ is two feedforward layers stacked on top of representation $r_\gamma(\mathbf{x})$. The model in ERM for $p_{tr}(\mathbf{y} \mid \mathbf{x})$ uses the same architecture as $\hat{p}_{\perp}(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ 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 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 (Kingma and Ba, 2014) optimizer with a learning rate of 10^{-3}. To train $p_{tr}(\mathbf{y} \mid \mathbf{z})$ we use batch size of 64 and a batch size of 256 for the distillation step to train $p_\theta(\mathbf{y} \mid r_\gamma(\mathbf{x}))$ and $p_\phi(\ell \mid \mathbf{y}, \mathbf{z}, r_\gamma(\mathbf{x}))$. We optimized for 10 epochs and chose the model based on the value of the objective in both stages of NuRD on a heldout subset of the training data set. The distillation step is run with $\lambda = 1$ and one epoch's worth of gradient steps (133) for the critic model for each gradient step of the predictive model and the representation. To prevent the critic model from overfitting, we randomly re-initialize p_ϕ (which includes s_ψ) after every gradient step of the predictive model.