Predict then Interpolate: A Simple Algorithm to Learn Stable Classifiers

Yujia Bao [1] **Shiyu Chang** [2] **Regina Barzilay** [3]

Abstract

We propose Predict then Interpolate (PI), a simple algorithm for learning correlations that are stable across environments. The algorithm follows from the intuition that when using a classifier trained on one environment to make predictions on examples from another environment, its mistakes are informative as to which correlations are unstable. In this work, we prove that by interpolating the distributions of the correct predictions and the wrong predictions, we can uncover an oracle distribution where the unstable correlation vanishes. Since the oracle interpolation coefficients are not accessible, we use group distributionally robust optimization to minimize the worst-case risk across all such interpolations. We evaluate our method on both text classification and image classification. Empirical results demonstrate that our algorithm is able to learn robust classifiers (outperforms IRM by 23.85% on synthetic environments and 12.41% on natural environments). Our code and data are available at `https://github.com/YujiaBao/Predict-then-Interpolate`.

1. Introduction

Distributionally robust optimization (DRO) alleviates model biases by minimizing the worst-case risk over a set of human-defined groups. However, in order to construct these groups, humans must identify and annotate these biases, a process as expensive as annotating the label itself (Ben-Tal et al., 2013; Duchi & Namkoong, 2018 ; Sagawa et al., 2019). In this paper we propose a simple algorithm to create groups that are informative of these biases, and use these groups to train stable classifiers.

Our algorithm operates on data split among multiple environments, across which correlations between bias features and the label may vary. Instead of handcrafting environments based on explicit, task-dependent biases, these environments can be determined by generic information that is easy to collect (Peters et al., 2015). For example, environments can represent data collection circumstances, like location and time. Our goal is to learn correlations that are stable across these environments.

Given these environments, one could directly use them as groups for DRO. Doing so would optimize the worst-case risk over all interpolations of the training environments. However, if the unstable (bias) features are positively *and* differentially correlated with the label in all training environments, the unstable correlation will be positive in any of their interpolations. DRO, optimizing for the best worst-case performance, will inevitably exploit these unstable features, and we fail to learn a stable classifier.

In this work, we propose Predict then Interpolate (PI), a simple recipe for creating groups whose interpolation yields a distribution with only stable correlations. Our idea follows from the intuition that when using a classifier trained on one environment to make predictions on examples from a different environment, its mistakes are informative of the unstable correlations. In fact, we can prove that if the unstable features and the label are positively correlated across all environments, the same correlation flips to negative in the set of mistakes. Therefore, by interpolating the distributions of correct and incorrect predictions, we can uncover an "oracle" distribution in which only stable features are correlated with the label. Although the oracle interpolation coefficients are not accessible, we can minimize the worst-case risk over all interpolations, providing an upper bound of the risk on the oracle distribution.

Our learning paradigm consists of three steps. First, we train an individual classifier for each environment to estimate the conditional distribution of the label given the input. These classifiers are biased, as they may rely on any correlations in the dataset. Next, we apply each environment's classifier to partition all other environments, based on prediction correctness. Finally, we obtain our robust classifier by minimizing the worst-case risk over all interpolations of the partitions.

Empirically, we evaluate our approach on both synthetic and real-world environments. First, we simulate unstable correlations in synthetic environments by appending spurious

[1]MIT CSAIL [2]MIT-IBM Watson AI Lab [3]MIT CSAIL. Correspondence to: Yujia Bao <yujia@csail.mit.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

features. Our results in both digit classification and aspect-level sentiment classification demonstrate that our method delivers significant performance gain (23.85% absolute accuracy) over invariant risk minimization (IRM), approaching oracle performance. Quantitative analyses confirm that our method generates partitions with opposite unstable correlations. Next, we applied our approach on natural environments defined by an existing attribute of the input. Our experiments on CelebA and ASK2ME showed that directly applying DRO on environments improves robust accuracy for known attributes, but this robustness doesn't generalize equally across other attributes that are unknown during train time. On the other hand, by creating partitions with opposite unstable correlations, our method is able to improve average worst-group accuracy by 12.41% compared to IRM.

2. Related work

Removing known biases: Large scale datasets are fraught with biases. For instance, in face recognition (Liu et al., 2015a), spurious associations may exist between different face attributes (e.g. hair color) and demographic information (e.g. ethnicity) (Buolamwini & Gebru, 2018). Furthermore, in natural language inference (Bowman et al., 2015), the entailment label can often be predicted from lexical overlap of the two inputs (McCoy et al., 2019). Finally, in molecular property prediction (Wu et al., 2018; Mayr et al., 2018), performance varies significantly across different scaffolds (Yang et al., 2019).

Many approaches have been proposed to mitigate biases when they are known beforehand. Examples include adversarial training to remove biases from representations (Belinkov et al., 2019; Stacey et al., 2020), re-weighting training examples (Schuster et al., 2019), and combining a biased model and the base model's predictions using a product of experts (Hinton, 2002; Clark et al., 2019; He et al., 2019; Mahabadi et al., 2020). These models are typically designed for a specific type of bias and thus require extra domain knowledge to generalize to new tasks.

Group DRO is another attractive framework since it allows explicit modeling of the distribution family that we want to optimize over. Previous work (Hu et al., 2018; Oren et al., 2019; Sagawa* et al., 2020) has shown the effectiveness of group DRO to train un-biased models. In these models, the groups are specified by human based on the knowledge of the bias attributes. Our work differs from them as we create groups using trained models. This allows us to apply group DRO when we don't have annotations for the bias attributes. Moreover, when the bias attributes are available, we can further refine our groups to reduce unknown biases.

Removing unknown biases: Determining dataset biases is time-consuming and often requires task-specific expert knowledge (Zellers et al., 2019; Sakaguchi et al., 2020). Thus, there are two lines of work that aim to build robust models without explicitly knowing the type of bias. The first assumes that weak models, which have limited capacity (Sanh et al., 2021) or are under-trained (Utama et al., 2020), are more prone to rely on shallow heuristics and rediscover previously human-identified dataset biases. By learning from the weak models' mistakes, we can obtain a more robust model. While these methods show empirical benefits on some NLP tasks, the extent to which their assumption holds is unclear. In fact, recent work (Sagawa et al., 2020) shows that over-parametrization may actually exacerbate unstable correlations for image classification.

The second line of work assumes that the training data are collected from separate environments, across which unstable features exhibit different correlations with the label (Peters et al., 2016; Krueger et al., 2020; Chang et al., 2020; Jin et al., 2020; Ahuja et al., 2020; Arjovsky et al., 2019). Invariant risk minimization (Arjovsky et al., 2019), a representative method along this line, learns representations that are simultaneously optimal across all environments. However, since this representation is trained across all environments, it can easily degenerate in real-world applications (Gulrajani & Lopez-Paz, 2020). One can consider an extreme case where the learned representation directly encodes the one-hot embedding of the label. While this learned representation is stable (invariant) according to the definition, the model can utilize *any unstable features* to generate this representation. We have no guarantee on how the model would generalize when the unstable correlations vanish.

Our algorithm instead decomposes the problem of learning stable classifiers into two parts: finding *unstable features* and training a robust model. By constraining the classifiers to be environment-specific in the first part, we are able to construct an oracle distribution where the unstable features are not correlated with the label. Our model then directly optimizes an upper bound of the risk on this oracle distribution. Empirically, we demonstrate that our method is able to eliminate biases not given during training on multiple real-world applications.

3. Method

We consider the standard setting (Arjovsky et al., 2019) where the training data are comprised of n environments $\mathcal{E} = \{E_1, \ldots, E_n\}$. For each environment E_i, we have input-label pairs $(x, y) \overset{\text{iid}}{\sim} P_i$. Our goal is to learn correlations that are *stable* across these environments (Woodward, 2005) so that the model can generalize to a new test environment E_{test} that has the same stable correlations.

3.1. Algorithm

Our intuition follows from a simple question.

What happens if we apply a classifier f_i trained on environment E_i to a different environment E_j?

Suppose we have enough data in E_i and the classifier f_i is able to perfectly fit the underlying conditional $P_i(y|x)$. Since E_i and E_j follow different distributions, the classifier f_i will make mistakes on E_j. These mistakes are natural products of the unstable correlation: if the correlation of the unstable feature is higher in E_i than in E_j, the classifier f_i will overuse this feature when making predictions in E_j.

In fact, we can show that under certain conditions, the unstable correlation within the subset of wrong predictions is opposite of that within the subset of correct predictions (Section 3.3). By interpolating between these two subsets, we can uncover an *oracle distribution* where the label is not correlated with the unstable feature. Since this interpolation coefficient is not accessible in practice, we adopt group DRO to minimize the worst-case risk over all interpolations of these subsets. This provides us an upper bound of the risk on the oracle distribution.

Concretely, our approach has three successive stages.

Stage 1: For each environment E_i, train an environment specific classifier f_i.

Stage 2: For each pair of environments E_i and E_j, use the trained classifier f_i to partition E_j into two sets

$$E_j = E_j^{i\checkmark} \cup E_j^{i\times}$$

where $E_j^{i\checkmark}$ contains examples that f_i predicted correctly and $E_j^{i\times}$ contains those predicted incorrectly.

Stage 3: Train the final model f by minimizing the worst-case risk over the set of all interpolations $\mathcal{Q}$:

$$\mathcal{Q} = \left\{ \sum_{i \neq j} \lambda_j^{i\checkmark} P_j^{i\checkmark} + \lambda_j^{i\times} P_j^{i\times} : \sum_{i \neq j} \lambda_j^{i\checkmark} + \lambda_j^{i\times} = 1 \right\},$$

where $P_j^{i\checkmark}$ and $P_j^{i\times}$ are the empirical distributions of $E_j^{i\checkmark}$ and $E_j^{i\times}$. Because the optimum value of a linear program must occur at a vertex, the worst-case risk over $\mathcal{Q}$ is equivalent to the maximum expected risk across all groups. This allows us to formulate the objective as a min-max problem:

$$\min_f \max_{P \in \mathcal{P}} \mathbb{E}_{(x,y)\sim P}[\mathcal{L}(x, y; f)],$$

where $\mathcal{L}(x, y; f)$ is the loss of the model f and $\mathcal{P}$ is the set of distributions $\{P_j^{i\checkmark}\}_{i \neq j} \cup \{P_j^{i\times}\}_{i \neq j}$.

Extensions of the algorithm: For regression tasks, we can set a threshold on the mean square error to partition environments. We can also apply the first two stages multiple times, treating new partitions as different environments, to iteratively refine the groups. In this work, we focus on the basic setting and leave the rest for future work.

3.2. A toy example

To understand the behavior of the algorithm, let's consider a simple example with two environments E_1 and E_2 (Figure 1). For each environment, the data are generated by the following process.[1]

- First, sample the feature $x_1 \in \{0, 1\}$ which takes the value 1 with probability 0.5. This is our stable feature.
- Next, sample the observed noisy label $y \in \{0, 1\}$ by flipping the value of x_1 with probability 0.2.
- Finally, for each environment E_i, sample the unstable feature $x_2 \in \{0, 1\}$ by flipping the value of y with probability η_i. Let $\eta_1 = 0$ and $\eta_2 = 0.1$.

Our goal is to learn a classifier that only uses feature x_1 to predict y. Since the unstable feature x_2 is positively correlated with the label across both environments, directly treating the environments as groups and applying group DRO will also exploit this correlation during training.

Let's take a step back and consider a classifier f_1 that is trained only on E_1. Since x_2 is identical to y and x_1 differs from y with probability 0.2, f_1 simply learns to ignore x_1 and predict y as x_2 (Figure 1a). When we apply f_1 to the other environment E_2, it will make mistakes on examples where x_2 is flipped from y. Moreover, we can check that the correlation coefficient between the unstable feature x_2 and y is 1 in the set of correct predictions $E_2^{1\checkmark}$ and it flips to -1 in the set of mistakes $E_2^{1\times}$ (Figure 1b). In this toy example, the *oracle distribution* P^*, where the correlation between x_2 and y is 0, can be obtained by simply averaging the empirical distribution of the two subsets (Figure 1c):

$$P^*(x_1, x_2, y) = 0.5P_2^{1\checkmark}(x_1, x_2, y) + 0.5P_2^{1\times}(x_1, x_2, y).$$

We can also verify that the optimal solution that minimize the worst-case risk across $E_2^{1\checkmark}$ and $E_2^{1\times}$ is to predict y only using x_1. (Appendix A).

Remark 1: In the algorithm, we also use the classifier f_2 trained on E_2 to partition E_1. The final model f is obtained by minimizing the worst-case risk over $P_1^{2\checkmark}, P_1^{2\times}, P_2^{1\checkmark}, P_2^{1\times}$.

[1] Arjovsky et al. (2019) used this process to construct the Colored-MNIST dataset.

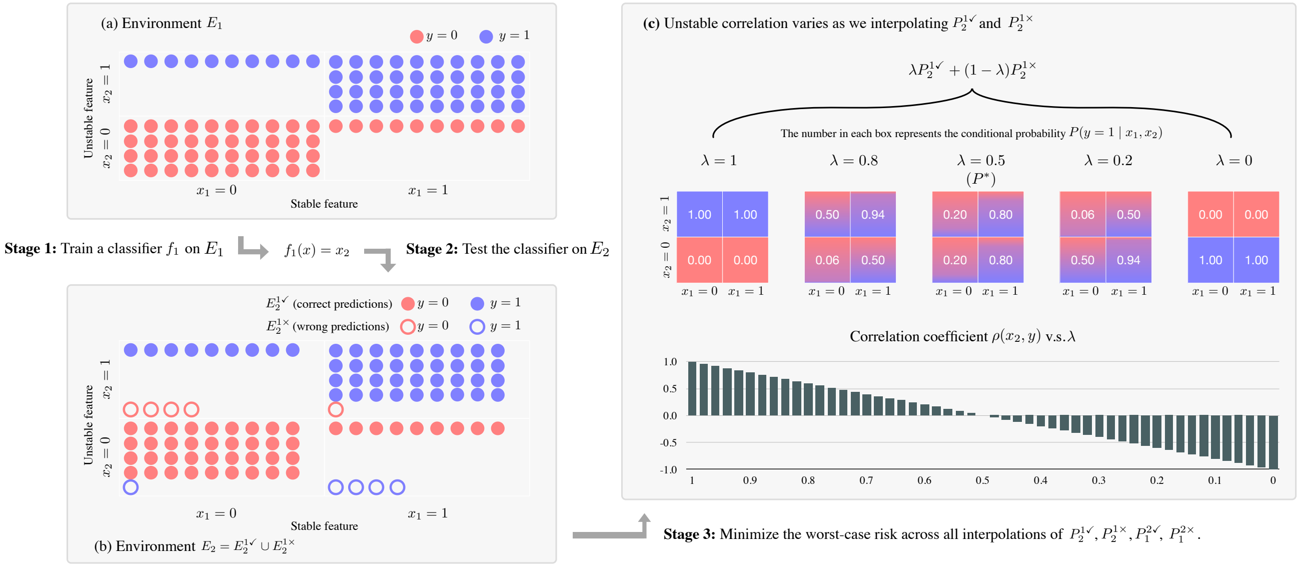
(a) Environment E_1
● $y=0$ ● $y=1$
Unstable feature
$x_2 = 1$
$x_2 = 0$
$x_1 = 0$
$x_1 = 1$
Stable feature
Stage 1: Train a classifier f_1 on E_1
$f_1(x) = x_2$
Stage 2: Test the classifier on E_2
$E_2^{1\checkmark}$ (correct predictions) ● $y=0$ ● $y=1$
$E_2^{1\times}$ (wrong predictions) ○ $y=0$ ○ $y=1$
Unstable feature
$x_2 = 1$
$x_2 = 0$
$x_1 = 0$
$x_1 = 1$
Stable feature
(b) Environment $E_2 = E_2^{1\checkmark} \cup E_2^{1\times}$
(c) Unstable correlation varies as we interpolating $P_2^{1\checkmark}$ and $P_2^{1\times}$
$\lambda P_2^{1\checkmark} + (1-\lambda)P_2^{1\times}$
The number in each box represents the conditional probability $P(y=1 \mid x_1, x_2)$
$\lambda = 1$
1.00 1.00
0.00 0.00
$\lambda = 0.8$
0.50 0.94
0.06 0.50
$\lambda = 0.5$
(P^*)
0.20 0.80
0.20 0.80
$\lambda = 0.2$
0.06 0.50
0.50 0.94
$\lambda = 0$
0.00 0.00
1.00 1.00
$x_2 = 1$
$x_2 = 0$
$x_1 = 0$ $x_1 = 1$
Correlation coefficient $\rho(x_2, y)$ v.s. λ
1.0
0.5
0.0
-0.5
-1.0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Stage 3: Minimize the worst-case risk across all interpolations of $P_2^{1\checkmark}, P_2^{1\times}, P_1^{2\checkmark}, P_1^{2\times}$.

Figure 1. Illustration of our algorithm on the toy example. The label y agrees with the stable feature x_1 with probability 0.8 on both environments. For the unstable feature x_2, the probability of $x_2 = y$ is 1.0 in E_1 and 0.9 in E_2. Stage 1: We train a classifier f_1 on E_1. It learns to make predictions solely based on the unstable feature x_2. Stage 2: We use f_1 to partition E_2 based on the prediction correctness. While the correlation of x_2 is positive for both E_1 and E_2, it flips to negative in set of wrong predictions $E_2^{1\times}$. Stage 3: Interpolating $P_2^{1\checkmark}$ and $P_2^{1\times}$ allows us to uncover an oracle distribution P^* where the unstable feature x_2 is not correlated with the label. Note that here we only illustrate how to partition E_2 using f_1. In our algorithm, we also use the classifier f_2 (trained on E_2) to partition E_1, and the final model f is obtained by minimizing the worst-case risk over all interpolations of $P_1^{2\checkmark}, P_1^{2\times}, P_2^{1\checkmark} P_2^{1\times}$.

Remark 2: Our algorithm optimizes an *upper bound* of the risk on the oracle distribution. In general, it *doesn't guarantee* that the unstable correlation is not utilized by the model when the worst-case performance is not achieved at the oracle distribution.

3.3. Theoretical analysis

In the previous example, we have seen that the unstable correlation flips in the set of mistakes $E_2^{1\times}$ compared to the set of correct predictions $E_2^{1\checkmark}$. Here, we would like to investigate how this property holds in general.[2] We focus our analysis on binary classification tasks where $y \in \{0, 1\}$. Let x_1 be the stable feature and x_2 be unstable feature that has various correlations across environments. We use capital letters X_1, X_2, Y to represent random variables and use lowercase letters x_1, x_2, y to denote their specific values.

Proposition 1. *For a pair of environments E_i and E_j, assuming that the classifier f_i is able to learn the true conditional $P_i(Y \mid X_1, X_2)$, we can write the joint distribution P_j of E_j as the mixture of $P_j^{i\checkmark}$ and $P_j^{i\times}$:*

$$P_j(x_1, x_2, y) = \alpha_j^i P_j^{i\checkmark}(x_1, x_2, y) + (1-\alpha_j^i) P_j^{i\times}(x_1, x_2, y),$$

where $\alpha_j^i = \sum_{x_1, x_2, y} P_j(x_1, x_2, y) \cdot P_i(y \mid x_1, x_2)$ *and*

$$P_j^{i\checkmark}(x_1, x_2, y) \propto P_j(x_1, x_2, y) \cdot P_i(y \mid x_1, x_2),$$
$$P_j^{i\times}(x_1, x_2, y) \propto P_j(x_1, x_2, y) \cdot P_i(1-y \mid x_1, x_2).$$

Intuitively, when partitioning the environment E_j, we are scaling its joint distribution based on the conditional on E_i.

Two degenerate cases: From Proposition 1, we see that the algorithm degenerates when $\alpha_j^i = 0$ (predictions of f_i are all wrong) or $\alpha_j^i = 1$ (predictions of f_i are all correct). The first case occurs when the unstable correlation is flipped between P_i and P_j. One may think about setting $\eta_1 = 0$ and $\eta_2 = 1$ in the toy example. In this case, we can obtain the oracle distribution by directly interpolating P_i and P_j. The second case implies that the conditional is the same across the environments: $P_i(Y \mid X_1, X_2) = P_j(Y \mid X_1, X_2)$. Since x_2 is the unstable feature, this equality holds when the conditional mutual information between X_2 and Y is zero given X_1, i.e., $P_i(Y \mid X_1, X_2) = P_i(Y \mid X_1)$. In this case, f_i already ignores the unstable feature x_2.

To carryout the following analysis, we assume that the marginal distribution of Y is uniform in all joint distributions, i.e., f_i performs equally well on different labels.

Theorem 1. *Suppose X_2 is independent of X_1 given Y. For any environment pair E_i and E_j, if $\sum_y P_i(x_2 \mid y) = \sum_y P_j(x_2 \mid y)$ for any x_2, then* $\mathrm{Cov}(X_2, Y; P_i) > \mathrm{Cov}(X_2, Y; P_j)$ *implies* $\mathrm{Cov}(X_2, Y; P_j^{i\times}) < 0$ *and* $\mathrm{Cov}(X_2, Y; P_i^{j\times}) > 0$.

The result follows from the connection between the covariance and the conditional. On one side, the covariance be-

[2] All proofs are relegated to Appendix B.

tween x_2 and Y captures the difference of their conditionals: $P(X_2 \mid Y=1) - P(X_2 \mid Y=0)$, On the other side, the conditional independence assumption allows us to factorize the joint distribution: $P_i(x_1, x_2, y) = P_i(x_1, y)P_i(x_2 \mid y)$. Combining them together finishes the proof.

Theorem 1 tells us no matter whether the spurious correlation is positive or negative, we can obtain an oracle distribution P^*, $\mathrm{Cov}(X_2, Y; P^*) = 0$ by interpolating across $P_j^{i\checkmark}$, $P_j^{i\times}$, $P_i^{j\checkmark}$, $P_i^{j\times}$. By optimizing the worst-case risk across all interpolations, our final model f provides an *upper bound* of the risk on the oracle distribution P^*.

We also note that the toy example in Section 3.2 is a special case of the assumption in Theorem 1. While many previous work also construct datasets with this assumption (Arjovsky et al., 2019; Choe et al., 2020), it may be too restrictive in practice. In the general case, although we cannot guarantee the sign of the correlation, we can still obtain an upper bound for $\mathrm{Cov}(X_2, Y; P_j^{i\times})$ and a lower bound for $\mathrm{Cov}(X_2, Y; P_i^{j\times})$:

Theorem 2. *For any environment pair E_i and E_j, $\mathrm{Cov}(X_2, Y; P_i) > \mathrm{Cov}(X_2, Y; P_j)$ implies*

$$\begin{aligned}
&\mathrm{Cov}(X_2, Y; P_j^{i\times}) \\
&< \frac{1-\alpha_j^i}{\alpha_i^i}\mathrm{Cov}(X_2, Y; P_i^{i\checkmark}) - \frac{1-\alpha_j^i}{\alpha_j^i}\mathrm{Cov}(X_2, Y; P_j^{i\checkmark}) \\
&\mathrm{Cov}(X_2, Y; P_i^{j\times}) \\
&> \frac{1-\alpha_i^j}{\alpha_j^j}\mathrm{Cov}(X_2, Y; P_j^{j\checkmark}) - \frac{1-\alpha_i^j}{\alpha_i^j}\mathrm{Cov}(X_2, Y; P_i^{j\checkmark})
\end{aligned}$$

where $P_i^{i\checkmark}$ is the distribution of the correct predictions when applying f_i on E_i.

Intuitively, if the correlation is stronger in E_i, then the classifier f_i will overuse this correlation and make mistakes on E_j when this stronger correlation doesn't hold. Conversely, the classifier f_j will underuse this correlation and make mistakes on E_i when the correlation is stronger.

4. Experimental setup

4.1. Datasets and Settings

Synthetic environments: To assess the empirical behavior of our algorithm, we start with controlled experiments where we can simulate spurious correlation. We consider two standard datasets: MNIST (LeCun et al., 1998) and BeerReview (McAuley et al., 2012).[3]

For MNIST, we adopt Arjovsky et al. (2019)'s approach for generating spurious correlation and extend it to a more challenging multi-class problem. For each image, we sample y, which takes on the same value as its numeric digit with 0.75 probability and a uniformly random other digit with the remaining probability. The spurious feature in sampled in a similar way: it takes on the same value as y with η probability and a uniformly random other value with the remaining probability. We color the image according to the value of the spurious feature. We set η to 0.9 and 0.8 respectively for the training environments E_1 and E_2. In the testing environment, η is set to 0.1.

For BeerReview, we consider three aspect-level sentiment classification tasks: look, aroma and palate (Lei et al., 2016; Bao et al., 2018). For each review, we append an artificial token (`art_pos` or `art_neg`) that is spuriously correlated with the binary sentiment label (`pos` or `neg`). The artificial token agrees with the sentiment label with probability 0.9 in environment E_1 and with probability 0.8 in environment E_2. In the testing environment, the probability reduces to 0.1. Unlike MNIST, here we do not inject artificial label noise to the datasets.

Validation set plays a crucial role when the training distribution is different from the testing distribution (Gulrajani & Lopez-Paz, 2020). For both datasets, we consider two different validation settings and report their performance separately: 1) sampling the validation set from the training environment; 2) sampling the validation set from the testing environment.

Natural environments: We also consider a practical setting where environments are naturally defined by some attributes of the input and we want to use them to reduce biases that are *unknown* during training and validation. We study two datasets: CelebA (Liu et al., 2015b) where the attributes are annotated by human and ASK2ME (Bao et al., 2019) where the attributes are automatically generated by rules.

CelebA is an image classification dataset where each input image (face) is paired with 40 binary attributes. We adopt Sagawa et al. (2019)'s setting and treat hair color ($y \in \{\texttt{blond}, \texttt{dark}\}$) as the target task. We use the gender attribute to define the two training environments, E_1=$\{\texttt{female}\}$ and E_2=$\{\texttt{male}\}$. Our goal is to learn a classifier that is robust to other unknown attributes such as `wearing_hat`. For model selection, we partition the validation data into four groups based on the gender value and the label value: $\{\texttt{female}, \texttt{blond}\}$, $\{\texttt{female}, \texttt{dark}\}$, $\{\texttt{male}, \texttt{blond}\}$, $\{\texttt{male}, \texttt{dark}\}$. We use the worst-group accuracy as our validation criteria.

ASK2ME is a text classification dataset where an input text (paper abstract from PubMed) is paired with 17 binary attributes, each indicating the presence of a different disease. The task is to predict whether the input is informative about the *risk* of cancer for gene

[3] All dataset statistics are relegated to Appendix C.1.

	ERM		DRO		IRM		RGM		PI (OURS)		ORACLE	
$P_{\text{val}} = P_{\text{test}}$?	✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
MNIST	26.15	14.25	32.51	21.06	45.41	13.13	42.49	15.33	**69.44**	**69.68**	71.40	71.60
Beer Look	64.63	60.96	64.53	62.75	65.83	63.31	66.31	61.51	**78.09**	**70.66**	80.32	73.51
Beer Aroma	55.25	51.99	57.08	53.39	60.25	53.25	66.33	57.91	**77.01**	**67.35**	77.34	69.99
Beer Palate	49.01	46.69	47.72	46.35	66.45	44.09	68.77	44.81	**74.14**	**61.51**	74.89	66.33

Table 1. Accuracy of different methods on image classification (majority baseline 10%) and aspect-level sentiment classification (majority baseline 50%). All methods are tuned based on a held-out validation set. We consider two validation settings: 1) sample the validation set from the testing environment ($P_{\text{val}} = P_{\text{test}}$); 2) sample the validation set from the training environment.

mutation carriers, rather than cancer itself (Deng et al., 2019). We define two training environments based on the breast_cancer attribute, E_1={breast_cancer=0} and E_2={breast_cancer=1}. We would like to see whether the classifier is able to remove spurious correlations from other diseases that are unknown during training. Similar to CelebA, we compute the worst-group accuracy based on the breast_cancer value and the label value and use it for validation.

At test time, we evaluate the classifier's prediction robustness on all attributes over a held-out test set. For each attribute, we report the worst-group accuracy and the average-group accuracy.

4.2. Baselines

We compare our algorithm against the following baselines:

ERM: We combine all environments together and apply standard empirical risk minimization.

IRM: Invariant risk minimization (Arjovsky et al., 2019) learns a representation such that the linear classifier on top of this representation is simultaneously optimal across different environments.

RGM: Regret minimization (Jin et al., 2020) simulates unseen environments by using part of the training set as held-out environments. It quantifies the generalization ability in terms of regret, the difference between the losses of two auxiliary predictors trained with and without examples in the current environment.

DRO: We can also apply DRO on groups defined by the environments and the labels. For example, in beer review, we can partition the training data into the four groups: {pos, E_1}, {neg, E_1} {pos, E_2}, {neg, E_2}. Minimizing the worst-case performance over these human-defined groups has shown success in improving model robustness (Sagawa et al., 2019).

Oracle: In the synthetic environments, we can use the spurious features to define groups and train an oracle DRO model. For example, in beer review, the oracle model will minimize the worst-case risk over the four groups: {pos, art_pos}, {pos, art_neg} {pos, art_pos}, {pos, art_neg}. This helps us analyze the contribution of our algorithm in isolation of the inherent limitations of the task.

	E_1	E_2	$E_2^{1\checkmark}$	$E_2^{1\times}$
MNIST	0.8955	0.7769	0.9961	−0.1040
Beer Look	0.8007	0.6006	0.8254	−0.8030
Beer Aroma	0.8007	0.6006	0.9165	−0.9303
Beer Palate	0.8007	0.6006	0.9394	−0.9189

Table 2. Pearson correlation coefficient between the spurious feature and the label across four datasets. While the correlation is positive for both training environments, it flips to negative in the set of wrong predictions $E_2^{1\times}$. Interpolating across $E_2^{1\checkmark}$ and $E_2^{1\times}$ allows us to remove the unstable correlation.

For fair comparison, all methods share the same model architecture.[4] Implementation details can be found in Appendix C.2.

5. Results

5.1. Synthetic environments

Table 1 summarizes the results on synthetic environments. As we expected, since the signs of the unstable correlation are the same across the training environments, both ERM and DRO exploit this information and fail to generalize when it changes in the testing environment. While IRM and RGM are able to learn stable correlations when we use the testing environment for model selection, their performance quickly drop to that of ERM when the validation data is drawn from the training environment, which also backs up the claim from Gulrajani & Lopez-Paz (2020).

Our algorithm obtains substantial gains across four tasks

[4] For IRM and RGM, in order to tune the weights and annealing strategy for the regularizer, the hyper-parameter search space is 21× larger than other methods.

	ERM		DRO		IRM		RGM		PI (OURS)	
	Worst	Avg	Worst	Avg	Worst	Avg	Worst	Avg	Worst	Avg
Kidney cancer	16.67	66.66	**50.00**	68.70	33.33	73.07	33.33	71.31	**50.00**	**74.76**
Adenocarcinoma	33.33	72.91	77.29	79.23	55.56	78.40	55.56	78.12	**80.24**	**84.74**
Lung cancer	44.44	74.76	62.50	74.58	38.89	74.28	50.00	74.75	**70.31**	**78.85**
Polyp syndrom	44.44	74.63	**77.29**	78.79	55.56	76.31	66.67	78.73	69.23	**81.28**
Hepatobiliary cancer	44.44	73.01	**60.00**	73.89	55.56	77.19	55.56	76.22	**60.00**	**78.94**
Breast cancer	66.49	80.47	75.00	78.84	66.87	80.56	64.38	79.81	**80.32**	**83.12**
Average*	54.80	77.73	67.34	77.58	57.86	79.18	60.81	79.03	**74.15**	**83.15**

Table 3. Worst-group and average-group accuracy on ASK2ME text classification. We show the results for the worst 5 attributes (sorted based on ERM) and the given attribute breast_cancer. Average is computed based on the performance across all attributes. See Appendix D for full results.

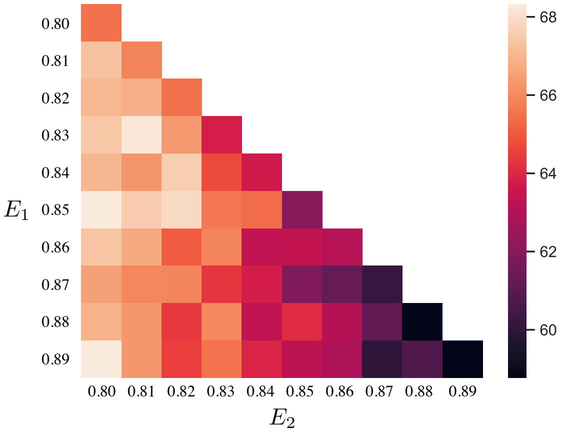

Figure 2. The ability of generalization changes as we vary the gap between the training environments. The x and y axes denote the probabilities that the injected artificial token agrees with the label. Heatmap corresponds to the testing accuracy for Beer Aroma.

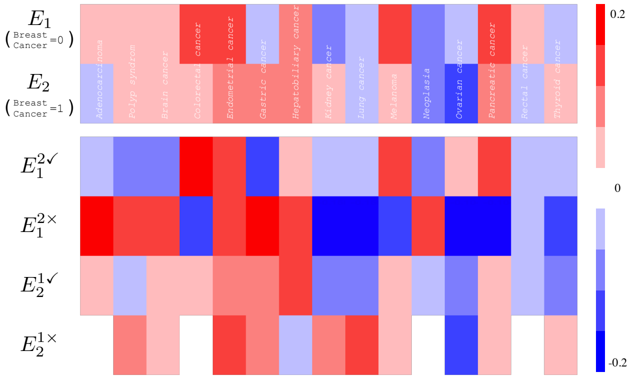

Figure 3. Visualization of the Pearson correlation coefficient between the label and the attribute on ASK2ME. Each column corresponds to a different attribute. We observe that correlations vary for inputs with different breast_cancer value. Our algorithm utilizes this difference to create partitions with opposite correlations (red vs. blue) so that we can uncover an oracle distribution (different for each attribute) by interpolating these partitions.

It performs much more stable under different validation settings. Specifically, comparing against the best baseline, our algorithm improves the accuracy by 20.06% when the validation set is drawn from the training environment and 12.97% when it is drawn from the testing environment. Its performance closely matches the oracle model with only 2% difference on average.

Why does partitioning the training environments help? To demystify the huge performance gain, we quantitatively analyze the partitions created by our algorithm in Table 2.[5] We see that while the unstable correlation is positive in both training environments, it flips to negative in the set of wrong predictions, confirming our theoretical analysis. In order to perform well across all partitions, our final classifier learns not to rely on the unstable features.

Do we need different training environments? We study the relation between the diversity of the training environments and the performance of the classifier on the beer review dataset. Specifically, we consider 45 different training environment pairs where we vary the probability that the artificial token agrees with the label from 0.80 to 0.89. We observe that the classifier performs better as we reduce the amount of spurious correlations (moving up along the diagonal in Figure 2). The classifier also generalizes better when we increase the gap between the two training environments (moving from right to left in Figure 2). In fact, when the training environments share the same distribution, the notion of stable correlation and unstable correlation is undefined. There is no signal for the algorithm to distinguish between spurious features and features that generalize.

[5] The partitions only depend on the training environments. It is independent to the choice of the validation data.

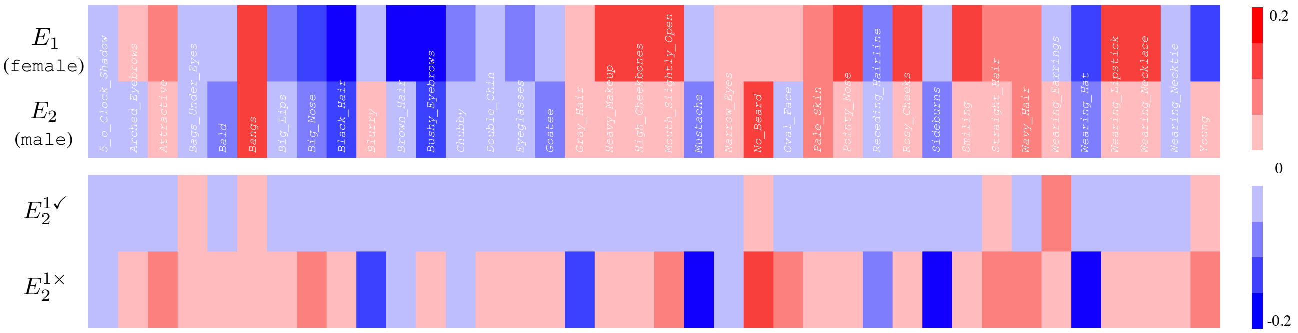

Figure 4. Visualization of the Pearson correlation coefficient between the label (hair color) and other attributes on CelebA. Each column corresponds to a different attribute. Due to the huge difference between the label marginals, $P_1(\texttt{blond}) = 0.24$ vs. $P_2(\texttt{blond}) = 0.02$, classifier f_2 predicts every example in environment E_1 as `dark`. The resulting partition, $E_1^{2\times} = \{\texttt{female, blond}\}$ and $E_1^{2\checkmark} = \{\texttt{female, dark}\}$, coincides with the human-defined groups in DRO. On the other hand, classifier f_1 is able to partition environment E_2 with opposite correlations (red vs. blue).

	ERM		DRO		IRM		RGM		PI (OURS)	
	Worst	Avg	Worst	Avg	Worst	Avg	Worst	Avg	Worst	Avg
Goatee	0.00	68.26	0.00	70.08	84.80	90.83	91.50	**95.66**	**91.63**	94.59
Wearing_Hat	7.69	70.39	46.15	82.28	46.15	77.34	**61.54**	**86.26**	53.85	84.56
Chubby	9.52	70.69	61.90	84.63	**76.19**	82.91	47.62	82.32	71.43	**86.59**
Wearing_Necktie	25.00	74.52	90.00	91.56	80.00	84.31	35.00	79.32	**91.44**	**92.53**
Sideburns	38.46	77.84	84.62	90.69	76.92	84.14	76.92	89.72	**91.35**	**93.75**
Gender	46.67	80.14	85.56	90.87	74.44	83.93	70.00	87.73	**90.56**	**91.52**
Average*	60.06	83.63	84.25	90.83	78.51	85.79	82.52	90.95	**87.04**	**91.41**

Table 4. Worst-group and average-group accuracy for hair color prediction on CelebA. We show the results for the worst 5 attributes (sorted based on ERM) and the given attribute `gender`. Average is computed based on the performance across all attributes. See Appendix D for full results.

5.2. Natural environments

Table 3 and 4 summarize the results on using natural environments to reduce biases from attributes that are unknown during both training and validation. We observe that directly applying DRO over human-defined groups already surpasses IRM and RGM on the worst-case accuracy averaged across all attributes. In addition, for the given attribute (`breast_cancer` and `gender`), DRO achieves nearly 10% more improvements over other baselines. However, this robustness doesn't generalize equally towards other attributes. By using the environment-specific classifier to create groups with contrasting unstable correlations, our algorithm delivers marked performance improvement over DRO, 6.81% on ASK2ME and 2.79% on CelebA.

How do we reduce bias from unknown attributes? Figure 3 and 4 visualize the correlation between each attribute and the label on ASK2ME and CelebA. We observe that although the signs of the correlation can be the same across the training environments, their magnitude may vary. Our algorithm makes use of this difference to create partitions that have opposite correlations for 13 (out of 15) attributes on ASK2ME and 22 (out of 38) attributes on CelebA. These opposite correlations help the classifier to avoid using unstable features during training.

6. Conclusion

In this paper, we propose a simple algorithm to learn correlations that are stable across environments. Specifically, we propose to use a classifier that is trained on one environment to partition another environment. By interpolating the distributions of its correct predictions and wrong predictions, we can uncover an oracle distribution where the unstable correlation vanishes. Experimental results on synthetic environments and natural environments validate that our algorithm is able to generate paritions with opposite unstable correlations and reduce bias that are unknown during training.

Acknowledgement

We thank Tofu for her love and support during this difficult time of pandemic. We thank all reviewers and the MIT NLP group for their thoughtful feedback. Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

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

Bao, Y., Chang, S., Yu, M., and Barzilay, R. Deriving machine attention from human rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1903–1913, 2018.

Bao, Y., Deng, Z., Wang, Y., Kim, H., Armengol, V. D., Acevedo, F., Ouardaoui, N., Wang, C., Parmigiani, G., Barzilay, R., Braun, D., and Hughes, K. S. Using machine learning and natural language processing to review and classify the medical literature on cancer susceptibility genes. *JCO Clinical Cancer Informatics*, (3):1–9, 2019. doi: 10.1200/CCI.19.00042. URL https://doi.org/10.1200/CCI.19.00042. PMID: 31545655.

Belinkov, Y., Poliak, A., Shieber, S. M., Van Durme, B., and Rush, A. M. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 877–891, 2019.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91, 2018.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. S. Invariant rationalization. *arXiv preprint arXiv:2003.09772*, 2020.

Choe, Y. J., Ham, J., and Park, K. An empirical study of invariant risk minimization. *arXiv preprint arXiv:2004.05007*, 2020.

Clark, C., Yatskar, M., and Zettlemoyer, L. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4060–4073, 2019.

Deng, Z., Yin, K., Bao, Y., Armengol, V. D., Wang, C., Tiwari, A., Barzilay, R., Parmigiani, G., Braun, D., and Hughes, K. S. Validation of a semiautomated natural language processing–based procedure for meta-analysis of cancer susceptibility gene penetrance. *JCO clinical cancer informatics*, 3:1–9, 2019.

Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

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

He, H., Zha, S., and Wang, H. Unlearn dataset bias in natural language inference by fitting the residual. *EMNLP-IJCNLP 2019*, pp. 132, 2019.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

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.

Jin, W., Barzilay, R., and Jaakkola, T. Enforcing predictive invariance across structured biomedical domains, 2020.

Kim, Y. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October

2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

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

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex), 2020.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, 2016.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015a.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015b.

Mahabadi, R. K., Belinkov, Y., and Henderson, J. End-to-end bias mitigation by modelling biases in corpora. ACL, 2020.

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J. K., Ceulemans, H., Clevert, D.-A., and Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451, 2018.

McAuley, J., Leskovec, J., and Jurafsky, D. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pp. 1020–1025. IEEE, 2012.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://www.aclweb.org/anthology/P19-1334.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Oren, Y., Sagawa, S., Hashimoto, T., and Liang, P. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4218–4228, 2019.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference using invariant prediction: identification and confidence intervals. *arXiv preprint arXiv:1501.01332*, 2015.

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.

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*, 2019.

Sagawa*, S., Koh*, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning (ICML)*, 2020.

Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020.

Sanh, V., Wolf, T., Belinkov, Y., and Rush, A. M. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Hf3qXoiNkR.

Schuster, T., Shah, D., Yeo, Y. J. S., Roberto Filizzola Ortiz, D., Santus, E., and Barzilay, R. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3410–3416, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1341. URL https://www.aclweb.org/anthology/D19-1341.

Stacey, J., Minervini, P., Dubossarsky, H., Riedel, S., and Rocktäschel, T. Avoiding the Hypothesis-Only Bias in Natural Language Inference via Ensemble Adversarial Training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8281–8291, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.665. URL https://www.aclweb.org/anthology/2020.emnlp-main.665.

Utama, P. A., Moosavi, N. S., and Gurevych, I. Towards debiasing nlu models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7597–7610, 2020.

Woodward, J. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://www.aclweb.org/anthology/P19-1472.