Contextual Reliability: When Different Features Matter in Different Contexts

Gaurav Ghosal [*1] **Amrith Setlur** [*2] **Daniel S. Brown** [3] **Anca D. Dragan** [1] **Aditi Raghunathan** [2]

Abstract

Deep neural networks often fail catastrophically by relying on spurious correlations. Most prior work assumes a clear dichotomy into spurious and reliable features; however, this is often unrealistic. For example, most of the time we do not want an autonomous car to simply copy the speed of surrounding cars—we don't want our car to run a red light if a neighboring car does so. However, we cannot simply enforce invariance to next-lane speed, since it could provide valuable information about an unobservable pedestrian at a crosswalk. Thus, universally ignoring features that are sometimes (but not always) reliable can lead to non-robust performance. We formalize a new setting called *contextual reliability* which accounts for the fact that the "right" features to use may vary depending on the context. We propose and analyze a two-stage framework called Explicit Non-spurious feature Prediction (ENP) which first identifies the relevant features to use for a given context, then trains a model to rely exclusively on these features. Our work theoretically and empirically demonstrates the advantages of ENP over existing methods and provides new benchmarks for contextual reliability.

1. Introduction

Despite remarkable performance on benchmarks, deep neural networks often fail catastrophically when deployed under slightly different conditions than they were trained on. Such failures are commonly attributed to the model relying on "spurious" features (*e.g.,* background) rather than "non-spurious" features that remain reliably predictive even for out of distribution inputs. Prior work has focused on learning models that rely exclusively on non-spurious features. We argue that such a neat delineation of features as "non-spurious" and "spurious" is often unrealistic. As an example, consider autonomous driving. In some cases, it can be dangerous to rely on the speed of cars in the neighboring lane. A neighboring car running a red light should not cause the agent to dangerously run a red light as well, and a neighboring car slowing to turn left should not cause an agent going straight to slow down. At a crosswalk, however, the braking of a neighboring car carries evidence of an unobserved pedestrian and should be treated as a cause for stopping. Thus, we observe that neighboring-lane speed cannot be globally treated either as spurious or non-spurious, it must be used or ignored *depending on the context*. Similarly, image backgrounds are often considered spurious in prior robustness research. However, we contend that there are contexts where the background is useful and should not be ignored, for example, when the foreground is occluded or ambiguous. In fact, humans often use the background to identify objects in such situations (Torralba, 2003).

In this work, we propose and study *contextual reliability*, a new setting that better captures the above nuances of real-world settings. We assume that the data comes from different latent contexts, each of which has a potentially different designation of spurious and non-spurious features. Thus, in contrast to prior settings, the optimal robust predictor might need to rely on different features in different contexts.

How to improve contextual reliability? The predominant approach to improving robustness enforces invariance across a fixed set of spurious features (Muandet et al., 2013; Blanchard et al., 2011; Gulrajani & Lopez-Paz, 2020; Rosenfeld et al., 2021) but such a set need not exist in our setting. The alternative active learning approaches do not require a predefined set of features, but actively source maximally uncertain data with the hope of breaking spurious correlations. These approaches also fail to address contextual reliability due to poor uncertainty estimates when different features are spurious across different contexts. Finally, robust optimization based approaches also fail to control for performance across multiple contexts without explicit information about the latent contexts of different training points. This suggests that we need *some additional information about the latent context* in order to improve contextual reliability.

[*]Equal contribution [1] University of California, Berkeley [2] Carnegie Mellon University [3] University of Utah. Correspondence to: Gaurav Ghosal <gauravrghosal@berkeley.edu>, Amrith Setlur <asetlur@cs.cmu.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

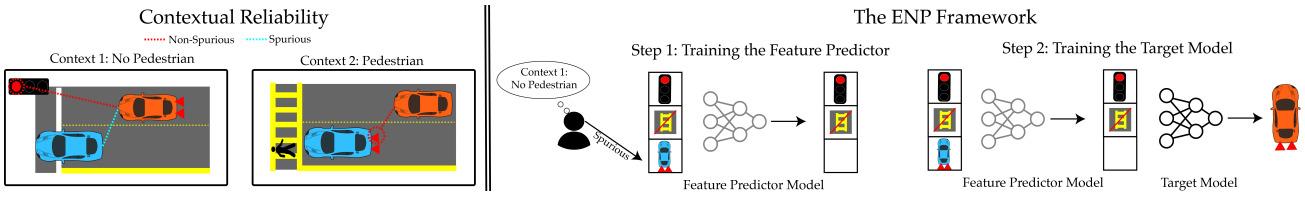

Figure 1. **Our Setting and Proposed Framework.** (**Left Panel**) Typically, it is dangerous for an autonomous car to get influenced by cars in the next lane—you do not want to run a red light if the car next to you does. Prior work provides methods to be invariant to the speed of neighboring cars. However, in the context of a pedestrian crossing, such an invariance is dangerous. The braking of the neighboring car can provide valuable information about a potential pedestrian. (**Right Panel**) We propose a two-stage framework for achieving reliable performance. In the first stage, a feature prediction model is trained to predict the set of non-spurious features from human annotations. In the second stage, we train a target model that is invariant to predicted spurious features.

Having established the necessity of context information, we consider the problem of harnessing it effectively. We propose a framework called *Explicit Non-spurious feature Prediction* (ENP). Ideally, we want to train a model that is invariant to the context-dependent spurious features. Rather than expecting end-to-end training to implicitly uncover contexts and respect their feature invariances, ENP employs a divide and conquer approach whereby the model first explicitly predicts the context before making a final prediction that respects the predicted context's invariances. To provide explicit supervision for the first identification step, we augment a *small* fraction of the training set with explicit annotations (termed *feature annotations*) on what features the optimal robust predictor should rely on. We analytically compare ENP to a variety of baselines (both with and without context information) in a simple linear setting. This allows us to precisely characterize the conditions when context information is helpful, and how different approaches to incorporating context compare. We also confirm these findings via simulations on linear models and neural networks.

Finally, we consider a variety of semi-synthetic and real-world datasets that require contextual reliability ranging from control to image classification and motion forecasting with real-world autonomous vehicle data from the Wayo Open Motion Dataset (WOMD) (Ettinger et al., 2021). On WOMD, we make use of crowd-sourced human annotations of vehicle spuriousness provided in (Roelofs et al., 2022). ENP offers consistent gains over baselines across *all* these settings, offerings gains of around 15% in control environments and 6% in image classification, and 5% on WOMD. Ultimately, we hope that our setting of contextual reliability and proposed ENP method serve as a setting and benchmark for addressing this important real-world challenge.

2. Related Works

Robustness in supervised learning. Prior works in machine learning have investigated various distribution shift types: subpopulation shifts (Hu et al., 2018; Sagawa et al., 2019b), input perturbations and adversarial shifts (Goodfellow et al., 2014; Raghunathan et al., 2018), and generalization to new domains (Gulrajani & Lopez-Paz, 2020). Our setting of contextual reliability is most closely related to subpopulation shifts and one line of work to address this is Distributionally Robust Optimization (DRO) (Duchi et al., 2019; Liu & Ziebart, 2014). DRO optimizes the worst-subpopulation performance, which can be over-conservative and statistically inefficient (Hu et al., 2018). A slightly different setting is domain generalization, where the goal is to learn a predictor that extrapolates to unseen subpopulations (Li et al., 2018). Usually, assumptions about the relationship between domains are made in order to allow the robust predictor to be reliably identified (Muandet et al., 2013). Our setting of contextual reliability is similar to domain generalization in that the goal is to be optimally robust on every context, however, we do not consider the task of extrapolating to new contexts and do not rely on end-to-end training. Rather, we explicitly infer the context and enforce invariance to the spurious features in each context.

Robustness in imitation learning. Imitation learning naturally suffers from distribution shift: during training, the distribution of observed states arises from the expert policy, while in testing it arises from the learned policy (Ross et al., 2011; Tien et al., 2023). As a result, prior works have contributed methods for achieving robustness to these shifts. De Haan et al. (2019) propose learning the true causal graph of an expert's policy through online execution and expert queries. Lyle et al. (2021) examine uncertainty-based exploration for disambiguating spurious correlations in both online and imitation learning settings. Although these methods have achieved success in previously studied robustness settings, they primarily rely on careful data collection in the environment. As we demonstrate in Section 6, techniques for achieving this often fail under contextual reliability. Other work on robust imitation learning considers Bayesian robustness to uncertainty over the objective (Brown et al., 2020; Javed et al., 2021), in contrast to the contextual feature reliability we study.

Incorporating prior knowledge. Recent works have proposed methods for leveraging prior human knowledge to improve neural network robustness. For example, Koh et al. (2020) introduces the concept bottleneck method for embedding interpretable human concepts within neural networks. Although the concept bottleneck shares a similar high-level approach to our paper of imposing explicit structure in neural networks, concept bottleneck models do not consider the problem of controlling when different features are utilized, as we are concerned with. Notably concept bottleneck models represent an ideal setting for our method as the presence of human-interpretable concepts makes providing the annotations we consider in this work feasible. Another approach for incorporating prior human knowledge is explicitly regularizing model saliency maps to align with human annotations (Ross et al., 2017a). In our work, we demonstrate that human annotation of relevant features is particularly essential in the contextual reliability setting. However, we avoid directly regularizing saliency maps in favor of data augmentations due to the fragility of saliency methods observed in prior work (Shah et al., 2021).

3. The Setting of Contextual Reliability

We first formalize relevant background and introduce the setting of contextual reliability. Next, we contrast our setting with prior distribution shift settings.

Preliminaries. We learn predictors that map an input $x \in X$ to some target $y \in Y$ where Y is a discrete set. The target could either be a class label as in supervised learning, or the expert action in imitation learning. We assume access to n sampled training points $\{(x^{(1)}, y^{(1)}), \ldots (x^{(n)}, y^{(n)})\}$. Let $\theta \in \Theta$ parameterize the class of predictors such that $f(x;\theta) \in \mathbb{R}$ and $\ell : \mathbb{R} \times Y \to \mathbb{R}$ is used to compute the loss $\ell(f(x;\theta), y)$ that evaluates the prediction at point x, for parameter θ.

Reliable performance. We are interested in models that work reliably, even under shifts between the train and test distributions. We consider two settings: supervised learning and imitation learning. In the supervised learning setting, we are interested in training robust models that work well across *all* subpopulations in the training data. For example, these partitions could each model different settings like normal driving conditions, slowdowns due to accidents etc. Typically some subpopulations (e.g. normal driving conditions) are more common than others (e.g. accident-induced slowdowns). However, at test time, it is imperative to achieve good performance in *all* subpopulations including the less frequent ones.

Formally, training inputs $\mathbf{z} = (x, y)$ are drawn from a mixture distribution over the set of K latent subpopulations, *i.e.,* $\mathbf{z} \sim P \stackrel{\text{def}}{=} \sum_{k \in [K]} \alpha_k P_k$, where P_k is the distribution over the k^{th} subpopulation. The goal is to control the worst-case performance across all subpopulations:

$$\text{Err}_{\text{rob}}(\theta) \stackrel{\text{def}}{=} \max_{k \in [K]} \mathbb{E}_{(x,y) \sim P_k}[\ell(f(x;\theta), y)], \tag{1}$$

where ℓ is some appropriate loss function.

In the imitation learning setting, distribution shifts naturally arise due to a difference between the train distribution (induced by the expert policy) and test distribution (induced by the learned policy) which often leads to poor test performance of such methods (De Haan et al., 2019). Our metric of interest $\text{Err}_{\text{rob}}(\theta)$ in these imitation learning settings is simply the total reward obtained by the policy induced by θ.

3.1. Background: Prior Robustness Methods

When training deep networks, it is widely observed that simply minimizing the empirical loss on the training data leads to poor performance under subpopulation shifts (Koh et al., 2021; Beery et al., 2021; Zech et al., 2018). Several approaches have been proposed to achieve robust performance under these shifts.

Consider the *optimal robust predictor* defined as follows.

$$\theta^\star_{\text{rob}} \stackrel{\text{def}}{=} \arg\min_{\theta \in \Theta} \text{Err}_{\text{rob}}(\theta). \tag{2}$$

Prior approaches consider different training methods that are aimed at retrieving $\theta^\star_{\text{rob}}$.

Invariance-based approaches. One popular approach to improve robustness is to enforce invariances that are displayed by the optimal robust predictor $\theta^\star_{\text{rob}}$. To do so, it is convenient to think of a model as using various "features" Φ, where each $\phi : X \to \mathbb{R} \in \Phi$ is non-spurious with respect to the optimal robust predictor $\theta^\star_{\text{rob}}$ if $f(x;\theta^\star_{\text{rob}})$ varies as the feature $\phi(x)$ varies. All other features are considered spurious, *i.e.,* $f(x;\theta^\star_{\text{rob}})$ is invariant to spurious features. Some approaches assume knowledge of the spurious features and directly enforce invariance during training via appropriate augmentations (Botev et al., 2022) or regularizing saliency maps (Ross et al., 2017b). Other works address the case where spurious features must be inferred automatically; however, these approaches offer limited gains in practice (Arjovsky et al., 2019; Heinze-Deml et al., 2018; Peters et al., 2016).

Robust optimization approaches. Another family of robust training methods minimize the worst-case loss across subpopulations in the training data (Sagawa et al., 2019b). These methods can be viewed as minimizing the empirical counterpart of the worst-case loss Err_{rob} described in Equation (1). Crucially, these methods require annotating the entire training set with the subpopulation identity.

Targeted data collection. A third family of approaches

for learning reliable models seeks to influence the data collection process such that the empirical risk minimizer over this new distribution is close to $\theta^{\star}_{\text{rob}}$. This usually involves collecting more data from subpopulations that are underrepresented in the original training distribution and requires the ability to either interact with the environment (De Haan et al., 2019; Lyle et al., 2021), or actively query labels for points from an unlabeled pool (Tamkin et al., 2022). Importantly, the success of these methods depends heavily on access to reliable uncertainty estimates (*e.g.,* Tamkin et al. (2022); Lyle et al. (2021)) that inform the collection process.

3.2. Our Setting: Contextual Reliability

In this section, we introduce a novel setting that better captures the nuances of reliable performance in the real world. Next, we compare it to previously studied settings.

We consider the following small twist to the data generation process. Given a discrete set of contexts $\mathsf{C} \stackrel{\text{def}}{=} \{c_1, c_2, \ldots c_k\}$, we first sample a context $\mathrm{c} \sim \mathsf{P}(\mathrm{c})$ (from a categorical distribution over C), and then sample x, y from a distribution $\mathsf{P}_{\mathrm{c}} \stackrel{\text{def}}{=} p((x, \mathrm{y}) \mid \mathrm{c})$. Our goal is now to achieve reliable performance in *all* contexts. Formally, we are interested in the following objective:

$$\text{ConErr}_{\text{rob}}(\theta) \stackrel{\text{def}}{=} \max_{\substack{k \in [K], \\ c \in \mathsf{C}}} \mathsf{E}_{(x,\mathrm{y}) \sim \mathsf{P}_{k,c}} [\ell(f(x;\theta), y)], \quad (3)$$

where $\mathsf{P}_{k,c}$ is the probability distribution over the k^{th} subpopulation in context c, and ℓ is an appropriate loss function.

As motivation, consider an autonomous driving setting with a *next-lane vehicle speed* feature and two contexts indicating the presence/absence of a pedestrian crossing. Across both contexts, agent speed and *next-lane vehicle speed* are generally correlated. However, in the context of no-pedestrian crossing, the slowing of a neighboring vehicle need not imply that the agent should slow, absent of other information. For example, the neighboring vehicle may be preparing to perform a turn or responding to an obstruction in its lane. Suddenly slowing to mimic this vehicle may be unnecessary and expose the agent to the risk of rear-end collisions. At a pedestrian crossing, however, braking of the neighboring vehicle may indicate an unobservable pedestrian entering the intersection and is evidence in itself of the need to stop. Thus, the *optimal robust predictor must rely on different features in different contexts.* Consequently, both existing context-invariant approaches and their context-incorporating extensions will fail to achieve reliable performance.

3.3. The Need to Incorporate Context Information

Without accounting for the context, we find all prior approaches can fail. We support this finding with intuition in this section, analytical proofs in Section 5, and experimental observations in Section 6.

Invariance-based methods train models that use the same set of features in all contexts. For our autonomous vehicle example, this would result in a model that either always uses the next lane speed (with dangerous outcomes when there is no pedestrian crossing and an irrelevant neighboring car slows down) or always ignores the next lane speed (with dangerous outcomes when there is a pedestrian crossing). In the extreme case where every feature is used by the optimal robust predictor in some context, invariance-based methods reduce to standard empirical minimization which is well documented to perform poorly under distribution shifts.

Robust optimization approaches on the other hand, make no assumptions of universal invariance with respect to features. However, they also fail if we do not incorporate context. Minimizing the empirical counterpart of the objective of interest in Equation (3) requires annotations of the context of training points. In the absence of context annotations, we can only minimize the empirical counterpart of Equation (1) which can differ wildly from Equation (3) if the contexts are imbalanced in the training data. Finally, our empirical investigation in Section 6 reveals that uncertainty-based data collection methods also fail to successfully handle multiple contexts. We hypothesize that this is due to the challenging nature of forming high-quality uncertainty estimates when confronted with latent contexts.

4. How to Incorporate Context Information?

In the previous section, we introduced the setting of contextual reliability, where the optimal robust predictor relies on different features in different contexts, and argued that achieving reliable performance requires access to context information. In this section, we explore different ways of collecting and incorporating this information into model training. We start with a natural extension of prior approaches and describe its limitations. We then present our proposed approach and demonstrate it is a viable method for addressing the limitations faced by the baseline method.

4.1. Context Identity Annotations

In order to incorporate context knowledge, we can annotate every training point with its corresponding context. Formally, we annotate each training point $(x^{(i)}, \mathrm{y}^{(i)})$ with $\mathrm{c}^{(i)}$ such that $(x^{(i)}, \mathrm{y}^{(i)}) \sim \mathsf{P}_{\mathrm{c}^{(i)}}$.

Independent Classifier Per-Context (ICC). With this context identity information, one natural baseline is to simply train a separate model (via empirical risk minimization) for each context. At test time, we first predict the context of the input and then use the corresponding predictor. We refer to this method as ICC (independent classifier per-context).

Context-Based Robust Optimization (conDRO). A more sophisticated way to leverage context annotations is via robust optimization. Robust optimization approaches already assume annotations of the subpopulation identities of the training data, where different subpopulations capture partitions of the input space across which we want to obtain good worst-case performance. Equipped with additional context identities of training points, we can partition the training data into $m = |\mathcal{C}| \cdot K$ groups and minimize the worst-case training loss across all m groups. Here, we have K sub-populations for each context and $|\mathcal{C}|$ contexts. This would be the empirical counterpart of our objective of interest in Equation (3). We refer to this method as *conDRO*, an extension of robust optimization with context information.

$$\theta_{\text{conDRO}} \stackrel{\text{def}}{=} \arg\min_{\theta \in \Theta} \max_{\substack{k \in [K], \\ c \in \mathcal{C}}} \mathbb{E}_{(x,y) \sim \hat{P}_{k,c}}[\ell(f(x;\theta), y)], \quad (4)$$

where $\hat{P}_{k,c}$ is the empirical distribution over all training points sampled from $P_{k,c}$. In this work, we propose a new framework for extracting and incorporating information about contexts: Explicit Non-spurious feature Prediction (ENP). We propose to use a different kind of annotation rather than the natural but naive annotation of context identities.

4.2. Explicit Non-Spurious Feature Prediction

Our framework is motivated by looking more carefully at what the optimal robust predictor should do in the contextual reliability setting. Recall that under contextual reliability, the optimal robust predictor relies on different features in different contexts. Therefore, the optimal robust predictor should first infer the context, and then leverage the contextually non-spurious features, while being invariant to the spurious ones. Rather than training a model end-end in some fashion and expecting this structure to emerge due to implicit biases in the training process, we propose to collect context information and explicitly insert this structure into the predictor. As the context affects the optimal predictor solely by determining the set of non-spurious features, we solicit context information in the form of explicit non-spurious feature annotations (formally defined below) instead of context identities.

Feature Annotations. Let $\theta^\star_{\text{rob}}(c)$ denote the optimal robust predictor for context c. A feature $\phi : \mathcal{X} \to \mathbb{R}$ in the set of countable features Φ is non-spurious in context c if the distribution of $f(x; \theta^\star_{\text{rob}}(c))$ is not invariant to the feature values $\phi(x)$ for inputs $x \sim P_c(\cdot)$. Let $N(c)$ denote the set of all such non-spurious features $\phi(x)$ in context c. We propose to annotate training point $x^{(i)}, y^{(i)}$ with the subset of non-spurious features $N^{(i)} = N(c^{(i)})$ where $x^{(i)}, y^{(i)} \sim P_{c^{(i)}}$. We do not require the entire training set to be annotated, only (without loss of generality) the first $n^0 < n$ examples. With these annotations, we propose a two-step methodology:

Step one: Train a feature predictor. Given training data $(x^{(1)}, N^{(1)}), \ldots (x^{(n^0)}, N^{(n^0)})$, we learn a predictor $g : \mathcal{X} \to 2^{\Phi}$ that maps inputs to their corresponding set of non-spurious features, where Φ is the set of all features.

Step two: Train a target model that relies exclusively on predicted non-spurious features. We train a target model that takes as input the pair of original datapoint and its feature annotations (x, N), and returns the prediction $f(x;\theta)$ such that $f(x;\theta)$ is invariant to the spurious features, *i.e.,* all features $\phi \in \Phi \setminus N$. This model is trained on training data comprising $(x^{(i)}, y^{(i)}, N^{(i)})$ for $i = 1, \ldots n^0$ and $(x^{(i)}, y^{(i)}, g(x^{(i)}))$ for $i = n^0 + 1, \ldots n$, where g is the trained feature predictor obtained from step one. In other words, we use the ground-truth feature annotations when they are provided and the *predicted* features annotations on unannotated data points.

At test time, given an input x, we first apply the feature prediction model g to obtain non-spurious features $g(x)$. We then pass $(x, g(x))$ as input to the target model and obtain final predictions. We enforce invariance at both test and training time via augmentations that perturb the values of spurious features (either by adding noise or zeroing them out) such that they cannot be relied upon by the target model. Our core methodological contribution is this two-step process where, rather than training an end-end model, we explicitly induce the structure that different features should be used in different contexts for reliable performance. Next, we discuss the benefits of our proposed method (ENP) over alternatives.

4.3. Benefits of Explicit Non-Spurious Prediction

We identify three axes along which ENP outperforms alternative approaches to incorporating context information.

(1) Annotation cost. Our method only requires non-spurious feature annotations on a subset of the training set and can train a downstream model on the full training set by using *predicted* feature annotations. In contrast, end-end approaches such as conDRO require context annotations to be provided on the entire training set. Across a variety of semi-synthetic and real datasets, we are able to achieve good non-spurious feature prediction accuracy with just a small fraction of the training set annotated.

(2) Annotation feasibility. Often, it is easier for experts to think in terms of the reliability of features for a given input, rather than identifying a dataset-wide partition of points into contexts. It is clear to an expert that next lane speed should be used when they see a pedestrian crossing. However, simply given an input with a pedestrian crossing, it might be hard to know apriori that this corresponds to a distinct context. In recent work, Roelofs et al. (2022) use a

similar rationale to crowdsource the annotations of "causal agents" in a driving scenario (discussed in Section 6).

(3) Preventing overfitting. Even if we allow context annotations on all training points, we find that methods like conDRO fail. This is an issue with the inductive bias of current training algorithms. In the limit of infinite training data, conDRO (Equation (4)) should also minimize our metric of interest (Equation (3)). However, conDRO fails to do so with finite data because it can overfit and fail to learn from minority subpopulations and contexts (Sagawa et al., 2019b; 2020). Furthermore, training such a model end-end suffers from a chicken and egg problem as described in Liu et al. (2021b). In order to learn how to use non-spurious features, the model must have first learned to disambiguate the context. But without an explicit signal to disambiguate contexts, the only signal to disambiguate comes from different features being non-spurious across contexts. The model needs to already know how to use non-spurious features in order to access this signal. Our ENP framework breaks this chicken-egg problem by providing explicit supervision about the set of non-spurious features.

5. Analysing ENP in a Simplified Setup

In a simplified setting that distills contextual reliability, we contrast ENP that uses non-spurious feature annotations with four baselines: (i) IRM (Arjovsky et al., 2019) that learns a context invariant predictor; (ii) ICC where a separate predictor is learned for each context independently; (iii) conDRO (Sagawa et al., 2019b) that optimizes for worst context performance (IRM, conDRO and ICC use context labels); and (iv) ERM which minimizes loss on labeled examples without knowledge of contexts or feature annotations. We show why each baseline performs suboptimally (compared to ENP), either due to its over-conservative nature in learning worst-case robust/invariant predictors or due to its statistical inefficiency caused by failure to share features across contexts. ENP's two stage procedure affords benefits specifically when the non-spurious feature predictor is easier to learn compared to learning the contextually reliable predictor end to end. Details on the data distribution and precise objectives for algorithms are in Appendix A and proofs for our theoretical results are in Appendix B.

Setup. For a binary classification problem with $Y \stackrel{\text{def}}{=} \{-1, 1\}$, the inputs $x = [x_1, x_2, x_3]$ (where, $x_1, x_2, x_3 \in \mathbb{R}^d$) span two contexts $C \stackrel{\text{def}}{=} \{c_1, c_2\}$. The feature annotations for contexts c_1, c_2 are denoted by masks $\mathrm{C}_1, \mathrm{C}_2 \in \{0,1\}^{3\mathrm{d}}$ respectively. For each context, a different set of features is non-spurious: $\{x_1, x_2\}$ in c_1; and $\{x_1\}$ in c_2. Thus, $\mathrm{C}_1^{(\mathrm{j})} = \mathbf{1}(\mathrm{j} \quad 2\mathrm{d})$ and $\mathrm{C}_2^{(\mathrm{j})} = \mathbf{1}(\mathrm{j} \quad \mathrm{d})$ where $\mathrm{C}^{(\mathrm{j})}$ is the j^{th} coordinate for annotation C. For more discussion on the annotations and other details on the data distribution please refer to Appendix A. In this setting, we theoretically analyze estimates returned by ERM, IRM, conDRO, ICC, and ENP for a class of linear predictors $W_1 \stackrel{\text{def}}{=} \{w \in \mathbb{R}^{3d} : \|w\|_2 \quad 1\}$; and empirically evaluate solutions returned when optimizing them over deep nets.

ENP has lower asymptotic errors than conDRO, IRM and ERM. In Theorem 5.1, for linear models, we compare the asymptotic classification errors for all algorithms ($n \to \infty$). We see that both conDRO and IRM yield suboptimal performance (specifically on c_1) because: (i) IRM is only restricted to use the invariant feature x_1, which is less predictive of the label than x_2 in context c_1 when $\gamma \quad 1$; (ii) the conDRO objective enforces its solution to have high but uniform accuracies across both c_1 and c_2, and since any component along x_2 would affect the losses in both contexts in opposite ways, conDRO is forced to forego components along x_2. On the other hand, ENP improves over both since it is allowed to use different features in c_1 (both x_1, x_2) and c_2 (only x_2). The ERM solution relies too heavily on x_2 since this significantly reduces the loss in the majority context c_1, but leads to worse than random performance on c_2. This is because correlations for x_2 are flipped between c_1 and c_2. While it may seem that ERM suffers because W_1 class does not contain a predictor that is uniformly optimal on both contexts, in the subsection that follows we show that similar failures exist even when the model class is more expressive (deep nets). On the other hand, components along x_2 do not effect the predictions on c_2 for ENP since these components are effectively masked by the annotations C_2. The solution for ENP is also comparable to Bayes optimal solutions found by ICC as is evident from corollary 5.2 which follows immediately from Theorem 5.1.

Theorem 5.1 (test accuracies on population data)**.** *For* $\rho_1 \stackrel{\text{def}}{=} \|\mu\|_2 / \sigma^2$, $\rho_2 \stackrel{\text{def}}{=} \|\mu\|_2 / \eta^2$, *and* $\gamma \quad 1$, *given population access, the following test accuracies are afforded by solutions for different optimization objectives over* W_1. *For IRM, conDRO the accuracy* $\{p_c\}$ *is* $0.5\ \mathrm{erfc}(\quad \rho_1)$ *on both* c_1, c_2*; ERM achieves* $0.5 \quad \mathrm{erfc}(\quad \rho_1\sqrt{1+1/\gamma})$ *on* c_1 *and* $0.5\ \mathrm{erfc}(\quad \rho_1(\gamma \quad 1)/\sqrt{\gamma^2+1/\gamma})$ *on* c_2 *as* $p_c \to 1$*; ENP achieves* $\{p_c\} \quad 0.5$ *at least* $0.25 \quad \mathrm{erfc}(\quad \rho_2) \quad \mathrm{erfc}(\quad \rho_1\sqrt{1+1/\gamma})$ *on* c_1 *and* $0.25 \quad \mathrm{erfc}(\quad \rho_2) \quad \mathrm{erfc}(\quad \rho_1/\sqrt{1+1/\gamma^3})$ *on* c_2*; and ICC achieves* $\{p_c\}\ 0.5 \quad \mathrm{erfc}(\quad \rho_1\sqrt{1+1/\gamma})$ *on* c_1 *and* $0.5\ \mathrm{erfc}(\quad \rho_1\sqrt{1+\gamma})$ *on* c_2. *Here,* $\mathrm{erfc}(x) = 2/\sqrt{\pi}\int_x^1 e^{\quad t^2}\mathrm{d}t$.

Corollary 5.2 (Almost Bayes optimality of ENP)**.** *As the feature predictor in the first stage of ENP gets easier to learn* ($\eta \to 0$)*, the ratio of accuracies for ENP solution and Bayes optimal predictor approaches* 1 *on* c_1 *and* $\mathrm{erfc}(\quad \rho_1/\sqrt{1+1/\gamma^3})/\mathrm{erfc}(\quad \rho_1)$ *on* c_2.

ENP is statistically more efficient than ICC when contexts share some features. The distribution of x_1 is identical in both c_1 and c_2. But, recall that the ICC method (which

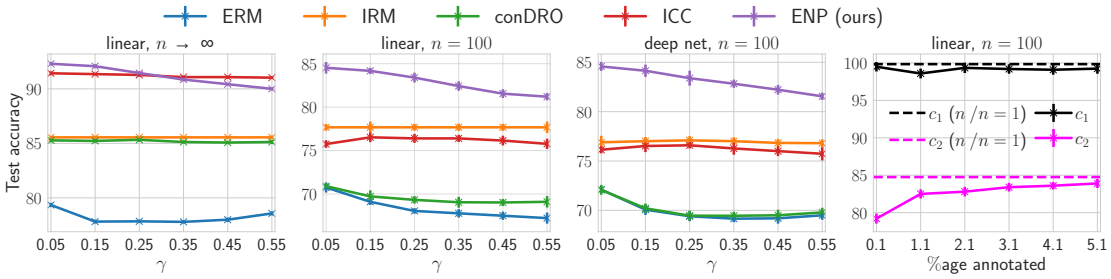

Figure 2. **Empirical evaluations of ENP and baselines (ERM, IRM, ICC, conDRO) on the simplified two-context setup in Section 5:** For each method, we plot the test accuracy averaged over both contexts (balanced average). In (a), (b) each method learns a linear predictor (in W_1), where the asymptotic results ($n \to \infty$) are in (a), and the finite sample results ($n = 100$) are in (b). Similarly, in (c) we compare finite sample results when the target predictor is a non-linear deep network. In all three plots we vary the problem parameter γ, where reducing γ makes the feature x_2 more predictive of the target in c_1 and less predictive in c_2. Finally, in (d) we plot the performance of ENP as we increase the fraction of non-spurious feature annotations in the training set. For each value in the plot, we also report the standard deviation over five independent runs. Details on exact problem parameters for these runs are in Appendix A.

also uses context annotations similar to conDRO) only relies on samples from c_2 to learn a classifier for c_2 (independent of any information from c_1). This can be statistically sub-optimal when the distribution over contexts is skewed and consequently there are much fewer training samples drawn from c_2 (*i.e.,* when $p_c \to 1$). On the other hand, due to explicit annotations, ENP is aware that x_1 is non-spurious in both contexts. In Appendix B.3, we formally show that the generalization error upper bound for ICC is worse than ENP by a factor of $O(1/\sqrt{1 - p_c})$ on context c_2.

5.1. Empirical Results in our Simplified Contextual Reliability Setting

The empirical results in Figure 2(a) show asymptotic errors for the various methods and the results agree with our theoretical findings in Theorem 5.1. First, we see that both IRM and conDRO only learn x_1 and have similar (suboptimal) asymptotic errors on c_1, compared to other methods (*e.g.,* ENP) that also learn feature x_2. Second, ERM has poor test accuracy since it relies too heavily on the more predictive feature x_2 in the majority context c_1, and since x_2 is anti-correlated on minority, ERM's accuracy on c_2 drops below that of random baseline. Additionally, as γ increases we see a slight drop in the performance of ENP since the signal to noise ratio for x_2 decreases in c_1 (where it is used by ENP) and increases in c_2 (where it is ignored). Next, in the finite sample setting, we see the poor performance of ICC in Figure 2(b), since it fails to leverage the shared feature x_1 when trying to learn independent classifiers per-context. The baseline conDRO also overfits on the minority examples from c_2, by memorizing high dimensional noise along x_2, x_3 (Sagawa et al., 2020). On the other hand, compared to ICC and conDRO, for any value of γ, using only finite data ENP achieves performance closer to that of its asymptotic value in Figure 2(a), indicating that it suffers less from finite sample estimation errors (see Theorem B.9 in Appendix B.3). ENP improves over ICC since it can efficiently learn the feature x_1 by using samples from both contexts, and it improves over conDRO since the feature annotation $\mathbf{C}_2$ masks out any noise along x_2, x_3 in context c_2, preventing memorization explicitly. In Figure 2(c), we plot the performance of methods when the model class is one hidden layer deep networks with 512 ReLU activations. Here, even though the model class is expressive enough to contain predictors that are uniformly optimal over both contexts, IRM continues to fail as it enforces invariance whereas conDRO, ERM and ICC suffer more from statistical inefficiencies. Finally, in Figure 2(d) we plot the test accuracy of ENP as the fraction of samples with feature annotations is increased. This improves the performance of the learned feature predictor which in turn improves the test performance for the target predictor, corroborating our results in Corollary 5.2.

6. Experiments

In this section, we study contextual reliability in three settings spanning supervised learning for classification (Corrupted Waterbirds), imitation learning for policies (Noisy Mountain Car), and real-world vehicle trajectory prediction (Waymo Open Motion Dataset). We compare appropriate baselines in each setting to ENP and demonstrate that the theoretical benefits of ENP (Section 5) transfer to practice. Further experimental details can be found in Appendices C and D (for the WOMD).

6.1. Setting One: Corrupted Waterbirds

Setting. We adapt the standard Waterbirds robustness benchmark demonstrated in (Sagawa et al., 2019a) to generate a data-set where the foreground bird images are blurred and randomly cropped with probability 0.05. In this setting, simply relying on the foreground bird images (as done in prior

Table 1. **Corrupted Waterbirds classification accuracies.** We provide the worst-group accuracy for our Corrupted Waterbirds setting, where groups are defined in terms of both the spurious attribute and context. We test methods that don't make use of context: ERM and GroupDRO (groups assigned without context information) as well as methods that use varying amounts of context information: conDRO and GT-Aug. require context information on all training points, while ENP requires only annotations on 10% of the training points.

METHOD	WORST CASE ACC.
ERM	0.67
GDRO	0.60
CONDRO	**0.73**
GT-AUG.	**0.73**
ENP	**0.728**

works) is suboptimal: when the foreground is corrupted, the highly correlated background provides useful information. On the other hand, when the foreground bird is unambiguous, we want to avoid relying on the background as it is not always predictive (for e.g. water birds in land background).

Methods. We test the following methods on Corrupted Waterbirds: ERM, Group DRO (where groups are defined using the spurious/core features (Sagawa et al., 2019a)), conDRO (group definitions are augmented with the ground-truth context), and ENP. We also compare to an oracle version of ENP where we augment according to the ground-truth (rather than predicted) context labels (GT-Aug.). We omit baselines such as Just Train Twice (Liu et al., 2021a) and Learning from Failure (Nam et al., 2020) as they strictly underperform Group DRO. We report the worst group accuracy across contexts (corrupted and clear foreground) in Table 1.

Results. First, we note ERM has poor worst-group accuracy (67%). Standard group DRO (which is state-of-the-art on WaterBirds) actually harms robustness in our setting with a worst-group accuracy of 60%. Thus, we cannot ignore the context structure for improving contextual reliability. Next, we test ENP and compare it to two methods that assume ground-truth context information on all training points: GT-Aug. and conDRO. Even with far less context information (10%), ENP performs comparably to both methods (72.8%). In Appendix E, we test the performance of our feature predictor with varying feature annotation budgets.

6.2. Setting Two: Noisy MountainCar

Setting. We study contextual reliability in imitation learning by extending the setting studied in (De Haan et al., 2019) where adding the previous action to the state causes the policy to underperform due to spurious correlations. In the original setting, it is optimal to always ignore the previous action. However, we hypothesize that when the state is noisy, historical actions can be useful to disambiguate it. To test this, we construct a modified version of the MountainCar environment where noise is added to the velocity in a subset of the state space. Since the optimal MountainCar policy must take different actions at a given x-position depending on which direction it is heading (up or down the slope), making selective use of the previous action is necessary to recover the heading information lost by the state noise.

Table 2. **Noisy MountainCar policy returns.** We show the policy evaluation returns of various imitation learning methods. We consider two standard imitation learning methods with either access to or no access to the previous action (respectively With(out) Prev. Action. We test two baselines that are successful in the universally spurious feature setting (Policy Exec Intervention and Targeted Exploration). Finally, we test conDRO (with ground-truth access to context on all points) and ENP (with ground truth access to feature annotations on 10% of the training data).

METHOD	TEST REWARD	
WITHOUT PREV. ACTION	-170.4	9.7
WITH PREV. ACTION	-194	4.6
POLICY EXEC INTERVENTION	-188.3	7.2
TARGETED EXPLORATION	-195.2	4.2
CONDRO	-188.3	4.26
ENP	**-139.5**	**13.6**

Results for baselines. We first test two approaches based on standard imitation learning: Without Prev. Action assumes no access to the previous action and With Prev. Action allows access to it. As shown in Table 2, both methods perform poorly, demonstrating the insufficiency of universal invariance to the previous action. Next, we consider learning a causal graph of the optimal action through policy execution (De Haan et al., 2019) (Policy Exec Intervention). We find that the learned causal graph often does not contain the previous action despite it being useful on the noisy examples in our setting, resulting in poor policy performance.

Prior work has also considered training an exploration policy to directly visit high uncertainty states and demonstrated the capabilities of this approach when there exists a universal set of reliable features. We test this method (Targeted Exploration) (implementation details in Appendix C) and our results in Table 2 demonstrate the insufficiency of this approach and exemplify the challenges of uncertainty based active-learning in the contextual reliability setting.

Finally we test conDRO in the imitation learning setting, defining groups using the ground-truth context, current action, and previous action. We find that it performs poorly (188.3) and hypothesize that this arises from a mismatch between the conDRO objective and the evaluation metric of policy execution reward.

Results for ENP. We test the ENP framework in the Noisy MountainCar setting. We train a non-spurious feature pre-

Table 3. **Validation minADE on Waymo Open Motion Dataset.** We present the minimum average displacement error (minADE) of various MultiPath++ models, evaluated on a validation set with all agents labeled spurious by humans removed. We compare with two methods that do not use any annotation information: Standard training and Random Augmentations (where agents are randomly deleted during training). Given access to spuriousness annotations, we test a method where only human-annotated samples are augmented (Annotated Augmentations) and ENP.

METHOD	PERTURBED MINADE
STANDARD	0.817
RANDOM AUGMENTATIONS	0.815
ANNOTATED AUGMENTATIONS	0.801
ENP	**0.774**

diction model by subsampling 10% of the training data and providing feature annotations on these points. We use this predictor model to label all training and test states and enforce invariance through augmentations that randomly perturb the previous action (when spurious) by selecting uniformly from all possible actions. ENP's performance (-139.5) outperforms all baselines (best -170.4), showing our method is successful in appropriately making use of the previous action, while not over-relying on it.

6.3. Setting Three: Waymo Open Motion Dataset

Setting. As a preliminary evaluation of ENP on real-world data, we perform experiments using a subset of the Waymo Open Motion Dataset (WOMD) Ettinger et al. (2021) on the task of predicting the future trajectory of an autonomous vehicle. Roelofs et al. (2022) demonstrated that many state-of-the-art models base their predictions on agents that human drivers ignore as spurious and released crowd-sourced agent spuriousness labels (termed causal agent labels) on a subset of the data. For simplicity, we treat road agents as features in this setting. We test all methods on a held-out subset of the annotated data and perturb these samples by deleting all spurious labeled agents. On a large and complex dataset such as WOMD, it can be particularly challenging to a priori specify an appropriate set of contexts, whereas pointwise annotation of spuriousness can be performed easily by human drivers. This makes it impossible to test *conDRO* on WOMD and provides evidence of the enhanced annotation feasibility enjoyed by ENP (see Section 4.3).

Baselines. We compare ENP to two methods that do not make use of feature annotations: a standard-trained MultiPath++ model (Standard) and data augmentations which randomly delete agents during training (Random Augmentations). In addition, we examine a method that incorporates 20% of the ground-truth annotated data into the training set and generates data augmentations that delete spurious-labeled agents (Annotated Augmentations). We see that Standard and Random Augmentations perform comparably (0.817 and 0.815), while Annotated Augmentations result in improved performance (0.801). This further demonstrates the importance of incorporating spuriousness annotations for achieving reliable performance.

ENP: Training a feature predictor. For the first step of ENP, we train a feature predictor by sampling 20% of the annotated samples and training a model to predict the spuriousness of a given agent on this data. We find that we are able to use a much smaller architecture, relative to the full MultiPath++ model and achieve 84.9% performance on the spuriousness prediction task. The relatively small dataset size and simple model used in our method provide evidence of our hypothesis that learning the rules governing the spuriousness of agents is much easier than achieving good performance on the target task (trajectory prediction).

ENP: Training the target model. Using our trained feature predictor model, we predict feature spuriousness labels on all trajectories in our trajectory prediction training set. We implement ENP analogously to Annotation Augmentations, except we are able to generate augmentations on *all* data points using our feature prediction model. ENP achieves a significant improvement (0.774) over Annotated Augmentations (0.801), providing evidence of ENP's efficacy in improving model reliability with limited access to spuriousness annotations.

7. Conclusion

We introduce and study a new setting of contextual reliability where it is optimal to rely on different features in different contexts. This captures several realistic settings and introduces new challenges to robust machine learning. Our theory and experiments show that methods that do not incorporate context information struggle to improve contextual reliability. Incorporating context boils down to eliciting information from an expert about the latent context. We propose and advocate for a framework that uses explicit annotations of the non-spurious features for a small fraction of the training data. The success of our method relies on two ingredients. The first is the ability to effectively annotate non-spurious features. As representation learning methods improve via large-scale pretraining, it is an interesting future direction to consider annotations in terms of higher-level learned features. The second ingredient is the ability to successfully learn a high-quality predictor that maps inputs to non-spurious features. We provide evidence here that this is indeed already possible for the real-world setting of motion prediction in driving. We believe an exciting line of future work is to consider even more complex context-prediction scenarios perhaps by allowing for test-time interventions with an expert.

References

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

Beery, S., Agarwal, A., Cole, E., and Birodkar, V. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/b571ecea16a9824023ee1af16897a582-Paper.pdf.

Botev, A., Bauer, M., and De, S. Regularising for invariance to data augmentation improves supervised learning. *arXiv preprint arXiv:2203.03304*, 2022.

Brown, D., Niekum, S., and Petrik, M. Bayesian robust optimization for imitation learning. *Advances in Neural Information Processing Systems*, 33:2479–2491, 2020.

De Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Duchi, J., Khosravi, K., and Ruan, F. Multiclass classification, information, divergence and surrogate risk. *The Annals of Statistics*, 2018.

Duchi, J. C., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *Under review*, 2:1, 2019.

Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C. R., Zhou, Y., et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

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

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization, 2020.

Heinze-Deml, C., Peters, J., and Meinshausen, N. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018.

Hu, W., Niu, G., Sato, I., and Sugiyama, M. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pp. 2029–2037. PMLR, 2018.

Javed, Z., Brown, D. S., Sharma, S., Zhu, J., Balakrishna, A., Petrik, M., Dragan, A., and Goldberg, K. Policy gradient bayesian robust optimization for imitation learning. In *International Conference on Machine Learning*, pp. 4785–4796. PMLR, 2021.

Kakade, S. M., Sridharan, K., and Tewari, A. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Advances in neural information processing systems*, 21, 2008.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models, 2020. URL https://arxiv.org/abs/2007.04612.

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

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Liu, A. and Ziebart, B. Robust classification under sample selection bias. *Advances in neural information processing systems*, 27, 2014.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information, 2021a. URL https://arxiv.org/abs/2107.09044.

Liu, E. Z., Raghunathan, A., Liang, P., and Finn, C. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International conference on machine learning*, pp. 6925–6935. PMLR, 2021b.

Lyle, C., Zhang, A., Jiang, M., Pineau, J., and Gal, Y. Resolving causal confusion in reinforcement learning via robust exploration. In *Self-Supervision for Reinforcement Learning Workshop-ICLR 2021*, 2021.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In

International conference on machine learning, pp. 10–18. PMLR, 2013.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: Training debiased classifier from biased classifier, 2020. URL https://arxiv.org/abs/2007.02561.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

Roelofs, R., Sun, L., Caine, B., Refaat, K. S., Sapp, B., Ettinger, S., and Chai, W. Causalagents: A robustness benchmark for motion forecasting using causal relationships, 2022. URL https://arxiv.org/abs/2207.03586.

Rosenfeld, E., Ravikumar, P., and Risteski, A. The risks of invariant risk minimization, 2021.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. Right for the right reasons: Training differentiable models by constraining their explanations. *CoRR*, abs/1703.03717, 2017a. URL http://arxiv.org/abs/1703.03717.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017b.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *CoRR*, abs/1911.08731, 2019a. URL http://arxiv.org/abs/1911.08731.

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

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Shah, H., Jain, P., and Netrapalli, P. Do input gradients highlight discriminative features?, 2021. URL https://arxiv.org/abs/2102.12781.

Tamkin, A., Nguyen, D., Deshpande, S., Mu, J., and Goodman, N. Active learning helps pretrained models learn the intended task, 2022. URL https://arxiv.org/abs/2204.08491.

Tien, J., He, J. Z.-Y., Erickson, Z., Dragan, A., and Brown, D. S. Causal confusion and reward misidentification in preference-based reward learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Torralba, A. Contextual priming for object detection. *International journal of computer vision*, 53:169–191, 2003.

Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K. S., Nayakanti, N., Cornman, A., Chen, K., Douillard, B., Lam, C. P., Anguelov, D., and Sapp, B. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction, 2021. URL https://arxiv.org/abs/2111.14973.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

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

Appendix Outline

A. Additional details for the setup in Section 5.

In this section we provide details on the setup used for our analysis in section 5. We begin by describing the data distribution and non-spurious feature annotations for each context. Then, we provide details on the various objectives we theoretically and empirically analyze in this setup.

Data distribution. The data distribution P (over X Y) for our setup is described in (5) where p_c is the probability of choosing the context c_1 (majority context when p_c 0.5), η is the signal to noise ratio that controls the hardness of learning the non-spurious feature predictor, and γ 1 controls the signal to noise ratio (hardness of learning) for feature x_2 over x_1 in c_1, and x_1 over x_2 in c_2. Note, that this setup distills contextual reliability in the sense that the feature x_2 is much more useful in predicting the label in context c_1 (over c_2), and the invariant feature x_1 is predictive of the label to the same degree in both contexts.

$$
\begin{aligned}
&\text{For } \mu \in \mathsf{R}^d, \text{ context } \mathrm{c} = c_1 \text{ with prob. } p_c, \text{ and } \mathrm{y} \quad \mathrm{Unif}(\{-1, 1\}),\\
&\qquad x_1 \mid \mathrm{y} \quad \mathcal{N}(\mu\mathrm{y},\ \sigma^2\mathbf{I}_d)\\
&\qquad x_2 \mid (\mathrm{y},\ \mathrm{c} = c_1) \quad \mathcal{N}(\mu\mathrm{y},\ \gamma^2\sigma^2\mathbf{I}_d)\\
&\qquad x_2 \mid (\mathrm{y},\ \mathrm{c} = c_2) \quad \mathcal{N}(-\mu\mathrm{y},\ 1/\gamma^2\ \sigma^2\mathbf{I}_d)\\
&\qquad x_3 \mid \mathrm{c} = c_1 \quad \mathcal{N}(\mu,\ \eta^2\mathbf{I}_d)\\
&\qquad x_3 \mid \mathrm{c} = c_2 \quad \mathcal{N}(-\mu,\ \eta^2\mathbf{I}_d)
\end{aligned} \tag{5}
$$

Models. For the theoretical analysis we restrict ourselves to a linear model class. In Section 5 we also have experimental results with deep networks. We use $\mathcal{W}_1$ to denote the class of linear predictors that are bounded in l_2 norm: $\mathcal{W}_1 \stackrel{\text{def}}{=} \{x \mapsto w^\top x : w \in \mathsf{R}^{3d},\ \|w\|_2 \leq 1\}$. A label classifier that is used to predict the task label is evaluated using the loss function $\ell : \mathsf{R} \times \mathcal{Y} \mapsto \mathsf{R}$ evaluates classifiers $w \in \mathcal{W}_1$, and is a surrogate loss for the $\ell_{0/1}$ error where $\ell_{0/1}(w^\top x, \mathrm{y}) = \mathbf{1}(\mathrm{sgn}(w^\top x) = \mathrm{y})$. On the other hand, a classifier that is used to predict the context of a particular input is evaluated using the loss function $\ell' : \mathsf{R} \times \mathcal{C} \to \mathsf{R}$. For the theory, both the label classifier and the non-spurious feature predictor are restricted to the linear class $\mathcal{W}_1$. For experiments with deep nets, we set the deep network to be a one-hidden layer ReLU network with 512 activations.

Labels and annotations. From the distribution P above, we are given an *i.i.d.* sampled dataset $\hat{\mathsf{P}} \stackrel{\text{def}}{=} \{(x^{(i)}, \mathrm{y}^{(i)})\}_{i=1}^n$ P^n. When clear from context, we will also use $\hat{\mathsf{P}}$ to denote to denote the empirical distribution over the sampled data. The context conditional distribution $(x, \mathrm{y}) \mid \mathrm{c}$ is denoted by P_c. For ENP (our method) we assume access to an *iid* subset of n' n samples for which we have the feature annotations. We have two annotations C_1 and C_2 for c_1 and c_2 respectively. The annotation is as follows. The j^{th} co-ordinate of the feature annotation $\mathrm{C}^{(\mathrm{j})} = \mathbb{1}(\mathrm{j}$ $2\mathrm{d})$ if $\mathrm{c} = c_1$, and $\mathrm{C}^{(\mathrm{j})} = \mathbb{1}(\mathrm{j}$ $\mathrm{d})$ if $\mathrm{c} = c_2$. This is because, when γ 1 is small, both x_1 and x_2 are predictive of the label in c_1, whereas only x_1 is predictive in c_2 since the signal-to-noise ratio is poor for x_2 in c_2. For the conDRO baseline, we assume access to context labels c (but not feature annotations). For label classification, we use the exponential loss: $\ell(z, \mathrm{y}) = \exp(-z\ \mathrm{y})$ where $\mathrm{y} \in \{-1, 1\}$ and $z \in \mathsf{R}$. For context classification, we also use an exponential loss but one that now treats context c_1 as label $+1$ and context c_2 as label -1, *i.e.,* $\ell'(z, \mathrm{c}) = \exp(-z\ \mathbf{1}(\mathrm{c} = c_1) + z\ \mathbf{1}(\mathrm{c} = c_2))$.

Algorithms. The goal of our analysis is to compare the performance of conDRO, ERM, and IRM with ENP by analyzing the asymptotic error for the solution found by each method, and also its statistical efficiency. Here, we write the objectives for linear predictors. For non-linear functions, the map $x \mapsto w^\top x$ is replaced with a deep neural network: $f : \mathcal{X} \mapsto \mathsf{R}$.

First, we begin with the ERM and IRM objectives that uses no other auxiliary information apart from the label for each example. The former minimizes average loss using all the features in the input, while the latter does the same only using the invariant feature (across contexts), which is x_1.

$$
\text{ERM:} \quad \min_{w \in \mathcal{W}_1} \mathsf{E}_\mathsf{P}\, \ell(w^\top x,\ \mathrm{y}) \tag{6}
$$

$$
\text{IRM:} \quad \min_{w \in \mathcal{W}_{1,\mathrm{irm}}} \mathsf{E}_\mathsf{P}\, \ell(w^\top x,\ \mathrm{y}), \tag{7}
$$

where $\mathcal{W}_{1,\mathrm{irm}}$ is the class of norm bounded linear predictors that only use feature x_1 *i.e.,*, $\mathcal{W}_{1,\mathrm{irm}} \stackrel{\text{def}}{=} \{w \in \mathcal{W}_1 : w = [w^0, \mathbf{0}_d, \mathbf{0}_d]\}$ ($w^0 \in \mathsf{R}^d$ and $\mathbf{0}_d$ is a d-dimensional vector of 0s).

Next, we consider objectives that use context information (in addition to labels): (i) conDRO: optimizes for the worst performance across contexts; (ii) ICC: learns a different classifier for each context using only samples drawn from that context. Note the ICC learns Bayes optimal predictors for each context and thus has the lowest asymptotic errors.

$$\text{conDRO:} \quad \min_{w \in W_1} \max_{c \in C} \mathsf{E}_{P_c} \ell(w^\top x, \mathrm{y}) \tag{8}$$

$$\text{ICC:} \quad \min_{\substack{w_1, w_2 \\ \in W_1}} \mathsf{E}_{P_{c_1}} \ell(w_1^\top x, \mathrm{y}) + \mathsf{E}_{P_{c_2}} \ell(w_2^\top x, \mathrm{y}) \tag{9}$$

Finally, we describe ENP that learns two predictors: (i) non-spurious feature predictor that predicts the context (and consequently the corresponding annotation) for each test example; (ii) the label predictor

$$\text{ENP (target predictor)}: \quad \min_{w \in W_1} p_c \;\; \mathsf{E}_P \, \ell\left(w^\top (\mathrm{C}_1 \;\; \mathrm{x})\right), \mathrm{y}) + (1 \;\; p_c) \;\; \mathsf{E}_P \, \ell\left(w^\top (\mathrm{C}_2 \;\; \mathrm{x})\right), \mathrm{y}), \tag{10}$$

$$\text{ENP (feature predictor)}: \quad \min_{g \in W_1} \mathsf{E}_P \, \ell^{0}(g^\top x, \mathrm{c}), \tag{11}$$

where, represents Hadamard product.

Note, that for conDRO we use context annotations to optimize for the worst context performance, and since this is clearly more optimal for contextual reliability (over traditional group DRO methods that do not use context information), this is the only DRO baseline we analyze. Subsequently, we shall see why even this strategy can be inefficient at learning the optimal robust predictor for each context. In the IRM objective, we restrict optimization over linear predictors that make predictions solely using x_1, the only feature whose class distribution is invariant across both contexts (for each label).

B. Omitted proofs and formal statements for the analysis in Section 5

In this section, we provide proofs for our theorem statements in Section 5 of the main paper. We also provide formal discussion on the generalization results for ENP and ICC.

B.1. Proof for Theorem 5.1

In this subsection, we prove claims regarding the asymptotic errors attained by solving population versions of the objectives in Section A, when the model class is linear (W_1). We look at each objective separately, but before that we introduce the following two lemmas on optimal linear target predictors for each context, and accuracies on each context.

Lemma B.1 (optimal linear predictors for c_1, c_2)**.** *The linear predictor in W_1 with the least $\ell_{0/1}$ error for context c_1 is* $1/(k\mu k_2 \sqrt{1+\gamma^2})$ $[\mu\gamma, \mu, \mathbf{0}_d]$*, and for context c_2 is* $1/(k\mu k_2 \sqrt{1+\gamma^2})$ $[\mu, \;\; \mu\gamma, \mathbf{0}_d]$*. Here $\ell_{0/1}$ is the 0-1 loss: $\ell_{0/1}(z, \mathrm{y}) = \mathbb{1}(\mathrm{sgn}(z) = \mathrm{y})$.*

Proof. For Gaussian data with the same covariance matrices for class conditionals $\mathsf{P}(x \mid \mathrm{y} = 1)$ and $\mathsf{P}(x \mid \mathrm{y} = \;\; 1)$, the Bayes decision rule is given by the Fisher's linear discriminant direction (Chapter 4; Bishop (2006)):

$$h(x) = \begin{cases} 1, & \text{if } h^\top x > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $h = 2 \;\; 1/\sigma^2 \, [\mu, \mu/\gamma, \mathbf{0}_d]$ for context c_1, and $h = 2 \;\; 1/\sigma^2 \, [\mu, \;\; \gamma\mu, \mathbf{0}_d]$ for context c_2 (using the covariance matrices from the data distribution for each context). Here, $\mathbf{0}_d$ is a d dimensional vector of 0s. Since, the direction of h solely determines the $\ell_{0/1}$ error of the predictor, the optimal linear predictors in W_1 are obtained by dividing them both by their corresponding norms. □

Lemma B.2 (per-context accuracy)**.** *The accuracy of predictor $w = [w_1, w_2, \mathbf{0}_d] \;\; 2 \;\; W_1$ on context c_1 is* $0.5 \; \mathrm{erfc}\left(\frac{-p}{\sigma} \frac{(w_1 + w_2)^\top \mu}{2(k w_1 k_2^2 + \gamma k w_2 k_2^2)} \right)$ *and on context c_2 is* $0.5 \;\; \mathrm{erfc}\left(\frac{-p}{\sigma} \frac{(w_1 \;\; w_2)^\top \mu}{2(k w_1 k_2^2 + 1/\gamma k w_2 k_2^2)} \right)$*, where* $\mathrm{erfc}(x) = 2/\sqrt{\pi} \int_x^1 e^{\;\; t^2} \mathrm{d}t$.

Proof. Let P_{c_1} be the probability distribution for context c_1, and P_{c_2} be the distribution for c_2. Let z_1 and z_2 be random variables distributed as $N(\mathbf{0}_d, \sigma^2 \mathbf{I}_d)$. Then the accuracy on context c_1 is,

$$\begin{aligned}
\mathsf{P}_{c_1}(\operatorname{sgn}(w^\top x) = y) &= \mathsf{P}_{c_1}(\operatorname{sgn}(w^\top x)\, y > 0) \\
&= \mathsf{P}_{c_1}(y w_1^\top \mu + y w_2^\top \mu + y w_1^\top z_1 + y w_2^\top z_2 > 0) \\
&= \mathsf{P}(\tilde{z} > 0) \\
&= \mathsf{P}(z > -\tilde{\mu}/\tilde{\sigma}) \\
&= \mathsf{P}(z > -\tilde{\mu}/\tilde{\sigma}) \\
&= 0.5\ \operatorname{erfc}(-\tilde{\mu}/\sqrt{2}\tilde{\sigma}),
\end{aligned}$$

where z is distributed as standard Gaussian, and $\tilde{z}$ is a Gaussian random variable with mean $\tilde{\mu} \stackrel{\text{def}}{=} \mu^\top (w_1 + w_2)$ and variance $\tilde{\sigma}^2 \stackrel{\text{def}}{=} (\|w_1\|^2 + \gamma\ \|w_2\|_2^2)\sigma^2$. The last equality uses the definition of the $\operatorname{erfc}(\ )$ function. The calculation for accuracy on c_2 remains the same except now $\tilde{\mu} \stackrel{\text{def}}{=} \mu^\top (w_1 - w_2)$, and $\tilde{\sigma}^2 \stackrel{\text{def}}{=} (\|w_1\|^2 + 1/\gamma\ \|w_2\|_2^2)\sigma^2$. □

Lemma B.3 (solutions lie in a low dimensional subspace)**.** *For ERM conDRO, and ENP, their corresponding solutions would belong to the set* $U \stackrel{\text{def}}{=} \{\lambda_1\ [\mu, \mathbf{0}_d, \mathbf{0}_d] + \lambda_2\ [\mathbf{0}_d, \mu, \mathbf{0}_d] : \lambda_1^2 + \lambda_2^2 = 1\}$.

Proof. First, we will show that the component along x_3 will be 0. Let's say the component along x_i is w_i. Then, for any context, the conditional variance $\mathrm{V}[\mathrm{y}(\mathrm{w}^\top \mathrm{x}) \mid \mathrm{y}]$, denoted as σ_0^2 is $\sigma_0^2 \stackrel{\text{def}}{=} w_1^\top \mathrm{V}[x_1 \mid \mathrm{y}] w_1 + w_2^\top \mathrm{V}[x_2 \mid \mathrm{y}] w_2 + w_3^\top \mathrm{V}[x_3 \mid \mathrm{y}] w_3$, and the mean is $\mu_0 \stackrel{\text{def}}{=} w_1^\top \mu + \mathbb{1}(\mathrm{c} = c_1) w_2^\top \mu - \mathbb{1}(\mathrm{c} = c_2) w_2^\top \mu$. Here, $\mathrm{V}[x] \in \mathbb{R}^{d \times d}$ is a positive semidefinite covariance matrix. For any context c_1 or c_2, the per-context accuracy improves as σ_0 decreases (as per Lemma B.2) without changing μ_0. This is true when $\|w_3\|_2$ decreases monotonically. Since the loss is classification calibrated, the loss also decreases monotonically as $\|w_3\|_2$ decreases. Hence, the optimal solution would necessarily have $w_3 = \mathbf{0}_d$.

Next, we consider the component along x_1 and assume $x_1 = \alpha_1\ \mu + \alpha_2\ v$. Assume that for the solutions of ERM, ENP and conDRO: $\omega_1, \omega_2 \neq 0$ and $v^\top \mu = 0$. The component $\alpha_2\ v$ will contribute to σ_0^2 with the additive term $\alpha_2^2 \sigma^2 \|v\|_2^2$, without having any effect on μ_0. This means that we can improve the accuracy for both contexts (reduce loss ℓ) further by reducing α_2. This contradicts the assumption that $\alpha_2 \neq 0$ for the solutions of ERM, conDRO and ENP. Thus, for all the objectives the solution would not have any component in the null space of μ along x_1. Similar argument can be used to prove that the component along the null space μ would be zero for x_2 as well.

Combining the above two arguments on the component along x_3 and components along null space of μ for x_1, x_2 we can conclude that the solutions for ERM, conDRO and ENP would necessarily lie in the two rank subspace U. □

Now, we are ready to start the proof of Theorem 5.1, and for the benefit of the reader we shall first restate the theorem statement.

Theorem B.4 (test accuracies on population data (restated))**.** *For* $\rho_1 \stackrel{\text{def}}{=} \|\mu\|_2/\sqrt{2}\sigma$, $\rho_2 \stackrel{\text{def}}{=} \|\mu\|_2/\sqrt{2}\eta$, *and* $\gamma \geq 1$, *given population access, the following test accuracies are afforded by solutions for different optimization objectives over* W_1. *For IRM, conDRO the accuracy* δp_c *is* $0.5 \operatorname{erfc}(\rho_1)$ *on both* c_1, c_2*; ERM achieves* $0.5 \operatorname{erfc}(\rho_1 \sqrt{1 + 1/\gamma})$ *on* c_1 *and* $0.5 \operatorname{erfc}(\rho_1(\gamma - 1)/\sqrt{\gamma^2 + 1/\gamma})$ *on* c_2 *as* $p_c > 1$*; ENP achieves* $\delta p_c \geq 0.5$ *at least* $0.25 \operatorname{erfc}(\rho_2) \operatorname{erfc}(\rho_1 \sqrt{1 + 1/\gamma})$ *on* c_1 *and* $0.25 \operatorname{erfc}(\rho_2) \operatorname{erfc}(\rho_1/\sqrt{1 + 1/\gamma^3})$ *on* c_2*; and ICC achieves* δp_c $0.5 \operatorname{erfc}(\rho_1 \sqrt{1 + 1/\gamma})$ *on* c_1 *and* $0.5 \operatorname{erfc}(\rho_1 \sqrt{1 + \gamma})$ *on* c_2. *Note that* $\operatorname{erfc}(x) > 2$ *as* $x < 0$ *and* < 1 *since* $\operatorname{erfc}(x) = 2/\sqrt{\pi} \int_x^\infty e^{-t^2} \mathrm{d}t$.

Proof. We start with the easier cases of IRM and ICC where we directly use results from the above two lemmas. Then we shall look at conDRO and ERM where we need to deal with mixture of per-context losses. Finally, we look at ENP, where we need to analyze both feature and target predictors.

IRM. Recall that the $W_{\mathsf{irm},1}$ class only consists of unit norm bounded predictors along attribute x_1. Since the exponential loss is a surrogate (Duchi et al., 2018), the predictor minimizing the exponential loss ℓ is also the one with the highest 0-1 accuracy. Thus, we can use similar arguments as in Lemma B.1 to conclude that the optimal predictor is $\mu/\|\mu\|_2$, and from arguments similar to the ones in Lemma B.2 we can conclude that the target accuracy $0.5 \operatorname{erfc}(\|\mu\|_2/\sqrt{2}\sigma) = 0.5 \operatorname{erfc}(\rho_1)$.

ICC. Since the exponential loss ℓ is classification calibrated, the minimizer of this loss on c_1 and c_2 individually also has the least $\ell_{0/1}$ error in W_1, which is exactly the predictor defined in Lemma B.1. Directly applying Lemma B.2 on this predictor, with $w_1 = \mu\gamma/k_\mu k_2 \sqrt{1+\gamma^2}$ and $w_2 = \mu/k_\mu k_2 \sqrt{1+\gamma^2}$ we conclude that test accuracy for ICC predictor on c_1 is $0.5 \ \ \mathrm{erfc}\left(\ (w_1+w_2)^\top \mu/\sigma \sqrt{2(k w_1 k_2^2 + \gamma k w_2 k_2^2)}\right)$. Similarly, with $w_1 = \mu/k_\mu k_2 \sqrt{1+\gamma^2}$ and $w_2 = \ \mu\gamma/k_\mu k_2 \sqrt{1+\gamma^2} = 0.5 \ \ \mathrm{erfc}(\ \rho_1 \sqrt{1/\gamma + 1})$, the test accuracy of ICC on c_2 is $0.5 \ \ \mathrm{erfc}\left(\ (w_1 \ \ w_2)^\top \mu/\sigma \sqrt{2(k w_1 k_2^2 + 1/\gamma k w_2 k_2^2)}\right) = 0.5 \ \ \mathrm{erfc}(\ \ \rho_1 \sqrt{1+\gamma})$.

conDRO. From Lemma B.3 we know that the solution for conDRO is of the form $\lambda_1^\star \ v_1 + \lambda_2^\star \ v_2$ where $v_1 \stackrel{\text{def}}{=} [\mu, \mathbf{0}_d, \mathbf{0}_d]$ and $v_2 \stackrel{\text{def}}{=} [\mathbf{0}_d, \mu, \mathbf{0}_d]$. Recall that $\rho_1 \stackrel{\text{def}}{=} k_\mu k_2/\sqrt{2}\sigma$. Since the exponential loss is classification calibrated and $(\lambda_1^\star)^2 + (\lambda_2^\star)^2 = 1$, we can say that:

$$
\begin{aligned}
\lambda_1^\star \ \ 2 \ \ \arg \inf_{\substack{\lambda_1 2[\ \ 1,1], \\ \lambda_2^2 = 1 \ \ \lambda_1^2}} \ \max\left(\mathsf{E}_{\mathsf{P}_{c_1}} \ell(\lambda_1 \mu^\top x_1 + \lambda_2 \mu^\top x_2, \mathrm{y}), \mathsf{E}_{\mathsf{P}_{c_2}} \ell(\lambda_1 \mu^\top x_1 + \lambda_2 \mu^\top x_2, \mathrm{y})\right) \\
= \ \arg \sup_{\substack{\lambda_1 2[\ \ 1,1], \\ \lambda_2^2 = 1 \ \ \lambda_1^2}} \ \min\left(\mathrm{erfc}\left(\ \ \rho_1 \ \ \frac{\lambda_1 + \lambda_2}{\sqrt{\lambda_1^2 + \gamma\lambda_2^2}}\right), \mathrm{erfc}\left(\ \ \rho_1 \ \ \frac{\lambda_1 \ \ \lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2/\gamma}}\right)\right) \\
= \ \arg \sup_{\lambda_1 2[\ \ 1,1]} \ \min\left(\mathrm{erfc}\left(\ \ \rho_1 \ \ \frac{\lambda_1 \ \ \sqrt{1 \ \ \lambda_1^2}}{\sqrt{\lambda_1^2 + \gamma(1 \ \ \lambda_1^2)}}\right), \mathrm{erfc}\left(\ \ \rho_1 \ \ \frac{\lambda_1 \ \ \sqrt{1 \ \ \lambda_1^2}}{\sqrt{\lambda_1^2 + (1 \ \ \lambda_1^2)/\gamma}}\right)\right)
\end{aligned}
$$

Note that to minimize $\mathrm{erfc}(\ )$ terms we need to increase the value of c when the terms are of the form $\mathrm{erfc}(\ \ \rho_1 \ \ c)$. Thus it is clear that $\lambda_1^\star > 0$. Further, since $\gamma \ \ \ 1$, we also know that $\lambda_1^2 + \gamma(1 \ \ \ \lambda_1^2) < \lambda_1^2 + (1/\gamma) \ \ (1 \ \ \ \lambda_1^2)$. Thus, if assume that $\lambda_2^\star \ \ \ 0$, then the optimal value is $\lambda_1^\star = 1$ and $\lambda_2^\star = 0$. On the other hand, if we assume that $\lambda_2^\star < 0$, then the minimum of the erfc terms is clearly lower than $\mathrm{erfc}(\ \ \rho_1)$, which would be the value of the above objective at $\lambda_1^\star = 1$. Therefore, we conclude that $\lambda_1^\star = 1, \lambda_2^\star = 0$, which yields the following solution for conDRO: $[\mu/k_\mu k_2, \mathbf{0}_d, \mathbf{0}_d]$. From Lemma B.2 we know that on both contexts this solution has accuracy $0.5 \ \ \mathrm{erfc}(\ \ \rho_1)$ which also matches the performance of IRM.

ERM. Once again because of classification calibrated losses, and Lemma B.3, similar to conDRO, we can re-write the ERM problem as the following optimization objective:

$$
\inf_{\lambda_1 2[\ \ 1,1], \lambda_2^2 = 1 \ \ \lambda_1^2} \ p_c \ \ \mathsf{E}_{\mathsf{P}_{c_1}} \ell(\lambda_1 \mu^\top x_1 + \lambda_2 \mu^\top x_2, \mathrm{y}) \ + \ (1 \ \ \ p_c) \ \ \mathsf{E}_{\mathsf{P}_{c_2}} \ell(\lambda_1 \mu^\top x_1 + \lambda_2 \mu^\top x_2, \mathrm{y})
$$

Since, for every p_c we can construct a Cauchy sequence of $\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(n)}$ that converges uniformly to the $\arg\inf$ at the given value of p_c, we can apply Moore-Osgood theorem. The following interchanges limits and finds the solution for ERM at $p_c \ \ ! \ \ 1$.

$$
\lambda_1^\star \ \ 2 \ \ \arg \inf_{\substack{\lambda_1 2[\ \ 1,1], \\ \lambda_2^2 = 1 \ \ \lambda_1^2}} \ \mathsf{E}_{\mathsf{P}_{c_1}} \ell(\lambda_1 \mu^\top x_1 + \lambda_2 \mu^\top x_2, \mathrm{y})
$$

Now, using Lemma B.1 we get $\lambda_1^\star = \gamma/k_\mu k_2 \sqrt{1+\gamma^2}$ and $\lambda_2^\star = 1/k_\mu k_2 \sqrt{1+\gamma^2}$. Then, we finally apply Lemma B.2 to conclude that the accuracy of ERM solution as $p_c \ \ ! \ \ 1$ is $0.5 \ \ \mathrm{erfc}(\ \ \rho_1 \sqrt{1 + 1/\gamma})$ on c_1 and $0.5 \ \ \mathrm{erfc}(\ \ \rho_1(\gamma \ \ 1)/\sqrt{\gamma^2 + 1/\gamma})$ on c_2. □

ENP. Recall that $\rho_2 \stackrel{\text{def}}{=} k_\mu k_2/\sqrt{2}\eta$. Thus, using arguments similar to the ones in the proof of Lemma B.1, the optimal context predictor in W_1 will have accuracy of $0.5 \ \ \mathrm{erfc}(\ \ \rho_2)$ on the context prediction problem. Here, we treated the context prediction problem as binary classification with balanced context labels. We can always do this since we have population access to P_{c_1} and P_{c_2}, and thus we can upsample the examples from the minority context. In our simplified setting, the feature predictor is given directly by the context predictor since each context maps to a unique annotation.

Now, to train the target predictor we use ground truth annotations, and given population access we assume that each data point also has the corresponding ground truth annotation, *i.e.,* if the datapoint is from context c_1, then the annotation is C_1, else it is C_2. Consequently, using Lemma B.3 and given classification calibrated exponential loss, we can rewrite the optimization problem for ENP as:

$$
\sup_{\substack{\lambda_1 2[\ \ 1,1], \\ \lambda_2^2 = 1 \ \ \lambda_1^2}} \ p_c \ \ \mathrm{erfc}\left(\ \ \rho_1 \ \ \frac{\lambda_1 + \lambda_2}{\sqrt{\lambda_1^2 + \gamma\lambda_2^2}}\right) \ + \ (1 \ \ \ p_c) \ \ \mathrm{erfc}\left(\ \ \rho_1 \ \ \frac{\lambda_1}{\sqrt{\lambda_1^2 + \lambda_2^2/\gamma}}\right)
$$

For all $\lambda_2 > 0$, we know that $\lambda_1+\lambda_2/\sqrt{\lambda_1^2+\gamma\lambda_2^2} > \lambda_1/\sqrt{\lambda_1^2+(1/\gamma)\lambda_2^2}$, when γ 1. When p_c 0.5, then p_c erfc(ρ_1 $\lambda_1+\lambda_2/\sqrt{\lambda_1^2+\gamma\lambda_2^2}$) (1 p_c) erfc(ρ_1 $\lambda_1/\sqrt{\lambda_1^2+\lambda_2^2/\gamma}$) for all values of λ_1 2 $[0,1]$. Thus, from Lemma B.1 we conclude that $\lambda_1^\star = 1/k\mu k_2\sqrt{1+\gamma^2}\,[\mu\gamma, \mu, \mathbf{0}_d]$. When we have perfect ground truth annotations, then plugging this value into the equation we above, we find that the accuracy on c_1 is 0.5 erfc$\left(\rho_1\sqrt{1+1/\gamma}\right)$ and on c_2 is 0.5 erfc$\left(\frac{\rho_1}{\sqrt{1+1/\gamma^3}}\right)$.

At test time, when we do not have perfect feature annotations on each input, we use the trained feature predictor which has an accuracy of 0.5 erfc(ρ_2). Thus the accuracy of ENP on c_1 is (0.5 erfc(ρ_2)) $\left(0.5 \text{ erfc}\left(\rho_1\sqrt{1+1/\gamma}\right)\right)$ = 0.25 erfc(ρ_2) erfc$\left(\rho_1\sqrt{1+1/\gamma}\right)$. Similarly on c_2 it is (0.5 erfc(ρ_2)) (0.5 erfc($\rho_1/\sqrt{1+1/\gamma^3}$)) = 0.25 erfc(ρ_2) erfc($\rho_1/\sqrt{1+1/\gamma^3}$).

B.2. Proof for Corollary 5.2

Corollary B.5 (Almost Bayes optimality of ENP)**.** *As non-spurious feature predictor becomes easier to learn* (η ! 0)*, the ratio of accuracies for ENP solution and Bayes optimal predictor approaches* 1 *on* c_1 *and* erfc($\rho_1/\sqrt{1+1/\gamma^3}$)/erfc(ρ_1) *on* c_2.

Proof. The proof of this corollary directly uses the results regarding the asymptotic performance of ENP from Theorem B.4. Since $\lim_{\eta\,!\,0}$ erfc$\left(\frac{k\mu k_2}{\sqrt{2}\eta}\right) = 2$, the performance of ENP on c_1 approaches 0.5 erfc($\rho_1\sqrt{1+1/\gamma}$) which is Bayes optimal on c_1. Similarly, on c_2 it approaches 0.5 erfc($\rho_1/\sqrt{1+1/\gamma^3}$). From this we get the performance ratios stated in Corollary B.5. □

B.3. Discussion on generalization error for ENP vs. ICC.

In the previous sections, for the class of linear predictors, we say that the asymptotic error for ENP is lower than IRM and conDRO on context c_1 and lower than ERM on c_2, under some conditions on problem parameters γ, p_c. Here, we will discuss why ENP performs better than ICC given only finite samples from the distribution. The main intuition behind this is that ICC learns a separate predictor for each context and consequently fails to learn the shared feature x_1 jointly using samples from both. Thus, for the minority context the learned predictor would generalize poorly. On the other hand, ENP learns a single predictor for both contexts and instead uses different augmentations for samples from each context. This allows ENP to use samples from the majority context to learn the shared feature x_1 that works well on the minority context as well.

We will now formalize this argument by relying upon existing generalization bounds in prior works for l_2 norm bounded linear predictors. Specifically, we reuse the following generalization bound that is derived using a union bound argument, and thus is applicable to any linear predictor in W_1 (including ERM estimate).

Lemma B.6 (Corollary 4 from Kakade et al. (2008))**.** *Let* ℓ *be a L-Lipschitz loss function,* S *a closed convex set and* $1/p + 1/q = 1$*. Suppose that* $X = \{x : \|x\|_p \quad X\}$ *and* $W = \{w \in S : \|w\|_q \quad W\}$*. Then we have for any* $\delta > 0$*, the generalization error of any* $w \in W$ *is bounded with probability* 1 δ*.*

$$\ell(\langle w, x\rangle, y) \quad \frac{1}{n}\sum_{i=1}^{n}\ell(\langle w, x^{(i)}\rangle, y^{(i)}) \quad LXW\sqrt{\frac{p-1}{n}} + LXW\sqrt{\frac{\log(1/\delta)}{2n}} \tag{12}$$

In particular, when we consider $p = q = 2$ and our bounded set of predictors W_1, we recover the bound:

$$\ell(\langle w, x\rangle, y) \quad \frac{1}{n}\sum_{i=1}^{n}\ell(\langle w, x^{(i)}\rangle, y^{(i)}) \quad LX\sqrt{\frac{1}{n}} + LX\sqrt{\frac{\log(1/\delta)}{2n}} \tag{13}$$

In order to use the above result, we need a high probability bound over the l_2 norm of the covariates: $\|x\|_2$ (denoted in the lemma as X), which we look into next.

Proposition B.7 (high probability bound over $\|x\|_2$)**.** *With probability* 1 $\frac{\delta}{2}$*, we can bound* $\|x\|_2$ *using Lemma B.8,*

$$\|x\|_2 < \max\{\sigma/\sqrt{\gamma}, \eta\}\left(\sqrt{2(\log 2/\delta)} + \sqrt{3d}\right) + \sqrt{3k\mu k_2^2}$$

Proof. Recall that conditioned on the label and context x follows a multivariate Gaussian distribution, as specified by (5). Now, for a multivariate Gaussian distribution centered at v 2 R^{3d} and with covariance Σ 2 $\mathsf{R}^{3d \quad 3d}$, we can use triangle inequality to conclude that $kxk_2 \quad kvk_2 + k\Sigma^{1/2}zk_2$. This is because we can write $x = v + \Sigma^{1/2}z$ where $z \quad N(\mathbf{0}_{3d}, \mathbf{I}_{3d})$

Hence, all we need to do is get a high probability bound over $k\Sigma^{1/2}zk_2$ which is a function of $3d$ independent Gaussian variables. Thus, we can apply the concentration bound in Lemma B.8. But before that, we need to compute the Lipschitz constant for the the function z 7! $k\Sigma^{1/2}zk_2$ in the euclidean norm.

$$jk\Sigma^{1/2}z_1k_2 \quad k\Sigma^{1/2}z_2k_2j \qquad k\Sigma^{1/2}(z_1 \quad z_2)k_2 \qquad \sqrt{k\Sigma k_{\mathsf{op}}} \quad kz_1 \quad z_2k_2 \tag{14}$$

Next, with the Lipschitz constant as $\sqrt{k\Sigma k_{\mathsf{op}}}$ we use Lemma B.8, to arrive at the following inequality which holds with probability at least $1 \quad \frac{\delta}{2}$.

$$kxk_2 \quad \sqrt{2k\Sigma k_{\mathsf{op}} \quad \log 2/\delta} + \mathsf{E}[k\Sigma^{1/2}zk_2] + kvk_2 \tag{15}$$

Finally, we can use Jensen to bound $\mathsf{E}[k\Sigma^{1/2}zk_2]$, *i.e.,*

$$\mathsf{E}\left[\sqrt{k\Sigma^{1/2}zk_2^2}\right] \quad \sqrt{\mathsf{E}\left[k\Sigma^{1/2}zk_2^2\right]} = \sqrt{\mathrm{tr}\left(\Sigma\right)}. \tag{16}$$

Here, we simplified $\mathsf{E}\left[k\Sigma^{1/2}zk_2^2\right]$ in the following way:

$$\mathsf{E}\left[k\Sigma^{1/2}zk_2^2\right] = \mathsf{E}\left[\mathrm{tr}\left(z^\top \Sigma z\right)\right] = \mathrm{tr}\left(\Sigma \quad \mathsf{E}\left[zz^\top\right]\right) = \mathrm{tr}\left(\Sigma\right)$$

Since the upper bound worsens with $k\Sigma k_{\mathsf{op}}$ and $\mathrm{tr}(\Sigma)$, we consider the covariance matrix of the Gaussian with the worst $k\Sigma k_{\mathsf{op}}$ and $\mathrm{tr}(\Sigma)$ over the choice of context and label. Recall that $\gamma \qquad 1$. Thus, we take Σ as determined by context c_2, *i.e.,* it is given by the following diagonal matrix: $\Sigma = \mathrm{diag}(\sigma^2, \sigma^2 \ldots, \sigma^2, \sigma^2/\gamma, \sigma^2/\gamma, \ldots, \sigma^2/\gamma, \eta^2, \eta^2, \ldots, \eta^2)$. Plugging in $k(k_{\mathsf{op}}\Sigma) \qquad \max f\eta, \sigma/^p\bar{\gamma}g$ and $\mathrm{tr}(\Sigma) \qquad 3dk(k_{\mathsf{op}}\Sigma)$, and $kvk_2 = \sqrt{3k\mu k_2^2}$ into the equation: kxk_2 $\sqrt{2k\Sigma k_{\mathsf{op}} \quad \log 2/\delta} + \sqrt{\mathrm{tr}(\Sigma)} + kvk_2$, we get the result in the statement of Proposition B.7, *i.e.,*, with probability $1 \quad \delta/2$,

$$kxk_2 \quad \sqrt{2\max f(\sigma^2/\gamma), \eta^2 g \quad \log 2/\delta} + \sqrt{3d\max f(\sigma^2/\gamma), \eta^2 g} + \sqrt{3k\mu k_2^2}.$$

□

Lemma B.8 (Lipschitz functions of Gaussians from Wainwright (2019))**.** *Let $X_1, \ldots, X_n$ be a vector of i.i.d. Gaussian variables and $f : \mathsf{R}^n$ 7! R be L-Lipschitz with respect to the Euclidean norm. Then the random variable $f(X) \quad \mathsf{E}[f(X)]$ is sub-Gaussian with parameter at most L, thus:*

$$\mathsf{P}[jf(X) \quad \mathsf{E}[f(X)]j \quad t] \quad 2 \quad \exp\left(\quad \frac{t^2}{2L^2}\right), \quad 8t \quad 0.$$

We can now use the high probability bound on kxk_2 from Proposition B.7 in the result in Lemma B.6. We will use L to denote the Lipschitz constant of the exponential loss. Note that L is finite since we know kxk_2 is bounded. We also use n_0 to denote the number of samples from minority context. Finally we apply union bound over the result in Proposition B.7 and Lemma B.6 to get the following result that bounds the generalization error on the minority context.

With high probability $1 \quad \delta, 8w$ 2 W_1 we have:

$$\ell(hw, xi, \mathrm{y}) \quad \frac{1}{n}\sum_{i=1}^{n} \ell(hw, x^{(i)}i, \mathrm{y}^{(i)}) \quad L\left(\max f\frac{\sigma}{p\overline{\gamma}}, \eta g\left(\sqrt{2\left(\log\frac{2}{\delta}\right)} + {}^p\overline{3d}\right) + {}^p\overline{3}k\mu k_2\right)\left(p\frac{1}{\overline{n_0}} + \sqrt{\frac{\log(2/\delta)}{2n_0}}\right)$$

Given this generalization bound we now analyze the generalization gaps for ICC and ENP predictors on the minority context. We will use c_0 to denote the constant $p\overline{3}k\mu k_2 + ({}^p\overline{3d} + \sqrt{\log(2/\delta)})\max(\sigma/p\overline{\gamma}, \eta)$.

Recall that ICC simply runs ERM on points coming from each context individually. As a result, we can directly use the above generalization result to bound the generalization gap of ICC on the minority context which has n_0 labeled points. In our setting, the context assignment is modeled as a biased coin flip with probability p_c for context c_1. Thus denoting the number of points in the minority context as n_0, we have that n_0 binom$(n, 1 \quad p_c)$, were n is the total dataset size. We have that $\mathsf{E}[n_0] = n(1 \quad p_c)$ and $|n_0 \quad n(1 \quad p_c)| = O_p(1/\sqrt{n})$ by the Central Limit Theorem. This yields that the generalization bound is $O_p(Lc_0(1/\sqrt{n(1 \quad p_c)} + \sqrt{\log(2/\delta)/n(1 \quad p_c)}))$, where c_0 is as defined above.

In order to analyze ENP, we assume that (a) we have access to the ground-truth feature annotations, and (b) that we observe the samples after spurious features have been masked. Effectively, we consider that we are learning a linear predictor over the input space $\mathrm{C}_1 \quad \mathrm{x}$ when sample is from context c_1 and over input space $\mathrm{C}_2 \quad \mathrm{x}$ when sample is from c_2. Now, the bound over the constant X is given by a high probability bound over $\mathrm{C}_1 \quad \mathrm{x}$ and $\mathrm{C}_2 \quad \mathrm{x}$. Trivially, both of these are upper bounded by kxk_2. While this constant remains the same as in ICC, the key difference is that for ICC the bound is realized with only n_0 minority samples, but since ENP trains jointly on samples from both datasets the generalization bound is realized by all n samples. Consequently, given a dataset of n points, we have a generalization bound that is $O(Lc_0(1/\sqrt{n} + \sqrt{\log(1/\delta)/n}))$ where c_0 is the constant defined above.

We can summarize the above comparison between ICC and ENP on the minority context in terms of the following result on the estimation error of the two estimators.

Theorem B.9 (estimation error)**.** *When the exponential loss ℓ is optimized over W_1 using finite samples in $\hat{\mathsf{P}}_n$, then with probability $1 \quad \delta$ the generalization error on the minority context c_2 is $O_p(Lc_0(1/\sqrt{n(1 \quad p_c)} + \sqrt{\log(2/\delta)/n(1 \quad p_c)}))$ for the solution found by ICC (9), and $O(Lc_0(1/\sqrt{n} + \sqrt{\log(2/\delta)/n}))$ for the solution found by ENP. Here, $c_0 = \sqrt{3}k\mu k_2 + (\sqrt{3d} + \sqrt{\log(2/\delta)}) \max(\sigma/\sqrt{\gamma}, \eta)$.*

C. Semi-Synthetic Experimental Details

C.1. Corrupted Waterbirds

Dataset and Architecture As in the standard Waterbirds construction, we generated images using the CUB 2011 dataset and a subset of the Places365 dataset. However, 5% of the CUB images were corrupted by a random crop corresponding to 30% of the image, as well as a Gaussian Blur of radius 20. Like in the standard Waterbirds construction in Sagawa et al. (2019a), both the test and validation datasets were generated such that the background and foreground were uncorrelated.

In all experiments, we conducted training by fine-tuning an Imagenet-pretrained ResNet50 model (as done by Sagawa et al. (2019a)). All model weights were available to be updated during model training and a linear classification layer was appended to the model to generate the final classifications.

Baseline Model Training Details For group DRO and conDRO experiments, we performed hyperparameter tuning in the intervals around the hyperparameter values used by Sagawa et al. (2019a) in their Waterbirds experiments. For the ERM and GT-Aug experiments, we used the standard weight decay parameter of 1e-4 and tuned the best epoch using the validation dataset.

ENP: Feature Predictor Model In the Corrupted Waterbirds setting, we trained a feature predictor model to identify whether the foreground was corrupted or not and then used access to ground-truth segmentation masks to generate pixel-level feature annotations. In order to train the foreground corruption detector, we used the same architecture and hyperparameters as the standard Waterbird task (Resnet50).

ENP: Target Model Invariance Given the pixel-level spuriousness labels obtained from our feature predictor model, we generated enforced invariance to the spurious-labelled pixels by generating augmentations that added Gaussian noise to them but had the same label as the original sample. At test-time, we further enforced invariance by generating predicted pixel-level spuriousness labels, generating a fixed number of augmentations per sample, and using the averaged logits to compute the final classification.

Test Set Construction and Metric We used the standard training/test/validation designations from the WOMD. In addition, we assume ground-truth segmentations of foreground and background on test and validation datasets in order to generate augmentations (for both GT-Augs. and ENP). We report the worst-context-group accuracy on the test set (using

ground-truth contexts) except we exclude the two groups in which the spurious correlation breaks and the foreground is corrupted (since under this setting, it is impossible to identify the correct label and these groups have very low accuracy. Thus, our metric is the empirical counterpart of :

$$\min_{c2C^0k2K} \mathsf{E}_{P_{c,k}} \mathbb{1}fw(x) = yg \tag{17}$$

where C^0 denotes all context-groups except the corrupted-correlation breaking ones.

C.2. Noisy MountainCar

Environment, Data, and Architecture We used an expert MountainCar policy in order to generate a demonstration dataset consisting of 100 demonstrations. During post-processing, we applied heavy Gaussian noise (stdev = 0.07) to the velocity component of the state and clipped the resulting values within the permissible range for the feature value. We used a three-layer policy network with a hidden layer of size 50 (as implemented by De Haan et al. (2019)). For training the causal-graph parameterized policy we used a larger 4-layer network - with the same hidden layer size of 50 neurons. Our implementation of this environment followed the open-sourced code released by (De Haan et al., 2019) found at https://github.com/pimdh/causal-confusion/.

Baseline details We trained all models for 80 epochs and performed model selection by performing online policy evaluation. For our standard imitation learning baselines (With Prev. Action) and (Without Prev. Action), as well as the Policy Exec. Intervention, we tuned hyperparameters on an interval around the final values used by De Haan et al. (2019). We implemented the targeted exploration (Lyle et al., 2021) baseline by training an ensemble of imitation learning policies on the imitation learning dataset and then training an exploration policy using proximal policy optimization (PPO) ensemble uncertainty as the reward function. We ran this policy online, collected states visited, and added them (with their corresponding expert action into the imitation learning dataset. Finally, for our conDRO method, we devised groups according to the current action (core feature), previous action (spurious feature), and group. As a result, we had a total of $3 \quad 3 \quad 2 = 18$ groups and we tuned both weight decay and learning rate in the range $f1e \quad 5, 1e \quad 4, 1e \quad 3, 1e \quad 2, 1e \quad 1g$. For all baselines except policy execution interventions, we used the same model architecture as standard imitation learning

Evaluation and Metric All imitation learning policies were evaluated with online execution in the modified MountainCar environment (with states noised on the subset of the state space) and with access to the previous action feature. We reported the average reward attained by the imitation learning agent over 10 independent runs (i.e. independent imitation learning datasets and trained models). The reward function for MountainCar is sparse (as reward is only attained once the goal is reached and negative reward until that time) and the minimum value of 200 is attained when the goal is not reached.

ENP: Training a feature predictor We used the same architecture as the imitation learning model and trained on the 3-target classification problem of predicting the subset of reliable features (since our augmented state vector contained 3 features. We trained this model with feature annotations on 10% of our training data and found this was sufficient for 100% validation accuracy.

ENP: Training the target model We trained our target model using the standard imitation learning loss with data augmentations to enforce invariance to the spurious features. Since the only potentially spurious feature was the previous action, we generated augmentations which (when the feature was labelled as spurious) randomly perturbed the previous action by selecting uniformly from all actions. We generated these augmentations at training and test time (using the predicted feature annotations from our model).

D. Extended Discussion and Implementation of WOMD

D.1. Dataset and Architecture Details

Dataset The Waymo Open Motion Dataset (WOMD) consists of vehicle trajectory data collected on real roads as an autonomous vehicle navigates diverse traffic scenarios (intersections, traffic lights, etc.) alongside a variety of other road users (i.e., other cars, pedestrians, and cyclists). In this setting, the number of contexts is unclear and each input has a varying number of spurious/non-spurious features. As noted by (Ettinger et al., 2021), 46% of driving scenes in this dataset have over 32 nearby agents, 57% of the scenes have a pedestrian (with 20% having more than 4), and 16% of all scenes

Table 4. **MultiPath++ Training Hyperparameters** We show the set of hyperparameters used in training all MultiPath++ models in our WOMD experiments.

PARAMETER	VALUE
BATCH SIZE	42
LEARNING RATE	1E-4
GRADIENT NORM CLIPPING	0.4
MASK HISTORY PERCENTAGE	0.15
TOTAL TRAINING EPOCHS	120
LEARNING RATE SCHEDULER-TYPE	REDUCE ON PLATEAU
LEARNING RATE SCHEDULER-FACTOR	0.5
LEARNING RATE SCHEDULER-PATIENCE	20

have at least 1 cyclist. As such, we believe that WOMD is representative of the real-world autonomous driving settings where there could be a diverse range of interactions between multiple road agents. The task in this dataset is to predict the autonomous vehicle (AV) trajectory given the historical trajectory of both the autonomous vehicle and other agents.

Base Target Model Our experiments are conducted on the MultiPath++ (Varadarajan et al., 2021) trajectory prediction model (using the implementation at https://github.com/stepankonev/waymo-motion-prediction-challenge-2022-multipath-plus-plus) which is currently a state-of-the-art model for vehicle motion prediction tasks. The MultiPath++ model consists of LSTM trajectory encoders and fully connected road-graph polyline encoders followed by multi-context gating layers to model interactions between agents and fuse the road and agent information. Finally, a multi-context gating-based decoding layer generates a set of candidate predicted trajectories (see [3] for more information). In total, this model consists of 21 million parameters. Recently, (Roelofs et al., 2022) released a subset of WOMD with labels for whether nearby agents presented spurious or robust information with respect to the prediction of the AV trajectory. These labels were collected through a large-scale human annotation process where annotators were shown driving scenes from the perspective of the autonomous vehicle and were asked to select non-spurious agents through a web-based interface (Roelofs et al., 2022).

D.2. Training Details

Data Preprocessing We pre-processed data according to the reference implementation of MultiPath++ with some minor modifications. Due to computational constraints, we selected a random sample of the full WOMD dataset by downloading 100 shards from the Google Cloud Store. As the human-labelers for agent spuriousness were presented with the autonomous vehicle's (AV) point of view when labeling, we only trained our model to predict the trajectory of the AV. During data preprocessing, all agent trajectory data (positions, orientations, and velocities) was transformed into the autonomous vehicle's reference frame before being fed into the MultiPath++ model. In many driving scenarios, there were agents labeled as invalid, for example, due to not being in the autonomous vehicle's field of view. In these cases, we zeroed out all agent data corresponding, as well as setting the *valid* feature (part of the canonical feature representation to 0).

Standard Training Details We used all standard hyperparameters released in the reference WOMD implementation (found in the file final_RoP_Cov_Single.yaml and shown in the Table 4. We also tested larger learning rate parameters in the set $f0.01, 0.001, 0.00001g$ and did not find improvements with these parameters. We leave more intensive hyparameter tuning experiments for future work.

Test Set Construction and Metric We sourced our test set as a subset of the annotated driving scenarios contained in WOMD. As specified in (Roelofs et al., 2022), we used the spuriousness labels in order to delete all spurious labels from test set (by setting the valid feature of these agents to 0) and zeroing out the associated data. Importantly, we note that our test set was a *subset* of the annotated data: we reserved 20% of this data to use during training models that used agent spuriousness annotations. We computed the minimum average displacement error (minADE) as our final metric as shown in Equation 18:

$$\min_{i2[1,6]} \frac{1}{T} \sum_{j=1}^{T} jjt_j^{\text{gt}} \quad t_j^{\text{pred},i} jj \tag{18}$$

Table 5. **Corrupted Waterbirds Ablation on Annotated Samples.** We show the effect of different training set sizes on the accuracy of the feature predictor on Corrupted Waterbirds.

% ANNOTATED TRAINING	0.5	1	2	5	10
FEATURE PREDICTOR ACC.	90%	95.3%	97.5%	99.7%	99.9%

Table 6. **WOMD Ablation on Annotated Samples.** We show the effect of different training set sizes on the accuracy of the feature predictor on the WOMD dataset.

% ANNOTATED TRAINING	0.1	1	5	20	50
FEATURE PREDICTOR ACC.	61%	77%	83%	84.9%	85%

Data Augmentation Details We adapt our data-augmentations strategy from (Roelofs et al., 2022). As introduced by that work, driving scenarios with associated spuriousness annotations were generated by randomly deleting (i.e. setting the valid feature to 0) all spurious-labelled agents with 10% probability. In our implementation of Annotation Augmentations, we followed this procedure exactly: 20% of the annotated data was added to the WOMD training dataset and all these added points were augmented according to the spuriousness labels. In our annotation-free baseline, Random Augmentations, we simply performed random deletion across *all agents*. In section D.3, we describe how augmentations were performed in ENP.

D.3. ENP Details

Feature Annotation Model We designed a lightweight feature annotation model based off of the Multipath++ architecture. Due to the variable number of agents in the scene, we opted to train an agent-conditioned model which took the road graph and other global information as input, as well as the trajectory for a given agent (in autonomous vehicle coordinates) and predicted spuriousness of the provided information. Therefore, we included all road graph embedding modules from the Multipath++ model and a single LSTM encoder for accepting the autonomous vehicle trajectory. All representations from these modules were concatenated and fed through a fully connected network in order to output the predicted spuriousness attribute. During preprocessing for our feature-prediction training set, we subsampled the number of invalid labeled agents in order to ensure dataset balance.

Target Model Training With our feature predictor, we went through all trajectories in our training set and labeled each agent as spurious or non-spurious using our model. During MultiPath++ training, we adopted an identical approach to the Annotated Augmentations except now all trajectories were augmented in accordance with the feature predictor's labels. Although the ENP framework also involves test-time augmentations, these were not applicable in the WOMD setting because all spurious agents were already removed from the dataset (also identical to the implementation of Annotated Augmentations).

E. Ablations

In this section, we conduct ablations on the number of explicit non-spurious feature annotated samples. In Table 5, we show the feature predictor accuracy given different percentages of feature annotations on Waterbirds and find that it is very high even with a very small percentage of annotated samples. We see a similar effect with the WOMD accuracy though the accuracy begins to decay quickly on smaller subsets of annotated samples (Table 6).