# NUISANCES VIA NEGATIVA: ADJUSTING FOR SPURIOUS CORRELATIONS VIA DATA AUGMENTATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

There exist features that are related to the label in the same way across different settings for that task; these are semantic features or *semantics*. Features with varying relationships to the label are *nuisances*. For example, in detecting cows from natural images, the shape of the head is a semantic and because images of cows often have grass backgrounds but not always, the background is a nuisance. Relationships between a nuisance and the label are unstable across settings and, consequently, models that exploit nuisance-label relationships face performance degradation when these relationships change. Direct knowledge of a nuisance helps build models that are robust to such changes, but knowledge of a nuisance requires extra annotations beyond the label and the covariates. In this paper, we develop an alternative way to produces robust models by data augmentation. These data augmentations corrupt semantic information to produce models that identify and adjust for where nuisances drive predictions. We study semantic corruptions in powering different robust-modeling methods for multiple out-of distribution (OOD) tasks like classifying waterbirds, natural language inference, and detecting Cardiomegaly in chest X-rays.

## 1 INTRODUCTION

Relationships between the label and the covariates can change across data collected at different places and times. For example, in classifying animals, data collected in natural habitats have cows appear on grasslands, while penguins appear on backgrounds of snow; these animal-background relationships do not hold outside natural habitats (Beery et al., 2018; Arjovsky et al., 2019). Some features, like an animal's shape, are predictive of the label across all settings for a task; these are semantic features, or semantics in short. Other features with varying relationships with the label, like the background, are nuisances. Even with semantics present, models trained via empirical risk minimization (ERM) can predict using nuisances and thus fail to generalize (Geirhos et al., 2020).

Models that rely only on the semantic features perform well even when the nuisance-label relationship changes, unlike models that rely on nuisances. Many methods exist to build models robust to changing nuisance-label relationships (Mahabadi et al., 2019; Makar et al., 2022; Liu et al., 2021; Puli et al., 2022; He et al., 2019); we call these nuisance-avoiding methods. These methods broadly fall into two classes: 1) methods that assume access to nuisances, like Nuisance-Randomized Distillation (NURD) (Puli et al., 2022), debiased focus loss (DFL), product of experts (POE) (Mahabadi et al., 2019), and 2) methods that rely on assumptions about ERM-trained models rely on nuisances like Just Train Twice (JTT) (Liu et al., 2021). The commonality between two classes of methods lies in building a model that predicts the label from the nuisance called a *biased model*. At a high level, biased models play a role in building robust predictive models by providing a way to measure when the nuisance can influence predictions.

To build the biased model, methods require either extra annotations in the form nuisances being known in the training data or assumptions like ERM-trained models rely on nuisances. In this work, we build robust models from a different and complementary source of assumptions: knowledge about semantics. Imagine using data augmentation to corrupt semantics in the covariates — if the resulting semantic-corrupted input can still predict the label, the prediction must rely on nuisances, thereby providing a window into nuisances that can be used to build a biased model.

Designing a data augmentation that corrupts semantics is easy. For example, replacing the covariates with random noise would fully corrupt the semantics. However, after such a corruption there is nothing that predicts the label meaning no nuisance information would be identified. A better

semantic corruption would corrupt the semantics, while preserving some nuisances. This preservation is possible when semantics and nuisances appear differently in the covariates; we call such a difference a separation. We identify two separations to develop semantic corruptions for object recognition and natural language inference (NLI).

The first separation is when semantics are global and nuisances are local. Formally, global semantics are position-dependent functions of the subsets of the covariates (patches in images, or words in sentences), while local nuisances are position-independent functions. For example, in recognizing cows, the shape of the animal structures the distant patches where the cow's eyes, ears, tail, hooves appear; nuisances like grass can appear anywhere without structure. In NLI, the meaning of the sentence dictates its structure via the order of words; nuisances like shared words between the premise and the hypothesis (McCoy et al., 2019) appear without order. Due to positional-dependence, randomizing positions of subsets of covariates corrupts global semantics; however, local nuisances are position-independent and are retained. Following this insight, we corrupt the global semantic shape in object recognition by permuting different patches, i.e. patch randomization (PATCH-RAND), and corrupt the global semantic word order in NLI via n-gram randomization (NGRAM-RAND).

The second separation is when certain parts of the input are required for semantics. For example in chest X-rays, lungs appear in the center, while nuisances like the scanner affect the border. In NLI, the premise sets up the context for detecting entailment. Without the premise, entailment cannot be determined by semantics, but the hypothesis retains some nuisances. For this separation, masking parts of the covariates corrupts semantics for object recognition via region-of-interest masking (ROI-MASK) and for the semantic context in NLI via premise masking (PREM-MASK).

We demonstrate the value of semantic corruption by using it to power a variety of methods, NURD (Puli et al., 2022), DFL, POE (Mahabadi et al., 2019), and JTT (Liu et al., 2021). We run these methods by building biased models using nuisance produced by semantic corruption. These methods with semantic corruptions outperform ERM on out-of distribution (OOD) generalization tasks like waterbirds (Sagawa et al., 2019), cardiomegaly detection from chest X-rays, and NLI. The performance of NURD, DFL, POE run with semantic corruption is similar to what the methods achieve with extra observed nuisance variables. Finally, JTT with semantic corruptions outperforms vanilla JTT.

## 2 What do methods need to reduce spurious correlations?

A spurious correlation is a relationship between the covariates and the label that changes across settings like time and location (Geirhos et al., 2020). Models that exploit a spurious correlation can perform poorly outside the training distribution. We focus on the class of methods that correct models using knowledge of nuisances or where they might appear (Mahabadi et al., 2019; Liu et al., 2021; Puli et al., 2022); we call these nuisance-avoiding methods. With label $\mathbf{y}$, a vector of nuisances $\mathbf{z}$, and covariates $\mathbf{x}$, the goal is to predict well on data regardless of the nuisance-label relationship. Next, we establish that the central part of several nuisance-avoiding methods is a model that predicts the label using nuisances, which we call the biased model.

**NURD.** Puli et al. (2022) identify a conditional that has performance guarantees on every $p_{te} \in \mathcal{F}$, and develop NURD to learn it using data from $p_{tr} \neq p_{te}$. NURD uses 1) the nuisance-randomized distribution, $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y}) p_{\perp\!\!\!\perp}(\mathbf{z}) p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$, where $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y}$, and 2) an uncorrelating representation $r(\mathbf{x})$ for which $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r(\mathbf{x})$. In $p_{\perp\!\!\!\perp}$, the nuisance alone cannot predict the label; this helps avoid features that depend only on the nuisance. Next, features that are mixed functions of the label and the nuisance (e.g. $\mathbf{x}_1 = \mathbf{y} + \mathbf{z}$) can also be spurious. Uncorrelating $r(\mathbf{x})$ avoid such features. With these insights, NURD builds models of the form $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ that are most informative of the label. We work with reweighting-NURD, which estimates $p_{\perp\!\!\!\perp}$ by weighting samples as $p(\mathbf{y}) / p_{tr}(\mathbf{y} \mid \mathbf{z}) p_{tr}(\mathbf{y}, \mathbf{z}, \mathbf{x})$. See appendix A for more detail.

**End-to-end bias mitigation.** Mahabadi et al. (2019) consider two methods to train a biased model and a base predictive model jointly to make the base model predict without relying on the biases. The methods use and fine-tune a BERT model (Devlin et al., 2019) and do not propagate the gradients of the biased model to update the common parameters (token embeddings in this case). They propose 1) POE, where the `log` of the product of the predicted probabilities of the two models is used to compute the classification loss and 2) DFL, where the biased model is used to weight the cross-entropy loss for the base model. For both methods, Mahabadi et al. (2019) build a biased model as

**Table 1:** Summary of NURD, JTT, POE, and DFL. Each method approximates what we call a biased model: $p_{tr}(\mathbf{y} \mid \mathbf{z})$. This table describes the different biased models, their names, how they are built. We build biased models as $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$ where $T(\mathbf{x})$ is a semantic corruption.

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

$p_{tr}(\mathbf{y} \mid \mathbf{z})$. The intuition is that the samples correctly classified by the biased model will have low loss and the base model focuses on classifying samples that the biased model misclassifies.

**Just Train Twice JTT.** Liu et al. (2021) aim to build models robust to group shift, where the relative mass of a fixed set of disjoint groups of the data changes between training and test times. The groups here are subsets of the data defined by a pair of values of discrete label and nuisance values. While they work without relying on training group annotations, i.e. without nuisances, they assume ERM builds models with high worst-group error. JTT first builds an "identification" model via ERM to isolate samples that are misclassified due to reliance on the nuisances. Then, JTT trains a model via ERM on data with the loss for the misclassified samples upweighted (by constant $\lambda$). The number of epochs to train the identification model and the upweighting constant are hyperparameters that require tuning using group annotations (Liu et al., 2021).

**The commonality of a biased model.** The central part between NURD, DFL, POE, and JTT is a model that predicts the label using nuisances, which we call the biased model as in He et al. (2019); Williams et al. (2018). While these methods reduce dependence on nuisances, they build biased models using additional annotations or require assumptions that ERM model predicts using only the nuisance. In the next section, we describe an alternative: corrupt semantic information with data augmentations to construct nuisances that can be used in a biased model.

## 3 Robustness via Semantic Corruptions

We define a data augmentation as a transformation of the covariates with random noise $\boldsymbol{\delta}$: $T(\mathbf{x}, \boldsymbol{\delta})$. Formally, an ideal semantic corruption is a data augmentation or transformation such that the label does not depend on the transformation given the nuisance $\mathbf{y} \perp\!\!\!\perp T(\mathbf{x}, \boldsymbol{\delta}) \mid \mathbf{z}$ and that the transformation is not independent of the nuisance $T(\mathbf{x}, \boldsymbol{\delta}) \not\perp\!\!\!\perp \mathbf{z}$. The first condition ensures that the biased model built from the semantic corruption only predicts the label because of the nuisance. The second condition ensures the semantic corruption depends on the nuisance. In designing a semantic corruption these two conditions are in tension. The former wants to destroy everything about the label unrelated to the nuisance, while the latter wants to retain everything about the nuisance, which may be hard to achieve without retaining something extra about the label. The design of a semantic corruption is made easier, when semantics and nuisances appear differently in the covariates; we call such a difference a separation. Focusing on two popular OOD tasks, object recognition and NLI, we identify separations and build semantic corruptions based on permutations and masking.

### 3.1 Semantic corruptions via permutations for a global/local separation

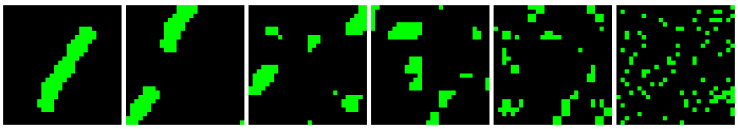
The first separation we consider is between semantics that appear as global structure and nuisances that appear as local structure. We give an intuitive example for such global semantics and local nuisances before formalizing them. As an illustrative example, consider the colored MNIST dataset (Arjovsky et al., 2019) where most images with the label 0 are red and most images with the label 1 are green. The semantic shape globally structures multiple pixels; e.g. the closed curve shape of a zero. However, any set of non-zero pixels determines color, wherever the pixels are. That is, semantics impose global structure while nuisances appear locally without structure. Formally, given subsets of the covariates $\mathbf{x}_1, \cdots \mathbf{x}_k$ extracted in a fixed order, semantics $s(\mathbf{x})$ are functions that depend on the order while nuisances $n(\mathbf{x})$ are permutation-invariant:

$$s(\mathbf{x}) = f(\mathbf{x}_1, \cdots \mathbf{x}_k), \qquad \forall \text{ permutations } \pi \quad n(\mathbf{x}) = g(\mathbf{x}_1, \cdots \mathbf{x}_k) = g(\mathbf{x}_{\pi(1)}, \cdots \mathbf{x}_{\pi(k)}) \tag{1}$$

We give a demonstrative example of a semantic corruption with global semantics and local nuisances. Consider a family of distributions $\mathcal{F} = \{p_D\}_{D \in R}$ with changing nuisance-label relationships. With $\mathcal{U}$ as the uniform distribution over $\{1, 2, 3\}$, and $\mathcal{N}$ as the normal distribution, samples from $p_D(\mathbf{y}, \mathbf{z}, \mathbf{x})$ come as discrete $\mathbf{y} \sim \mathcal{U}$, normal $\mathbf{z} \sim \mathcal{N}(D\mathbf{y}, 1)$, and a vector $\mathbf{x}$ generated as

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

**Figure 1:** PATCH-RAND on colored-MNIST. The original image is on the left followed by PATCH-RANDs of sizes $14, 7, 4, 2, 1$ respectively. While color is present at all sizes, shape is absent $< 7$.



In words, for any fixed $\mathbf{z}$, $\mathbf{y}$ determines the configuration of where the negation is. Determining $\mathbf{y}$ requires looking at two coordinates: $\mathbf{y} = 1$ if the second and third coordinates have the same sign, $\mathbf{y} = 2$ if the first and third do, and $\mathbf{y} = 3$ otherwise. In other words, the label is determined by a global relationship between fixed features. However, $\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$, meaning that $\mathbf{z}$ is determined regardless of where the negation is, i.e. it is local. Next, let $T(\mathbf{x}, \boldsymbol{\delta})$ be a random permutation of the three coordinates in $\mathbf{z}$, then $T(\mathbf{x}, \boldsymbol{\delta})$ is a semantic corruption. We omit the noise $\boldsymbol{\delta}$ as an argument to $T$ from here on for compactness. Since $T(\mathbf{x})$ randomly permutes the location of the negation, $T(\mathbf{x}) \mid \mathbf{y}, \mathbf{z}$ is distributed identically to $T(\mathbf{x}) \mid \mathbf{z}$. Thus, $T(\mathbf{x})$ is a semantic corruption because it also depends on $\mathbf{z}$. Next, we give semantic corruptions for tasks in vision and language with a global/local separation

**Patch randomization.** In image recognition, object recognition tasks where the object is a shape in the foreground and the color and texture in the background are nuisances, satisfy the global-semantics/local-nuisances separation. For illustration, consider the example of differentiating cows and penguins in natural images; here, shape is a global semantic feature, while grass in the background is local and can appear without structure. In another task, colored-MNIST, re-arranging patches in the image can corrupt the shape information leaving only color; see fig. 1. Similarly, consider the waterbirds dataset from (Sagawa et al., 2019) with waterbirds appearing predominantly on backgrounds with water and landbirds on backgrounds of land. Detecting the type of birds requires looking at semantics like wing shape, beak, and the presence of webbed feet. Such features can be corrupted by randomly permuting small patches. However, nuisances remain after such randomization because the background can be detected from small patches of the image due to colors and textures. See fig. 2 for an example. This is **patch randomization (PATCH-RAND)**.

**N-gram randomization.** Tasks like natural language inference (NLI) — where the goal is determining if a premise sentence entails a hypothesis — satisfy the global-semantics/local-nuisances separation. Consider this example: the sentence "Bob speaks but Jon does not" contradicts "Jon speaks but Bob does not" but entails "Bob speaks". The meaning imposes a global structure on the words and the order they appear in. Nuisances like the number of shared words between the hypothesis and the premise predict entailment, but do not impose order. Here, randomizing the order of the words corrupts the semantics. For example, one order randomization of the sentence "Jon speaks but Bob does not" is "Bob speaks but Jon does not"; the former entails "Jon speaks" but the latter contradicts it. Local nuisances, like the number of overlapping words or the presence of a certain negation word, are preserved after order randomization. We randomize the order by permuting different $n$-grams in each sentence; we call this **n-gram randomization (NGRAM-RAND)**.

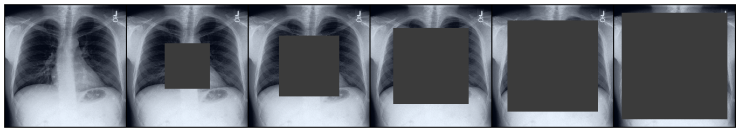
**Figure 2:** Patch randomization (PATCH-RAND) on waterbirds. The original is the left most, followed by PATCH-RANDs with sizes $112, 56, 28, 14, 7, 2$. At sizes less than 28, shape is hard to make out.

### 3.2 SEMANTIC CORRUPTIONS VIA MASKING FOR A LOCATION-BASED SEPARATION

The second separation we use to build semantic corruptions is based on when a certain subset of the covariates contain a necessary part of the semantic information, masking, by removing that subset or setting it to a constant, corrupts semantics. Such masking retains nuisances outside the subset. Formally, we corrupt the semantics by replacing subsets $\mathbf{x}_S$ with a value that is out of support: for example, in images where pixels lie in $(0, 1)$, we corrupt $\mathbf{x} = [\mathbf{x}_S, \mathbf{x}_{-S}]$ as $\mathbf{x}_{\text{corrupted}} = [0 * \mathbf{x}_S, \mathbf{x}_{-S}]$.

For example, consider the following family $\mathcal{F} = \{p_D\}_{D \in R}$ with varying nuisance-label relationships. With $\mathbf{a}, \mathbf{b}$ being random uniform binary random variables, $\mathbf{e}(\rho)$ as the exponential distribution

**Figure 3:** Masking to corrupt semantics in chest X-rays. The original is the left most, followed by ROI-MASK of size $84, 112, 140, 168, 196$. At sizes $140$ and more, the heart is blocked out.



with parameter $\rho$, and $s_+(u) = \log(1 + \exp(u))$ as soft-plus, let sampling from $p_D(\mathbf{y}, \mathbf{z}, \mathbf{x})$ be:

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

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

**Region-of-interest-masking for object recognition.** Semantics in images can often be localized to regions-of-interest. For example, in detecting cardiomegaly, the region-of-interest is the middle of the chest where the heart resides. Masking out the region of interest removes centrally located semantic information from the chest X-ray (fig. 3). However, nuisances like hospital-specific information (like alignment tokens (Zech et al., 2018)) are present in the border; see top right corner of the x-rays in fig. 3. We call this **ROI-MASK**. When the region-of-interest (ROI) is centrally located, the size of the mask can be varied and the resulting border-only images after masking are used as nuisances to build robust models.

**Premise-masking for NLI.** Semantic features in NLI rely on the meanings of the premise and the hypothesis sentences: for example, the premise states the occurrence of an event ("Alice sat while bob stood.") which can entail ("Alice sat.") or contradict ("Bob sat.") the hypothesis. The information about the setup in the premise is therefore crucial to differentiate between entailment and contradiction. If the context given by the premise is blocked out, the hypothesis sentence can predict the label only due to nuisances like the presence of negation words that correlate with contradictions (Gururangan et al., 2018). Thus, masking the premise is a semantic corruption for NLI that retains nuisance information present in the hypothesis; we call this **PREM-MASK**.

### 3.3 Using Semantic Corruptions in Practice

For each of the methods in table 1, we use a semantic corruption $T(\mathbf{x})$ in building a biased model $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$. For reweighting-NURD with semantic corruptions, we replace $p_{tr}(\mathbf{y} \mid \mathbf{z})$ with $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$ for a semantic corruption $T(\mathbf{x})$, for DFL and POE, we replace the model $p_{tr}(\mathbf{y} \mid \mathbf{z})$ with $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$, and for JTT, we use $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$ as the identification model.

**Choosing the "size" in PATCH-RAND, NGRAM-RAND, ROI-MASK.** For PATCH-RAND, NGRAM-RAND, and ROI-MASK, misspecifying the size parameter, such as setting it be too large or too small, runs the risk of either retaining semantics or corrupting nuisances. For example, with large patches, PATCH-RAND may not corrupt semantics; then the biased model may rely on semantics and in turn encourage the predictive model to not rely on them, thus resulting in poor performance. When a patch is a single pixel, PATCH-RAND can corrupt nuisances; for example, in fig. 2, PATCH-RAND with patch size 2 looks like noise. Then, the biased model may not rely on a nuisance which in turn leads the predictive models to rely on that nuisance.

One way to select the size parameter is using a small evaluation set from a distribution with a sufficiently different nuisance-label relationship from the training data; recent work (Mahabadi et al., 2019; Krueger et al., 2020) uses a small set from a distribution other than the training one. In our experiments, we report results for all considered size parameters. Then, we give results for selecting the size parameter alone using a small evaluation set from the test distribution.

## 4 Experiments

In this section, we study semantic corruptions in powering NURD (Puli et al., 2022), JTT (Liu et al., 2021), and POE and DFL (Mahabadi et al., 2019). To be faithful to the original evaluations of each method, we run them on tasks from their respective papers: NURD on waterbirds, JTT on waterbirds

**Table 2:** Mean test accuracy across 10 seeds of NURD with semantic corruptions on classifying waterbirds. *Known*-nuisance NURD uses a label for the type of background as the nuisance. Selecting sizes in PATCH-RAND (PR) and ROI-MASK (RM) based on a small evaluation set from the test distribution gives PR-size 14 and RM-size 196. With these sizes NURD with RM and PR close 99% and 93% of the gap between ERM and known-nuisance NURD respectively.

| *known* **z** | RM 196 | RM 168 | RM 140 | RM 112 | PR 7 | PR 14 | PR 28 | PR 56 | ERM |
|---|---|---|---|---|---|---|---|---|---|
| 87.2% | 87.0% | 87.6% | 85.1% | 85.5% | 82.6% | 86.0% | 83.9% | 80.4% | 69.2% |

and NLI where the nuisance is the presence of a negation word, and POE and DFL on NLI evaluated on a challenging test dataset, HANS (McCoy et al., 2019). Further, we run NURD on chest X-rays but focus on the task of Cardiomegaly detection as it is an object recognition task, instead of the original Pneumonia detection (Puli et al., 2022) which is a texture recognition task. See appendix B for implementation details.

**Metrics and model selection.** We report the average test accuracy for every method. To be able to compare to what JTT is trained for in Liu et al. (2021), we also report worst-group accuracy for JTT. For each method, we compare the performance of the original method to that of the methods run with semantic corruption. For every method being run with semantic corruptions, group annotations and nuisances are unavailable in the training data. Known-nuisance versions of POE, DFL, and NURD use direct knowledge of one or more nuisances during training. Vanilla JTT and JTT with semantic corruption use validation group annotations to early stop and tune hyperparameters. For each method with semantic corruption, we report performance for all considered size parameters. To provide a fair comparison to vanilla JTT, we follow Liu et al. (2021) and select size using the highest worst-group accuracy on the validation data. For the other methods, we show that selecting the size parameter alone on a small evaluation set from the test distribution helps close the gap between using semantic corruption and using known nuisances; this follows the common practice of using limited queries on a small test dataset to tune hyperparameters (Gulrajani and Lopez-Paz, 2020).

### 4.1 Object recognition tasks

To be faithful to the original evaluations of each method, we evaluate JTT on waterbirds, and NURD on both waterbirds and detecting cardiomegaly; both tasks have images of size $224 \times 224 \times 3$. For both tasks, we use PATCH-RAND (of patch sizes $7, 14, 28, 56$) and ROI-MASK (of mask sizes $112, 140, 168, 196$) as semantic corruptions. Both Puli et al. (2022) and Liu et al. (2021) use the raw waterbirds data from Sagawa et al. (2019), where the task is detecting the type of bird (water or land) from images where the background is a nuisance. Unlike Liu et al. (2021), Puli et al. (2022) process the waterbirds to get a different setup from Sagawa et al. (2019). To stay true to the original evaluations of the methods, we recreate the setups as described in their respective papers.

**NURD on waterbirds.** For NURD, we recreate the waterbirds experiment from Puli et al. (2022) where the full waterbirds data from Sagawa et al. (2019) was subsampled into training, validation, and test datasets each with label balance. However, unlike Sagawa et al. (2019), the validation data comes from the same distribution as the training data. The training and validation datasets have 90% waterbirds on backgrounds with water and 90% landbirds on backgrounds with land. The test data has a flipped relationship. Known-nuisance NURD uses an additional label denoting the background-type as the nuisance.

Table 2 gives results. NURD with PATCH-RAND for any patch-size or ROI-MASK for any mask-size outperforms ERM (69.2%). Selecting the size hyperparameter based on the average accuracy over 10 seeds on an evaluation dataset (200 samples) gives PATCH-RAND size 14 and ROI-MASK size 196. With ROI-MASK of size 196, NURD achieves 87% test accuracy, which closes 99% of the gap in test accuracy between ERM and *known-nuisance* NURD (87.2%). NURD with PATCH-RAND of size 14 achieves a test accuracy of 86%, which closes 93% of the gap. To report the same metrics as Puli et al. (2022), we give the standard error over 10 seeds in table 7 in appendix B.

**JTT on waterbirds.** For JTT, we repeat the waterbirds experiment from Liu et al. (2021) which uses the original data from (Sagawa et al., 2019). The training data has 95% waterbirds on backgrounds with water and 95% landbirds on backgrounds with land. Both the validation and test datasets have bird label independent of the background. The groups here are subsets of the data that correspond to a pair of values of bird-type and background-type. Like vanilla JTT, we use group

**Table 3:** Test worst-group accuracies of JTT with semantic corruptions on waterbirds. Selecting sizes in PATCH-RAND (PR) and ROI-MASK (RM) based on validation worst-group accuracy gives PR-size 14 and RM-size 196. With these sizes, JTT with RM and PR outperform vanilla JTT.

| *Vanilla* JTT | RM 196 | RM 168 | RM 140 | RM 112 | PR 7 | PR 14 | PR 28 | PR 56 | ERM |
|---|---|---|---|---|---|---|---|---|---|
| 86.5% | 88.2% | 88.0% | 86.9% | 86.2% | 89.3% | 89.0% | 88.9% | 89.1% | 72% |

**Table 4:** Mean test accuracy across 10 seeds of NURD with semantic corruptions on detecting cardiomegaly from chest X-rays. *Known*-nuisance NURD uses the hospital as the nuisance. Selecting sizes in PATCH-RAND (PR) and ROI-MASK (RM) based on a small evaluation set from the test distribution gives PR-size 7 and RM-size 196. With these sizes, NURD with RM and PR close 77% and 79% of the gap between ERM and known-nuisance NURD respectively.

| *known* **z** | RM 196 | RM 168 | RM 140 | RM 112 | PR 7 | PR 14 | PR 28 | PR 56 | ERM |
|---|---|---|---|---|---|---|---|---|---|
| 81.7% | 77.2% | 75.0% | 74.4% | 68.9% | 77.5% | 75.2% | 72.2% | 69.2% | 62% |

annotations in the validation data to compute worst-group error and early stop training when using PATCH-RAND. The results for vanilla JTT are from our run using the reported optimal hyperparameters from (Liu et al., 2021).

Table 3 gives results. JTT with PATCH-RAND and ROI-MASK beats ERM at every patch/border-size. JTT with PATCH-RAND outperforms vanilla JTT (86.5%) at *every* patch size on test worst-group accuracy. Selecting the patch-size size hyperparameter on validation worst-group accuracy gives a size 14, which corresponds to a worst-group test accuracy of 89%. For JTT with ROI-MASK, selecting the size hyperparameter on validation worst-group accuracy gives a mask-size 196, which corresponds to a worst-group test accuracy of 88.2%, again outperforming vanilla JTT.

**NURD on detecting cardiomegaly** In chest X-ray classification, differences between hospitals, like the scanners used to produce the X-rays, are known to correlate thorasic conditions with non-physiological aspects in the image; for example, only some scanners render the air in the lungs in white (Zech et al., 2018). We consider the shape-based object recognition task of cardiomegaly (an irregularly sized heart) detection and, following Puli et al. (2022), construct a dataset by mixing two chest X-ray datasets: chexpert (Irvin et al., 2019) and MIMIC (Johnson et al., 2019). The training and validation datasets have 90% cardiomegaly images from MIMIC and 90% healthy images from chexpert. The test data has 90% of the cardiomegaly images from chexpert and 90% of the healthy images from MIMIC. Known-nuisance NURD uses a variable indicating the hospital identity as the nuisance.

See table 4 for results. NURD with PATCH-RAND and ROI-MASK outperform ERM at all sizes. Selecting the size hyperparameter based on the average accuracy over 10 seeds on an evaluation dataset (200 samples) gives PATCH-RAND size 7 and ROI-MASK size 196. With ROI-MASK of size 196, NURD achieves 77.2% test accuracy, which closes 77% of the gap in test accuracy between ERM and *known-nuisance* NURD (81.7%). NURD with PATCH-RAND of size 7 achieves a test accuracy of 77.5%, which closes 79% of the gap. To report the same metrics as Puli et al. (2022), we give the standard error over 10 seeds in table 8 in appendix B.

### 4.2 Natural Language Inference (NLI)

For methods POE, DFL, and JTT, we use the MNLI dataset (Williams et al., 2018) during training. The evaluations of these methods in their respective papers have different nuisances and, consequently, different test sets. In accordance, we describe the setup and the results separately. We use NGRAM-RAND (sizes

**Table 5:** Average accuracies (over 4 seeds) of POE and DFL with semantic corruptions on the HANS dataset. We report the performances of POE and DFL from Mahabadi et al. (2019) as *known*, where both methods use known nuisances. For both methods, selecting the size hyperparameter based on the average accuracy on a small evaluation dataset (1000 samples) from the test distribution gives $n = 3$. With this size, POE with NGRAM-RAND performs better than known-nuisance POE while DFL with NGRAM-RAND closes 84% of the gap between ERM and known-nuisance DFL.

| **z** | POE | DFL |
|---|---|---|
| *Known* | 66.3% | 69.3% |
| 1-gram | 65.7% | 66.5% |
| 2-gram | 66.0% | 68.5% |
| 3-gram | 66.7% | 68.4% |
| 4-gram | 66.2% | 65.0% |
| ERM | – | 63.6%. |

$1, 2, 3, 4$) to produce nuisances for both setups. We do not run PREM-MASK for evaluating POE/DFL because PREM-MASK corrupts nuisances shared across the sentences that HANS tests.

**POE and DFL** For POE and DFL, we report test accuracies on the HANS dataset McCoy et al. (2019) as in Mahabadi et al. (2019). HANS was created to test the reliance of models on three known nuisances: 1) lexical overlap, 2) subsequence match, and 3) constituent matching subtrees in the parse trees. Known-nuisance POE and DFL use exact knowledge of these nuisances.

Table 5 gives the results. Both methods beat ERM for all $n$-gram sizes. For both DFL and POE, selecting the size hyperparameter based on the average accuracy on a small evaluation dataset (1000 samples) from the test distribution gives $n = 3$. POE with NGRAM-RAND of size 3 achieves a test accuracy of $66.7\%$, improving over POE with known nuisances ($66.3\%$). DFL with NGRAM-RAND of size 3, achieves a test accuracy of $68.4\%$, closing $84\%$ of the gap between ERM and known-nuisance DFL ($69.3\%$). To report the same metrics as Mahabadi et al. (2019), we give standard deviations in table 9 in appendix B.

**Table 6:** Worst-group and average test accuracies of JTT with semantic corruptions on NLI. JTT with PREM-MASK and NGRAM-RAND of every size outperforms vanilla JTT. Selecting the size hyperparameter for NGRAM-RAND using validation worst-group accuracy, like Liu et al. (2021) do for vanilla JTT, gives $n = 1$. At this size, JTT with NGRAM-RAND outperforms vanilla JTT by 3 points of accuracy.

| | Worst-group | Average |
|---|---|---|
| *Vanilla* JTT | $71.3\%$ | $79.1\%$ |
| PREM-MASK | $72.1\%$ | $79.9\%$ |
| 1-gram | $74.3\%$ | $79.7\%$ |
| 2-gram | $71.9\%$ | $80.0\%$ |
| 3-gram | $72.0\%$ | $80.1\%$ |
| 4-gram | $73.4\%$ | $80.4\%$ |
| ERM | $67.9\%$ | – |

**JTT** For JTT, we repeat the NLI experiment from Liu et al. (2021), where the presence of a negation word in the hypothesis sentence is the nuisance. The groups here are subsets of the data that correspond to a value of the label and whether or not there is a negation word in the hypothesis. Vanilla JTT uses group annotations in the validation data to tune the hyperparameters and early stop training. For each $n$-gram size, we run JTT with NGRAM-RAND for two values of the number of epochs of training for the identification model: $2, 3$. Following the hyperparameter selection procedure from Liu et al. (2021), for each $n$-gram size, we give the results for the run with the higher validation worst-group accuracy. We run *vanilla* JTT using the reported optimization hyperparameters from (Liu et al., 2021).

Table 6 gives the results. JTT using NGRAM-RAND at *every* size or PREM-MASK perform better than both vanilla JTT ($71.3\%$) and ERM ($67.9\%$). Selecting the size hyperparameter for NGRAM-RAND using validation worst-group accuracy, like Liu et al. (2021) do for JTT, gives $n = 1$ with test worst-group accuracy of $74.3\%$, again better than vanilla JTT's $71.3\%$.

## 5 Related Work

Different nuisance-avoiding methods obtain nuisances using different assumptions. Many works (Veitch et al., 2021; Clark et al., 2019; Puli et al., 2022; He et al., 2019) obtain nuisances from the data using domain knowledge. Makar et al. (2022) assume the nuisance is available as an auxiliary label during training. Semantic corruptions offer a complementary approach to hand-crafting nuisances or obtaining auxiliary labels. Semantic corruption acts as a catchall by capturing every nuisance of one type (e.g. local nuisances in object recognition) by using the complementary knowledge about semantics.

Liu et al. (2021) assume that an ERM model built on the training data will exploit the nuisance-label relationship and propose to use such a model to identify samples that are misclassified by the nuisance. However, when the identification model relies on semantics, upweighting misclassified samples produces data with a different label-semantic relationship from the one in the training data. Thus, a model trained on the upweighted data has suboptimal performance on any test data that has the same semantic relationship as the training data. Semantic corruptions help reduce the identification model's reliance on the semantics.

The idea of splitting an image into patches and randomizing local patches has been used in self-supervised learning and generative modelling (Sinha et al., 2021), and domain generalization (Carlucci et al., 2019). Sinha et al. (2021) only use the patch-randomized images to define the support for the generated images and representations but do not guarantee models avoid nuisances. Carlucci et al. (2019) use patch-randomized images to encourage a model to recover semantic structure. In

contrast, we use PATCH-RAND to corrupt semantics and build biased models that rely on the nuisances, which help build predictive models that avoid reliance on nuisances. Work like (He et al., 2019; Puli et al., 2022) also use semantic corruptions without pointing out the reliance on knowledge about semantic features in producing nuisances. He et al. (2019) use the hypothesis as a nuisance to build a biased model for NLI; this is the masking based semantic corruption PREM-MASK. Unlike NGRAM-RAND, such corruptions will also corrupt nuisances like the number of common words which depend on the premise. Puli et al. (2022) focus on chest X-ray classification, and use the out-of-body border of the X-ray as a nuisance; this is corrupting semantics via ROI-MASK.

Bahng et al. (2020) use CNNs with small receptive fields (RFs), to help capture local nuisances. However, their method is typically limited to very small filters because at size 3x3, deep neural networks like VGG detect non-local semantics like shapes. In contrast, the size choice in PATCH-RAND has no bearing on the choice of the model; we used default vision models. Bras et al. (2020) automatically identify and remove examples with nuisances using adversarial filtering, but risk removing genuinely easy examples. Qin et al. (2021) work solely with vision transformers and consider why labels can be predicted from transformations akin to patch-randomized images. Concluding that this can only be due to nuisances, they propose to encourage the transformer to have predictions and representations of the original images be dissimilar from those of patch-randomized images. In contrast, our work applies to general flexible models and shows that semantic corruptions can be used to break the label's relationship with nuisances in the original images.

## 6 DISCUSSION

We study the use of semantic knowledge in building robust models. Given a procedure to corrupt semantics, anything that predicts the label in the corrupted input is a nuisance. Using semantic corruptions, practitioners can run different kinds of nuisance-avoiding methods (NURD, JTT, DFL, POE). With these semantic corruptions, methods like NURD and DFL perform close to how they would with known nuisances, and methods like JTT perform better than how they would when relying on ERM on the raw covariates to build a nuisance.

**Limitations.** The quality of any semantic corruption, and thus the quality of the results, depends on the extent to which semantics are destroyed and nuisances are retained. PATCH-RAND and NGRAM-RAND are built to corrupt global semantics and retain local nuisances, ROI-MASK to retain nuisances outside the ROI and PREM-MASK to retain nuisances in hypothesis. When applied to cases other than what they were built for, these methods may not destroy all the semantics or retain all the nuisances and thus yield models that generalize.

For example, when PATCH-RAND is used blindly on covariates with local semantics, the biased model may rely on said semantics; this in turn guides the predictive model to ignore these semantics and, thus, lose predictive performance. Alternatively, when nuisances are global, PATCH-RAND may corrupt them. For example in detecting cows and penguins, other nuisance animals (like dogs) may co-occur with cows more often; PATCH-RAND would corrupt this nuisance animal. Using PATCH-RAND in a nuisance-avoiding method for such tasks could lead to non-robust predictive models that rely on corrupted nuisances.

Our experiments show blind usage does not lead to poor performance despite violations of the separation that underlies the semantic corruption. In both classifying waterbirds and NLI, there exist local semantics, like small beaks and individual words, that are not corrupted by PATCH-RAND and NGRAM-RAND respectively. However, in our Waterbirds and NLI experiments, we show models built using semantic corruptions close more than $80\%$ of the gap in test performance between ERM and the methods that use known nuisances. Similarly, ROI-MASK corrupts nuisances in the ROI and retains semantics outside it. However, in both waterbirds and cardiomegaly detection, where nuisances like hospital-specific tokens and background features lie in the ROI, ROI-MASK still helps NURD close $> 75\%$ of the gap in test performance between ERM and NURD with a known nuisance.

**Summary.** Semantic corruptions power nuisance-avoiding methods to build models robust to spurious correlations without requiring extra annotations in the form of known nuisances during training or relying on hard to verify assumptions like ERM-trained models relying on nuisances. As discussed above, our experiments point out that using semantic corruptions leads to improved robustness even under violations of the separation assumptions they are built off of. These two properties indicate the value of semantic corruptions as a way to build robust models.

## REFERENCES


Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

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

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

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. *arXiv preprint arXiv:1909.06321*, 2019.

Maggie Makar, Ben Packer, Dan Moldovan, Davis Blalock, Yoni Halpern, and Alexander D'Amour. Causally-motivated shortcut removal using auxiliary labels. In *AISTATS*, 2022.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

Aahlad Manas Puli, Lily H Zhang, Eric Karl Oermann, and Rajesh Ranganath. Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=12RoR2o32T.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. *arXiv preprint arXiv:1908.10763*, 2019.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

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

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 590–597, 2019.

Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019.

Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. *arXiv preprint arXiv:2102.05113*, 2021.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR, 2020.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *ICML*, 2020.

Yao Qin, Chiyuan Zhang, Ting Chen, Balaji Lakshminarayanan, Alex Beutel, and Xuezhi Wang. Understanding and improving robustness of vision transformers through patch-based negative augmentation. *arXiv preprint arXiv:2110.07858*, 2021.

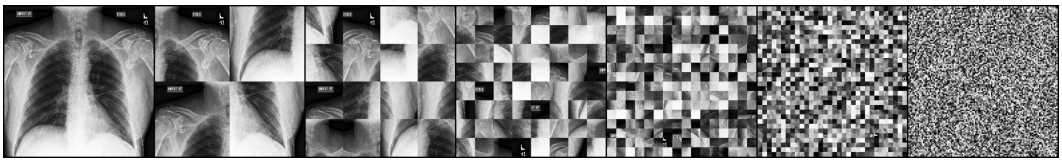


**Figure 4:** Example of PATCH-RAND of a chest X-ray image. The original image is the left most followed by PATCH-RANDs with sizes $112, 56, 28, 14, 7, 2$ respectively.

## A FURTHER DETAILS ABOUT NUISANCE-AVOIDING METHODS

**NURD.** Focusing on mitigating spurious correlations, Puli et al. (2022) identify a conditional that has performance guarantees on every test distribution within a family of distributions with varying nuisance-label relationships: $p_{te} \in \mathcal{F}$. They develop NURD to learn the conditional using data only from $p_{tr} \neq p_{te}$. NURD uses 1) the *nuisance-randomized distribution*, $p_{\perp\!\!\!\perp}(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p(\mathbf{y}) p_{\perp\!\!\!\perp}(\mathbf{z}) p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$, where $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y}$, and 2) an *uncorrelating representation* $r(\mathbf{x})$ for which $\mathbf{z} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{y} \mid r(\mathbf{x})$. NURD builds models of the form $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$ that are most informative of the label. To run reweighting-NURD with semantic corruptions, we replace $p_{tr}(\mathbf{y} \mid \mathbf{z})$ with $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$ for a semantic corruption $T(\mathbf{x})$. Our semantic corruptions are noisy functions of $\mathbf{x}$: with noise $\boldsymbol{\epsilon}$ such that $(\mathbf{y}, \mathbf{z}, \mathbf{x}) \perp\!\!\!\perp_{p_D} \boldsymbol{\epsilon}$, $T(\mathbf{x}) = U(\mathbf{x}, \boldsymbol{\epsilon})$. This implies

$$\mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \boldsymbol{\epsilon} \mid \mathbf{x} \implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} \mathbf{x}, \boldsymbol{\epsilon} \mid \mathbf{x} \implies \mathbf{y} \perp\!\!\!\perp_{p_{\perp\!\!\!\perp}} T(\mathbf{x}) \mid \mathbf{x}$$

Thus, $r(\mathbf{x}) = \mathbf{x}$ is uncorrelating and $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ achieves the optimality guarantees in Puli et al. (2022). These optimality guarantees imply that regardless of the test nuisance-label relationship, $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ will achieve optimal performance within the class of models like $p_{\perp\!\!\!\perp}(\mathbf{y} \mid r(\mathbf{x}))$. Thus, NURD with semantic corruption $T(\mathbf{x})$ learns $p_{\perp\!\!\!\perp}(\mathbf{y} \mid \mathbf{x})$ where $p_{\perp\!\!\!\perp}$ is the distribution $\frac{p(\mathbf{y})}{p_{tr}(\mathbf{y} \mid T(\mathbf{x}))} p_{tr}(\mathbf{y}, \mathbf{x})$.

**Just Train Twice (JTT).** JTT works in two stages: 1) build an "identification" model via ERM on the training data to isolate samples that are misclassified due to reliance on the nuisances and 2) train a model via ERM on data with the loss for the misclassified samples upweighted (by constant $\lambda$). In this work, we build the identification model on semantic corruptions i.e. we learn $p_{tr}(\mathbf{y} \mid T(\mathbf{x}))$. The training samples to be upweighted are the ones misclassified when predicting with the identification model on PATCH-RAND-versions of the sample. The second stage is run as in (Liu et al., 2021) with training data.

## B FURTHER EXPERIMENTAL DETAILS

See fig. 4 for an example of PATCH-RAND for chest X-rays.

### B.1 IMPLEMENTATION DETAILS

Each experiment in the paper was run on up to 2 RTX8000 GPUs. The hyperparameters for methods which use known nuisances in the training data like NURD, POE, DFL are tuned on validation data from the training distribution. We do the same when using semantic corruptions.

**Experimental details for Waterbirds** For the NURD setup, the training, validation, and test datasets have $3020, 756, 800$ samples respectively. We use a single architecture to parameterize the predictive model and the weight model in this experiment: two fully connected layers on top of a ResNet18 initialized at weights pretrained on Imagenet. We use the same training procedure for NURD with known nuisances or with semantic corruptions. Both models are trained with cross-entropy. The weight model is optimized with the default Adam optimizer for 20 epochs with a batch size of $64$. The predictive model is optimized with the Adam optimizer for 20 epochs with a learning rate of $0.0002$, a weight decay of $0.01$, and a batch size of $250$.

For the JTT setup, the training, validation, and test datasets have $4795, 1199, 5794$ samples respectively. For JTT, we use the same model and model parameters as Liu et al. (2021) using their released code. We repeat the details here for completeness. The model for both stages of JTT is a ResNet-50. Both models are optimized by stochastic gradient descent (SGD) with momentum $0.9$, weight decay

1.0, and learning rate $1 \times 10^{-5}$. Both models are trained for 300 epochs with batch size 64, using batch normalization and no data augmentation. The identification model used to select samples to upweight corresponds to epoch 60 and the upweighting constant is $\lambda = 100$.

**Experimental details for cardiomegaly detection.** The training, validation, and test datasets are fixed across seeds and have $18000, 2000, 1000$ samples respectively. To run reweighting-NURD, we use a single architecture to parameterize the predictive model and the weight model in this experiment: two fully connected layers on top of a ResNet18 initialized at weights pretrained on Imagenet. In known-nuisance NURD with the hospital as the nuisance, the biased model is an estimate of $p_{tr}(\mathbf{y} \mid \text{hospital})$, which is obtained by binning the samples based on the hospital and averaging the labels. We use the same training procedure for NURD with known nuisances or with semantic corruptions. Both weight and predictive models are trained with cross-entropy. The weight model and the predictive model are optimized with the Adam optimizer over 25 epochs with a batch size of 256, and learning rate 0.001.

**Implementation details for NLI** For POE and DFL, we build classifiers by fine-tuning a pretrained BERT model (Devlin et al., 2019) on the data. We follow the same training procedure and hyperparameter details as used in Mahabadi et al. (2019) — models were trained on the MNLI training dataset which consists of 392k examples, with a learning rate of $2 \times 10^{-5}$ with a batch size of 8 using the Adam Optimizer. All models are trained for 3 epochs. The development set contains 9815 examples and the HANS test contains 30000 examples. Since the HANS dataset has only two labels — 'entailment' and 'non-entailment' — we combine the neutral and contradiction classes during inference on HANS.

For the JTT setup, Liu et al. (2021) mix the training and development sets from MNLI and create their own training, validation, and test sets of sizes $206175, 82462, 123712$ respectively. For JTT, we use the same model and model parameters as Liu et al. (2021) using their released code. We use the optimal hyperparameters reported in Liu et al. (2021) for the learning rate, weight decay, and the upweighting constant. We repeat the details here for completeness. The model for both stages of JTT is a pretrained BERT model that is finetuned during training. Both models are optimized by the AdamW optimizer with clipping for the predictive model, no weight decay, and an initial learning rate of $2 \times 10^{-5}$. Both models are trained for 5 epochs with batch size 32 and dropout. The identification model used to select samples to upweight corresponds to epoch 2 for vanilla JTT (reported optimal in Liu et al. (2021)); for JTT with semantic corruption, we select one from $2, 3$ using validation group annotations. For both, the upweighting constant is $\lambda = 6$. Our runs with these parameters did not yield the test worst-group accuracy reported in (Liu et al., 2021) (72.6%); our experiments yielded a test worst-group accuracy 71.3%. We expect this may be due to the differences in the random seed; JTT is sensitive to hyperparameters and differences in order of batches may result in drops in performance.

In NGRAM-RAND, when the number of words in the sentence is not a multiple of $n$, there will be one $k$-gram ($k < n$). In implementing NGRAM-RAND, we ensure that the position of this k-gram is randomized i.e. we make sure that it does not always occur at the end of the sentence, for example. NGRAM-RAND is implemented before word-piece tokenization (which BERT uses), to ensure that we randomize words instead of subwords.

We also create a small HANS-like development set, which can be optionally used to tune the size parameter. This set is constructed by randomly sampling 1000 examples from the HANS training set, which does not have any overlap with the main HANS test set.

**Table 7:** Mean and standard error of test accuracy across 10 seeds of NURD using with semantic corruptions on classifying waterbirds. *Known*-nuisance NURD uses a label for the type of background as the nuisance. Selecting sizes in PATCH-RAND (PR) and ROI-MASK (RM) based on a small evaluation set from the test distribution gives PR-size 14 and RM-size 196. With these sizes NURD with RM and PR close 99% and 93% of the gap between ERM and known-nuisance NURD respectively.

| | *known* $\mathbf{z}$ | RM 196 | RM 168 | RM 140 | RM 112 | PR 7 | PR 14 | PR 28 | PR 56 | ERM |
|---|---|---|---|---|---|---|---|---|---|---|
| Mean | 87.2% | 87.0% | 87.6% | 85.1% | 85.5% | 82.6% | 86.0% | 83.9% | 80.4% | 69.2% |
| Std. err. | 1.0% | 1.0% | 1.8% | 1.0% | 1.6% | 2.2% | 1.5% | 1.2% | 2.0% | 2.1% |

### B.2 FULL RESULTS TABLES

We reproduce the results tables along with the uncertainty estimates; see table 7, table 8, and table 9. To report the same metrics as in Mahabadi et al. (2019) for POE and DFL and Puli et al. (2022) for NURD, we report standard error for NURD and standard deviation for POE and DFL.

**Table 8:** Mean and standard error of test accuracy across 10 seeds of NURD using with semantic corruptions on detecting cardiomegaly from chest X-rays. *Known*-nuisance NURD uses the hospital as the nuisance. Selecting sizes in PATCH-RAND (PR) and ROI-MASK (RM) based on a small evaluation set from the test distribution gives PR-size 7 and RM-size 196. With these sizes, NURD with RM and PR close 77% and 79% of the gap between ERM and known-nuisance NURD respectively.

| | *known* **z** | RM 196 | RM 168 | RM 140 | RM 112 | PR 7 | PR 14 | PR 28 | PR 56 | ERM |
|---|---|---|---|---|---|---|---|---|---|---|
| Mean | 81.7% | 77.2% | 75.0% | 74.4% | 68.9% | 77.5% | 75.2% | 72.2% | 69.2% | 62% |
| Std. err. | 0.3% | 0.8% | 0.9% | 1.3% | 1.6% | 0.6% | 0.8% | 1.3% | 2.3% | 2.0% |

**Table 9:** Average accuracies and standard deviation over 4 seeds of POE and DFL with semantic corruptions on the HANS dataset. We report the performances of POE and DFL from Mahabadi et al. (2019) as *known*, where both methods use known nuisances. For both methods, selecting the size hyperparameter based on the average accuracy on a small evaluation dataset (1000 samples) from the test distribution gives $n = 3$. With this size, POE with NGRAM-RAND performs better than known-nuisance POE while DFL with NGRAM-RAND closes 84% of the gap between ERM and known-nuisance DFL.

| **z** | POE | DFL |
|---|---|---|
| *Known* | $66.3 \pm 0.6\%$ | $69.3 \pm 0.2\%$ |
| 1-gram | $65.7 \pm 2.0\%$ | $66.5 \pm 1.5\%$ |
| 2-gram | $66.0 \pm 0.9\%$ | $68.5 \pm 0.7\%$ |
| 3-gram | $66.7 \pm 1.5\%$ | $68.4 \pm 1.5\%$ |
| 4-gram | $66.2 \pm 2.9\%$ | $65.0 \pm 2.0\%$ |
| ERM | – | 63.6%. |