Mitigating Simplicity Bias in Deep Learning for Improved OOD Generalization and Robustness

Bhavya Vasudeva Kameron Shahabi Vatsal Sharan
{bvasudev, kyshahab, vsharan}@usc.edu
University of Southern California

Abstract

Neural networks (NNs) are known to exhibit *simplicity bias* (SB) where they tend to prefer learning 'simple' features over more 'complex' ones, even when the latter may be more informative. SB can lead to the model making biased predictions which have poor out-of-distribution (OOD) generalization. To address this, we propose a framework that encourages the model to use a more diverse set of features to make predictions. We first train a simple model, and then regularize the conditional mutual information with respect to it to obtain the final model. We demonstrate the effectiveness of this framework in various problem settings and real-world applications, showing that it effectively addresses SB and leads to more features being used, enhances OOD generalization, and improves subgroup robustness and fairness. We complement these results with theoretical analyses of the effect of the regularization and its OOD generalization properties.

1 Introduction

Motivated by considerations of understanding generalization in deep learning, there has been a series of interesting studies [ZBH+17, FC19, NKB+20] on understanding function classes favored by current techniques for training large neural networks. An emerging hypothesis is that deep learning techniques prefer to learn simple functions over the data. While this inductive bias has benefits in terms of preventing overfitting and improving (in-distribution) generalization in several cases, it is not effective in all scenarios[1]. Specifically, it has been found that in the presence of multiple predictive features of varying complexity, neural networks tend to be overly reliant on simpler features while ignoring more complex features that may be equally or more informative of the target [STR+20, NKK+19, MBJN23]. This phenomenon has been termed *simplicity bias* (SB) and has several undesirable implications for robustness and out-of-distribution (OOD) generalization (or generalization under distribution shift).

As an illustrative example, consider the Waterbirds dataset [SKHL20]. The objective here is to predict a bird's type (landbird vs. waterbird) based on its image (see Fig. 1 for an example). Note that features such as the background (land vs water background) are easier to learn, and can have significant correlation with the bird's type (since most images of landbirds are on a land background, and vice versa). Although features such as the shape of the bird are more predictive of its type, these tend to be more complex. SB can cause the model to be highly dependent on simpler features which are also predictive, such as the background in this case. A model which puts high emphasis on the background for solving this task is not desirable, since its performance may not transfer across different environments. A similar story arises in many different tasks—Table 1 summarizes various datasets, where the target or task-relevant features (also known as invariant features in the literature), are more complex than surrogate features that are superficially correlated with the label (also known as spurious features). In such cases, SB causes NNs to heavily rely on these surrogate features for predictions.

While simplicity bias aligns with the Occam's razor principle—the simplest model that explains the data should be the best model— it only seems effective for in-dsitribution generalization. Generalization under distribution shift is significantly different and calls for other assumptions, in the form of additional knowledge about the data and/or an inductive bias about the model, that are more suitable for this setting. Several methods [ABGLP20, CJZ21, BCB21, BCL22] have been proposed to address the problems of OOD generalization and robustness to subpopulation shifts. However, most of them require some knowledge about the spurious features. OOD generalization methods rely on the invariance principle and require knowledge of different environments of interests or assumptions on the underlying causal graph to

[1]This follows from the No Free Lunch theorem [Wol96, SSBD14] – discussed in more detail in Section 7.

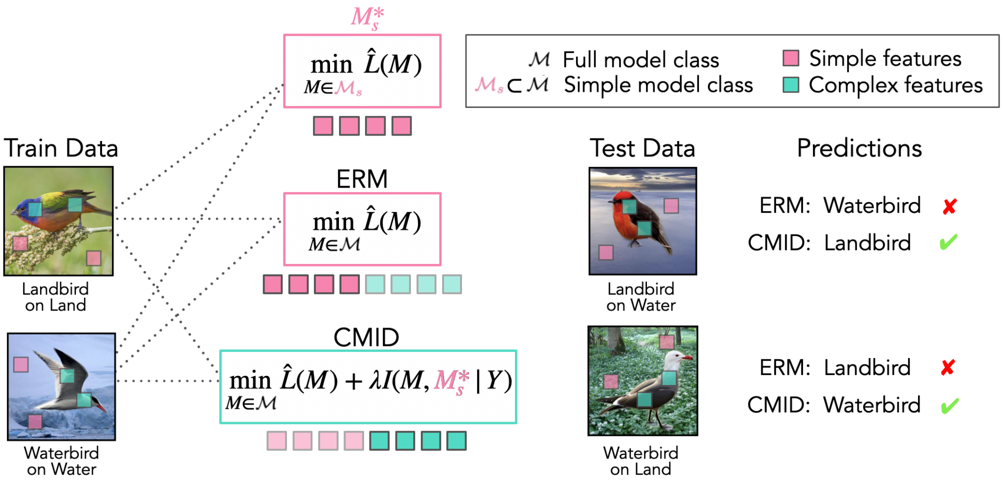

Figure 1: Summary of our approach. Models trained with ERM tend to focus on simple features (such as background) that do not generalize, whereas encouraging conditional independence with respect to a simple model increases reliance on complex features (such as shape) that generalize.

Dataset	Task-relevant/invariant feature	Surrogate/spurious feature
Waterbirds [SKHL20]	Bird type (waterbird/landbird)	Background (water/land)
CelebA [LLWT15]	Hair colour (blonde/other)	Gender (female/male)
MultiNLI [WNB18]	Reasoning (entailment/neutral/contradiction)	Negation words ('not' etc.)
CivilComments-WILDS [BDS+19]	Sentiment (toxic/non-toxic)	Demographic attributes (race, gender, religion)
Colored-MNIST [ABGLP20]	Digit (< 5 or ≥ 5)	Color (red/green)
Camelyon17-WILDS [BGM+19]	Diagnoses (tumor/no tumor)	Hospital
Adult-Confounded [CJZ21]	Income ($< \$50k$ or $\geq \$50k$)	Demographic attributes (race, gender)

Table 1: Summary of the datasets we consider. Spurious features seem *simpler* than invariant features.

decide whether a feature is spurious or invariant. Similarly, subgroup robustness methods often require group labels. To sidestep these requirements of additional knowledge about the data, *in this work we take the viewpoint that features that are usually regarded as being spurious for the task are often simple and quite predictive* (as suggested by Table 1). If this hypothesis is true, alleviating simplicity bias can be a useful inductive bias for improved robustness and OOD generalization.

With improved robustness and OOD generalization as major end goals in mind, we develop a new framework to mitigate simplicity bias and encourage the model to utilize a more diverse set of features for its predictions. Our high level goal is to ensure that the predictions of the trained model M have minimal conditional mutual information $I(M; S|Y)$[2] with any simple, predictive feature S, conditioned on the label Y. To achieve this, we first train a simple model M_s^* on the task, with the idea that this model captures the information in simple, predictive features. Subsequently, to train the final model M we add its conditional mutual information $I(M; M_s^*|Y)$ with the model M_s^* conditioned on the label Y as a regularizer to the usual empirical risk minimization (ERM) objective. With this regularization term, we incentivize the model to leverage additional, task-relevant features which may be more complex. We refer to our approach as *Conditional Mutual Information Debiasing or CMID* (see Fig. 1).

We demonstrate that our framework effectively mitigates simplicity bias, and achieves improved OOD generalization, sub-group robustness and fairness across a number of datasets. While several approaches have been proposed for each of these tasks, prior works tend to focus on one or two of these areas, whereas our approach is effective across all cases. We show that it leads to models which use more diverse features on certain tasks which have been previously proposed to measure simplicity bias. The method achieves improved sub-group robustness and OOD generalization on several benchmark tasks, including in the presence of multiple spurious features. It also leads to improved predictions from the perspective of fairness since the predictions are less dependent on protected attributes such as race or gender. In addition, *we prove theoretical results which provide insight and certain guarantees for our approach.* We analyze a Gaussian mixture model setup to understand the effect of our regularization and its ability to reduce dependence on spurious features. We also connect our approach to the causal learning literature by derive guarantees on OOD generalization in a causal learning framework.

[2]With slight abuse of notation, M denotes both the model and the random variable associated with its predictions.

We summarize our contributions as follows:

- We propose a novel framework to mitigate simplicity bias and encourage the model to utilize a more diverse set of features for its predictions. We show that features that are usually regarded as being spurious for a given task are often simpler than task-relevant features and alleviating simplicity bias can be a useful inductive bias for improved robustness and OOD generalization.
- Empirically, we demonstrate that our framework effectively mitigates simplicity bias, and achieves improved OOD generalization, sub-group robustness and fairness across 10 benchmark datasets (Section 5). These include different modalities such as image, text, and tabular data as well as application domains such as healthcare and fairness.
- Theoretically, we analyze the effect of our regularization and show its ability to reduce dependence on spurious features in a Gaussian mixture model setup. We also derive an OOD generalization guarantee in a causal learning framework (Section 4).
- Our approach is simple and easy to implement, and does not require additional knowledge in the form of group labels or unlabeled samples from another domain. While several approaches have been proposed for each of the goals we consider: mitigating simplicity bias, improving OOD generalization and subgroup robustness, prior work focuses on one or two of these applications, while our approach proves effective across all cases. In addition, our approach is theoretically grounded and explicit in its assumptions, which could make it easier for a practitioner to evaluate and use (Section 7).

1.1 Related Work

Simplicity Bias in NNs. Several works [GRM+22, AJB+17, VPCL19, GJM+20, STR+20, NKK+19, MBJN23, PKB+21] show that NNs trained with gradient-based methods prefer learning solutions which are 'simple'. [GRM+22] show that CNNs make predictions which depend more on image texture than image shape. [STR+20] create synthetic datasets (e.g. Slab data which we also consider) and show that in the presence of simple and complex features, NNs rely heavily on simple features even when both have equal predictive power. [NKK+19] show that the predictions of NNs trained by SGD can be approximated well by linear models during early stages of training. [MBJN23] show that 1-hidden layer NNs are biased towards using low-dimensional projections of the data for predictions.

OOD Generalization. Towards developing models which perform better in the real world, OOD generalization requires generalization to data from new *environments*. Environments are usually defined based on the correlation between some spurious feature and the label. Various methods aim to recover a predictor that is *invariant* across a set of environments. [ABGLP20] develop the invariant risk minimization (IRM) framework where environments are known, while [CJZ21] propose environment inference for invariant learning (EIIL), to recover the invariant predictor, when the environments are not known. Predict then interpolate (PI) [BCB21] and BLOOD [BCL22] use environment-specific ERMs to infer groups based on the correctness of predictions. [GSK+22] use targeted augmentations to only randomize domain-dependent features and retain domain-invariant features, to improve OOD generalization using data from fewer domains.

Subgroup Robustness. In many applications, models should not only do well on average but also do well on subgroups within the data. Several methods [SKHL20, NCA+20, KIW22, SSB+22, QPIW23] have been developed to improve the worst-group performance of a model. One widely used approach is to optimize the worst case risk over a set of subgroups in the data [DHN19, SKHL20, SDE+23]. CVaRDRO [DHN19] optimizes over all subgroups in the data, which is somewhat pessimistic, whereas GDRO [SKHL20] does this over a set of predefined groups. However, group knowledge may not always be available, various methods try to identify or infer groups and reweight minority groups in some way, when group labels are not available [NCA+20, LHC+21a, SDA+20] or partially available [SSB+22]. Just train twice (JTT) [LHC+21a] uses ERM to identify the groups based on correctness of predictions. Learning from failure (LfF) [NCA+20] simultaneously trains two NNs, encouraging one model to make biased predictions, and reweighting the samples it finds harder-to-learn (larger loss) to train the other model.

Fairness. Several works aim to ensure *fair* predictions of models across subgroups in the data, which are defined based on sensitive demographic information. Commonly used notions of group fairness include demographic parity [DHP+12], which aims to ensure that the representation of different demographic groups in the outcomes of a model is similar to their overall representation, and equalized odds [HPS16], which aims for equal predictive performance across groups using metrics such as true positive rate. Fairness interventions used when group information is available include: data reweighting to balance groups before training [KC12], learning separate models for different subgroups [DIKL18], and post-processing of trained models, such as adjusting prediction thresholds based on fairness-based metrics [HPS16].

Various approaches have been proposed to achieve fairness when demographic information may not be available. Multicalibration [HJKRR18] aims to learn a model whose predictions are calibrated for all subpopulations that can be identified in a computationally efficient way. [HSNL18] proposes a distributionally robust optimization (DRO)-based approach to minimize the worst-case risk over distributions close to the empirical distribution to ensure fairness. [LBC+20] proposes adversarial reweighted learning (ARL), where an auxiliary model identifies subgroups with inferior performance and the model of interest is retrained by reweighting these subgroups to reduce bias.

Feature Diversification and De-biasing Methods. Deep neural networks are known to exhibit unwanted biases. For instance, CNNs trained on image data may exhibit texture bias [GRM+22], and language models trained on certain datasets may exhibit annotation bias [GSL+18]. Several methods have been proposed to mitigate these biases. [BCY+20] introduce a framework to learn de-biased representations by encouraging them to differ from a reference set of biased representations. [LHX22] propose to trains two models alternatively, using one to identify biases using an equal opportunity violation criterion and training the other with a reweighted cross-entropy loss to make unbiased predictions. [DRL+23] reduce shortcut reliance by upweighting samples based on the misclassification probability of a simple model to train a complex model. [UMG20] propose a confidence regularization approach to encourage models to learn from all samples. Recent works also show that deep neural networks tend to amplify the societal biases present in training data [WZY+19, JMZC20] and they propose domain-specific strategies to mitigate such amplification.

Several works aim to improve feature learning for better generalization. [ZLPB22] aims to learn a shared representation using a succession of training episode, where they train new classifiers in each episode to do better on misclassified samples from previous episodes. [TALvdH22] also involves learning a shared representation and training multiple classifiers, where the alignment between their gradients is regularized to encourage feature diversity. [PJFK23, LYF22] do so by encouraging disagreement between two models leveraging unlabeled samples from the target domain.

2 Spurious Features are Simple and Predictive

In this section, we conduct an experiment to show that surrogate features are generally 'simpler' than invariant features (examples in Table 1). First, we define *simple models/features* for a task as follows:

Definition 1 (Simple models and features). *Consider a task on which benchmark models which achieve high in-distribution accuracy have a certain complexity (in terms of number of parameters, layers etc.). We consider models that have significantly lower complexity than benchmark models as* simple models*. Similarly, we consider features that can be effectively learned using simple models as* simple features*.*

Definition 1 proposes a metric of simplicity which is task dependent since it depends on the complexity of the models which get high accuracy on the task. We choose to not define quantitative measures in Definition 1 since those would be problem dependent. As an example, for colored-MNIST (CMNIST), neural networks with non-linearities are necessary to achieve high accuracy, and for Waterbirds, deeper networks such as ResNet-50 [HZRS16] achieve best results. Therefore, a linear model and a shallow CNN can be considered as simple models for these two datasets, respectively. We discuss simple model selection in more detail in Section 3.

Importantly, we observe that these simple models can still achieve good performance when the task is to distinguish between surrogate features even though they are not as accurate in predicting the invariant feature. Specifically, for CMNIST, we compare the performance of a linear model on color classification and digit classification on clean MNIST data. Similarly, for Waterbirds, we consider a shallow CNN (specifically, the `2DConvNet1` architecture in Appendix Fig. 9) and compare its performance on background classification (using images from the Places dataset [ZLK+17]) and bird type classification

Dataset	Simple Model				Benchmark Model	
	Predict surrogate feature		Predict invariant feature		Predict invariant feature	
	Train	Test	Train	Test	Train	Test
CMNIST	100	100	86.2 ± 0.2	86.6 ± 0.3	97.3 ± 0.1	96.7 ± 0.1
Waterbirds	79.6 ± 0.6	78.4 ± 0.6	60.5 ± 2.5	60.4 ± 2.4	98.8 ± 1.2	96.2 ± 1.1

Table 2: Comparison between performance for predicting the simple feature and the complex feature.

(using segmented images of birds from the CUB dataset [WBW+11]). The results in Table 2 indicate that surrogate features for these datasets are simple features, since they can be predicted much more accurately by simpler models than invariant features. Unsurprisingly, the respective benchmark models can capture the task-relevant features much more accurately as they are more complex than the simple models considered in each case.

Motivated by these observations, we define *spurious features* as follows. Operationally, our definition has the advantage that it does not require knowledge of some underlying causal graph or data from multiple environments to determine if a feature is spurious.

Definition 2 (Spurious features)**.** Spurious features *are simple features that are still reasonably correlated with the target label.*

Dataset	Train	Test
CMNIST	84.9 ± 0.2	10.7 ± 0.3
Waterbirds	93.3 ± 0.5	54.9 ± 1.1

Table 3: Train and test performance of the simple model on the target task.

In the presence of such features, simplicity bias leads the model to preferring such features over invariant features that are more complex. We also verify that when simple models are trained on the target tasks on these datasets, they tend to rely on these spurious features to make accurate predictions. Specifically, we consider the digit classification task using CMNIST data, where the correlation between the color and the label in the test set is 10%, and birdtype classification using Waterbirds data, where the test set consists of balanced groups (50% correlation). Table 3 shows that the test accuracy is close to the correlation between the spurious feature-based group label and the target label. This indicates that simple models trained on the target task utilize the spurious features to make predictions.

We briefly note here that spurious features as defined by us may not always be irrelevant for the task, and mitigating simplicity bias may not always be desirable (further discussion in Section 7). However, in most cases where the goal is to generalize under distribution shift, this assumption seems reasonable and mitigating simplicity bias proves effective, as we show later through experimental results in Section 5.

3 CMID: Learning in the Presence of Spurious Features

In this section, we outline our proposed approach to mitigate simplicity bias. Our approach leverages the fact that simple models can capture surrogate features much better than invariant features (Table 2), and rely on the spurious features to make predictions, even when trained on the target task (Table 3). Thus, by ensuring that the final model has low conditional mutual information with respect to such a simple model, we encourage it to utilize a more diverse set of features.

Notation. Let $Z = (X, Y)$ denote an input-label pair, where $X \in \mathbb{R}^d, Y \in \{0, 1\}$, sampled from some distribution $\mathcal{D}$, D denote a dataset with n samples, $\mathbb{E}_D$ denote the empirical mean over D, $\mathcal{X}$ denote the input space, $M(\theta) : \mathcal{X} \to [0, 1]$ denote a model, parameterized by θ (shorthand M). The predictions of the model are given by $\mathbb{1}[M(X) > 0.5]$. Subscripts $(\cdot)_s$ and $(\cdot)_c$ denote simple and complex, respectively. Let $\mathcal{M}$ denote the class of all models M, $\ell_M(Z) : \mathcal{X} \times \{0, 1\} \to \mathbb{R}$ denote a loss function, and $\sigma(x, T) = \dfrac{1}{1 + e^{-Tx}}$ denote the sigmoid function, with temperature parameter T.

Let $H(\cdot)$ denote the Shannon entropy of a random variable. $I(\cdot;\cdot)$ measures Shannon mutual information between two random variables. With slight abuse of notation, let M and Y also denote the (binary) random variables associated with the predictions of model M on datapoints X_i's and the labels Y_i's, across all $i \in [n]$, respectively. The conditional mutual information (CMI) between the outputs of two models given the label is denoted by $I(M_1; M_2|Y)$.

The empirical risk minimizer (ERM) for class $\mathcal{M}$ is denoted as M^*, and is given by:

$$M^* := \mathrm{ERM}(\mathcal{M}) = \operatorname*{arg\,min}_{M \in \mathcal{M}} \frac{1}{n} \sum_{i \in [n]} \ell_M(Z_i) = \operatorname*{arg\,min}_{M \in \mathcal{M}} \mathbb{E}_D \, \ell_M(Z).$$

We consider the class of simple models $\mathcal{M}_s \subset \mathcal{M}$ and complex models $\mathcal{M}_c = \mathcal{M} \setminus \mathcal{M}_s$. We now describe our proposed two-stage approach CMI Debiasing (CMID):

- First, learn a simple model M_s, which minimizes risk on the training data:

$$M_s^* = \mathrm{ERM}(\mathcal{M}_s).$$

- Next, learn a complex model M_c by regularizing its CMI with M_s.

$$M_c = \operatorname*{arg\,min}_{M \in \mathcal{M}} \mathbb{E}_D \, \ell_M(Z) + \lambda \hat{I}_n(M; M_s^* | Y),$$

  where $\hat{I}_n(M; M_s^*|Y)$ denotes the estimated CMI over D, and λ is the regularization parameter.

A few remarks are in order about the choice of our regularizer and the methodology to select the simple model class.

Note that we penalize the conditional mutual information instead of mutual information. This is because both M_s^* and M are expected to have information about Y (for e.g. in the Waterbirds dataset both bird type and background type are correlated with the label). Hence they will not be independent of each other, but they are closer to being independent when conditioned on the label. We also note that we use conditional mutual information to measure dependence, instead of other measures such as enforcing orthogonality of the predictions. This is for the simple reason that mutual information measures all—potentially non-linear—dependencies between the random variables.

We note that while several models can be considered simple based on Definition 1, the choice of the simple model class $\mathcal{M}_s$ used for our approach is task dependent. Intuitively, we want to use the simplest model that we expect to do reasonably well on the given task. Models that are too simple may not be able to capture surrogate features effectively, whereas models that are very complex may rely on task-relevant features, even though such reliance may be weak due to simplicity bias. Although the choice of $\mathcal{M}_s$ may impact the performance of CMID, we use a fairly simple selection rule throughout our experiments, which works well. In general, for datasets where the final model is a shallow NN, we used a linear model as the simple model. For image datasets where the final model is a ResNet or DenseNet-based model, we consider shallow 2D CNNs as simple models. For language datasets where BERT-based models are the final models, we consider shallow MLPs or 1D CNNs as simple models. Details about the simple model architectures as well as results showing the effect of using different simple model architectures for our approach can be found in the Appendix.

To estimate CMI, first consider the conditional (joint) distributions over n samples:

$$p(M\!=\!m|Y\!=\!y) = \frac{\Sigma_{i\in[n]} \mathbb{1}[Y_i\!=\!y] \, \zeta(M(X_i), m)}{\Sigma_{i\in[n]} \mathbb{1}[Y_i\!=\!y]},$$

$$p(M\!=\!m, M_s\!=\!m'|Y\!=\!y) = \frac{\Sigma_{i\in[n]} \mathbb{1}[Y_i\!=\!y] \, \zeta(M(X_i), m) \, \zeta(M_s(X_i), m')}{\Sigma_{i\in[n]} \mathbb{1}[Y_i\!=\!y]},$$

where $m, m', y \in \{0, 1\}$, $\zeta(M(X_i), 1)\!=\!\mathbb{1}[M(X_i)\!>\!0.5]$, and $\zeta(M(X_i), 0)\!=\!1\!-\!\mathbb{1}[M(X_i)\!>\!0.5]$.

Note that the CMI computed using these would not differentiable as these densities are computed by thresholding the outputs of the model. Since we want to add a CMI penalty as a regularizer to the ERM objective and optimize the proposed objective using standard gradient-based methods, we need a differentiable version of CMI. Thus, for practical purposes, we use an approximation of the indicator function $\mathbb{1}[x > 0.5]$, given by $\sigma(x - 0.5, T)$, where T determines the degree of smoothness or sharpness in the approximation.

This can be easily generalized for multi-class classification with C classes. In that case, $m, m', y \in \{0, \cdots, C\!-\!1\}$, $M(X_i)$ is a C-dimensional vector with the m^{th} entry indicating the probability of predicting class m, and $\zeta(M(X_i), m) = \mathbb{1}[\operatorname*{arg\,max}_{j\in[C]} M_j(X_i) = m]$. To make this differentiable, we use the softmax function with temperature parameter T to approximate the indicator function.

Using these densities, the estimated CMI is:

$$\hat{I}_n(M, M_s|Y) = \sum_y p(Y=y_i) \sum_{m,m'} p(M=m, M_s=m'|Y=y) \log\left[\frac{p(M=m, M_s=m'|Y=y)}{p(M=m|Y=y)p(M_s=m'|Y=y)}\right]. \tag{1}$$

This estimate is differentiable, making it compatible with gradient-based methods. Therefore, we utilize it as a regularizer for the proposed approach. We include comparison between the estimated CMI and the original CMI (computed with discretized model outputs) in the Appendix.

4 Theoretical Results

In this section, we analyze the effect of CMI regularization and show it leads to reduced dependence on spurious features in a Gaussian mixture model. We also obtain a simple OOD generalization guarantee for our approach in a causal learning framework.

4.1 Effect of CMI Regularization for Gaussian Features

We consider data generated from the following Gaussian mixture model (refer to the left-most panel in Fig. 2 for an example of data generated from this distribution).

Assumption 1. *Let the label $y \sim \mathcal{R}(0.5)$, where $\mathcal{R}(p)$ is a $\{\pm 1\}$ random variable which is 1 with probability p. Consider an* invariant *feature X_1 and a* spurious *feature X_2, with distributions:*

$$X_1 \sim \mathcal{N}(y\mu_1, \sigma_1^2) \text{ and } X_2 \sim \mathcal{N}(a\mu_2, \sigma_2^2),$$

where $a \sim y\mathcal{R}(\eta)$ is a spurious attribute, with an unstable correlation with y, and $\mu_1, \mu_2 > 0, \eta > 0.5$. Assume $X_1 \perp X_2 | y$ and let $\mu_2' = (2\eta - 1)\mu_2$.

We note that [SRKL20, RRR21] consider a similar Gaussian mixture based data model for core and spurious features. Intuitively, the mean of a feature determines the margin of the predictor that relies on that feature, while the variance of the feature determines the best possible accuracy for that predictor.

We consider linear models to predict y from the features X_1 and X_2. Let $\mathcal{M} = \{(w_1, w_2) : w_1, w_2 \in \mathbb{R}\}$ be all possible linear models and $\mathcal{M}_s = \{(0, w) : w \in \mathbb{R}\} \cup \{(w, 0) : w \in \mathbb{R}\}$ be a simpler model class which only uses one of the two features. We consider the mean squared error (MSE) loss, and the ERM solution is given by:

$$\text{ERM}(\mathcal{M}) = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{M}} \mathbb{E}\,(w_1 X_1 + w_2 X_2 - y)^2.$$

Proposition 1. *ERM($\mathcal{M}$) satisfies $\frac{w_1}{w_2} = \frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1^2}$. When $\frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1^2} < 1$, ERM($\mathcal{M}_s$)$= \left[0, \frac{\mu_2'}{\sigma_2^2}\right]$ (upto scaling).*

We now consider the effect of CMI regularization. Since both the features are of similar complexity here, we consider the core feature to have a lower signal-to-noise ratio, i.e. $\frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1^2} < 1$, to model the observation that the spurious feature is learned more easily with ERM. Then, as per Proposition 1 in the first step we learn a simple model $w_2^* X_2$ which uses only X_2. We now consider the ERM problem but with a constraint on the CMI:

$$\text{ERM}_{\mathcal{C}}(\mathcal{M}) = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{M}} \mathbb{E}\,(w_1 X_1 + w_2 X_2 - y)^2 \text{ s.t. } I(w_1 X_1 + w_2 X_2; w_2^* X_2 | y) \le \nu. \tag{2}$$

We show the following guarantee on the learned model.

Theorem 1. *Let data be generated as per Assumption 1. For $\nu = 0.5 \log(1 + c^2)$ for some c:*

1. *When $\frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1 \sigma_2} > \frac{1}{c}$, the solution to (2) is the same as ERM($\mathcal{M}$), so $\frac{w_1}{w_2} = \frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1^2}$.*
2. *Otherwise, w_1 is upweighted and the solution to (2) satisfies $\frac{|w_1|}{|w_2|} = \frac{1}{c} \frac{\sigma_2}{\sigma_1}$.*

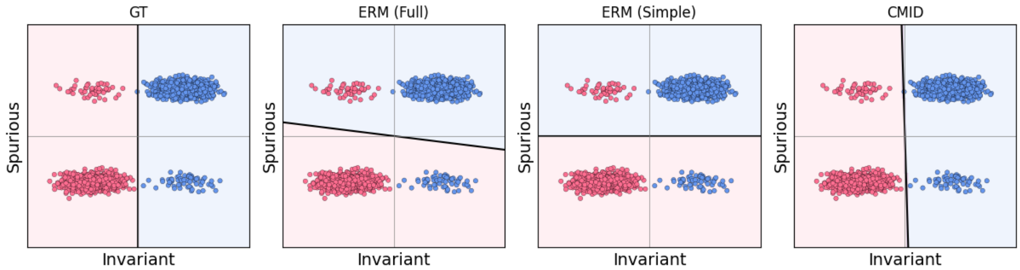

Figure 2: Results on synthetic Gaussian data generated as per Assumption 1, with 2000 samples, $\mu_1 = \mu_2 = 5$, $\sigma_1 = 1.5, \sigma_2 = 0.5$. Left to right: Ground truth (GT) predictor, ERM with $\mathcal{M}$ as the class of linear models, ERM with $\mathcal{M}$ as the class of threshold functions, ERM with CMI constraint, with $c = 0.01$.

Theorem 1 suggests that for an appropriately small c, regularizing CMI with the simple model leads to a predictor which mainly uses the invariant feature. This is supported by experimental results on data drawn according to Assumption 1, shown in Fig. 2.

In Fig. 3, we visualize the relationship between $\frac{w_1}{w_2}$ and $\frac{\sigma_2^2}{\sigma_1^2}$ predicted in Theorem 1 (assuming $\mu_1 = \mu_2$ and $\eta = 0.95$.) We see that a lower value of c promotes conditional independence with X_2 and upweighs w_1 more strongly. When $c \to 0$, $w_2 \to 0$.

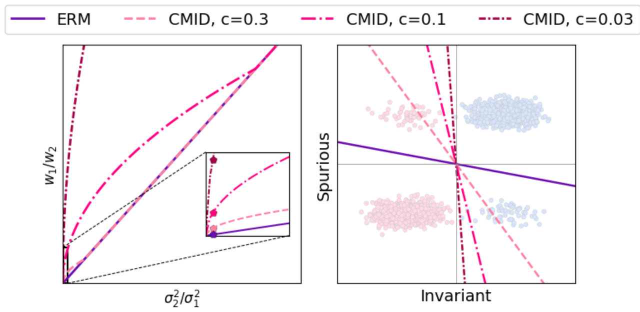

Figure 3: Effect of CMI regularization (wrt X_2) for different values of c (corresponding to regularization strength). Left: Lower values of c indicate stronger CMI regularization, resulting in more upweighting of w_1 wrt w_2. Inset shows a zoomed-in region and markers compare the three solutions when $\sigma_2^2/\sigma_1^2 = 1/6$. Right: Decision boundaries for predictors corresponding to the markers.

Next, we consider the case where there are multiple spurious features, by adding another feature

$$X_3 \sim \mathcal{N}(a'\mu_3, \sigma_3^2), \text{ where } a' \sim y\mathcal{R}(\eta') \tag{3}$$

to the setup in Assumption 1. We show that regularizing CMI with respect to the optimal predictor that uses X_2 and X_3 also results in w_1 being upweighted. We consider the constrained problem:

$$\begin{aligned} \mathrm{ERM}_{\mathcal{C}}(\mathcal{M}) = \underset{\boldsymbol{w} \in \mathcal{M}}{\arg\min}\ & \mathbb{E}\,(w_1X_1 + w_2X_2 + w_3X_3 - y)^2 \\ \text{s.t. }\ & I(w_1X_1 + w_2X_2 + w_3X_3; w_2^*X_2 + w_3^*X_3 | y) \leq \nu, \end{aligned} \tag{4}$$

where $(w_2^*, w_3^*) \propto \left(\mu_2'/\sigma_2'^2, \mu_3'/\sigma_3'^2\right)$ (the weights for the optimal linear model which uses X_2 and X_3). We show the following result.

Theorem 2. *Let the data be generated as per Assumption 1 and (3). Let $\nu = 0.5\log(1 + c^2)$ for some sufficiently small c, and $\frac{\sigma_2^2}{\sigma_2'^2} = \frac{\sigma_3^2}{\sigma_3'^2}$. Then the solution to (4) satisfies: $\frac{|w_2\mu_2' + w_3\mu_3'|}{|w_1|\sigma_1} = c'$, where $c' = 2c\sqrt{\frac{\mu_2'^2}{\sigma_2^2} + \frac{\mu_3'^2}{\sigma_3^2}}$. Moreover, $w_2, w_3 \propto c'w_1$.*

Similar to before, the result says that for a sufficiently small CMI constraint we learn a model which mainly uses the invariant feature.

4.2 OOD Generalization in a Causal Learning Framework

Following the setting in [ABGLP20, LHC+21b], we consider a dataset $D = \{D^e\}_{e \in \mathcal{E}_{tr}}$, which is composed of data $D^e \sim \mathcal{D}_e^{n_e}$ gathered from different training environments $e \in \mathcal{E}_{tr}$, where e denotes an environment label, n_e represents the number of samples in e. $\mathcal{E}_{tr}$ denotes the set of training environments.

The problem of finding a predictor with good OOD generalization performance, can be formalized as:

$$\arg\min_{M\in\mathcal{M}} \max_{e\in\mathcal{E}} \mathbb{E}_D[\ell_M(Z)|e],$$

i.e., optimizing over the worst-case risk on all environments in set $\mathcal{E}$. Usually, $\mathcal{E} \supset \mathcal{E}_{tr}$, and hence, the data and label distribution can differ significantly for $e \in \mathcal{E}_{tr}$ and $e \in \mathcal{E} \setminus \mathcal{E}_{tr}$. This makes the OOD generalization problem hard to solve.

The invariant learning literature assumes the existence of invariant and variant features. In this section, we assume that the model of interest, say $M(X)$ is composed of a featurizer Φ and a classifier ω on top of it, *i.e.* $M(X) = \omega \circ \Phi(X)$. For simplicity, we omit the argument X and assume that learning a featurization includes learning the corresponding classifier, so we can write $M = \Phi$. Let E be a random variable sampled from a distribution on $\mathcal{E}$.

Definition 3 (Invariant and Variant Predictors)**.** *A feature map* Φ *is called* invariant *and is denoted by* $\mathit{\Phi}$ *if* $Y \perp E|\mathit{\Phi}$*, whereas it is called* variant *and is denoted by* $\mathit{\Psi}$ *if* $Y \not\perp E|\mathit{\Psi}$*.*

Several works [ABGLP20, LHC+21b, CJZ21] attempt to recover the invariant feature map by proposing different ways to find the maximally invariant predictor [CZYJ20], defined as:

Definition 4 (Invariance Set and Maximal Invariant Predictor)**.** *The invariance set* $\mathcal{I}$ *with respect to environment set* $\mathcal{E}$ *and hypothesis class* $\mathcal{M}$ *is defined as:*

$$\mathcal{I}_{\mathcal{E}}(\mathcal{M}) = \{\mathit{\Phi} : Y \perp E|\mathit{\Phi}\} = \{\mathit{\Phi} : H[Y|\mathit{\Phi}] = H[Y|\mathit{\Phi}, E]\}.$$

The corresponding maximal invariant predictor (MIP) of $\mathcal{I}_{\mathcal{E}}(\mathcal{M})$ *is* $\mathit{\Phi}^* = \arg\max_{\mathit{\Phi}\in\mathcal{I}_{\mathcal{E}}(\mathcal{M})} I(Y;\mathit{\Phi})$*.*

The MIP is an invariant predictor that captures the most information about Y. Invariant predictors guarantee OOD generalization, making MIP the optimal invariant predictor (Theorem 2.1 in [LHC+21b]).

As discussed in Section 1, most current work assumes that the environment labels e for the datapoints are known. However, environment labels typically are not provided in real-world scenarios. In this work, we don't assume access to environment labels and instead, we rely on another aspect of these features: are they simple or complex? We formalize this below:

Assumption 2 (Simple and Complex Predictors)**.** *The invariant feature comprises of complex features, i.e.* $\mathit{\Phi}^* = [\mathit{\Phi}_c]$*, where* $\mathit{\Phi}_c \in \mathcal{M}_c$*. The variant feature comprises of simple and complex features, i.e.,* $\mathit{\Psi}^* = [\mathit{\Psi}_c, \mathit{\Psi}_s]$*, where* $\mathit{\Psi}_c \in \mathcal{M}_c$*,* $\mathit{\Psi}_s \in \mathcal{M}_s$ *and* $I(Y;\mathit{\Psi}_s) > 0$*.*

We consider the underlying causal model in [RRR21] which makes the following assumption.

Assumption 3 (Underlying Causal Model)**.** *Given the model in Fig. 4,* $I(\mathit{\Phi},\mathit{\Psi}|Y) = 0$ *and* $I(\mathit{\Psi}_s,\mathit{\Psi}_c|Y) > I(\mathit{\Psi}_s,\mathit{\Psi}_c|Y,E)$*.*

The following simple result shows that our method finds the maximal invariant predictor, and thus generalizes OOD.

Proposition 2. *Let* $ERM(\mathcal{M}_s) = M_s^*$*. Under Assumptions 2 and 3, the solution to the problem:*

$$\arg\min_{M\in\mathcal{M}} \mathbb{E}\,\ell_M(Z) \;\; s.t. \;\; I(M; M_s^*|Y) = 0 \tag{5}$$

is $M = \mathit{\Phi}^*$*, the maximal invariant predictor.*

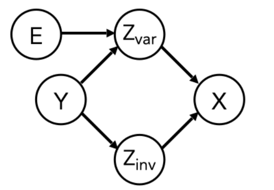

Figure 4: Our causal model. Latent variables Z_{inv} and Z_{var} correspond to invariant and variant features $\mathit{\Phi}^*$ and $\mathit{\Psi}^*$ respectively.

Here, we note that Assumption 2 is key to our result. Although this assumption may not always hold directly in practice, it characterizes the condition under which our approach can recover the invariant predictor even though it does not have an explicit causal/invariant learning motivation or access to environment labels.

5 Experiments

We show that CMID reduces simplicity bias, and yields improvements across a number of robustness, OOD generalization and fairness metrics. We first exhibit how CMID can mitigate simplicity bias in the Slab and ImageNet-9 datasets. We then provide results for CMID on various OOD generalization tasks. These include data with multiple spurious features, real-world medical data, and a fairness application. Finally, we test our approach on some benchmark classification tasks that evaluate subgroup robustness[3]. CMID's good performance on these tasks hints that the method impels models to learn invariant features in the presence of spurious ones. We note that past approaches usually specifically target one or two of these problem settings. Thus, in each section we choose the most task-relevant methods to compare with, as established in prior work on that task.

5.1 Mitigating Simplicity Bias

In this section, we show that CMID mitigates simplicity bias and encourages the model to learn more diverse features on two tasks. These include a synthetic Slab dataset, that was used to demonstrate extreme simplciity bias in NNs [STR+20], and ImageNet-9 dataset, where CNNs are known to rely more on texture rather than shapes to make predictions [GRM+22].

Slab Data. Slab data was proposed in [STR+20] to model simplicity bias. Each feature is composed of k data blocks or slabs. We consider two configurations of the slab data, namely 3-Slab and 5-Slab, as shown in Fig. 5. In both the cases, the first feature is linearly separable. The second feature has 3 slabs in the 3-Slab data and 5 slabs in the 5-Slab data. The first feature is simple since it is linearly separable, while the features with more slabs involve a piece-wise linear model and are thus complex. The linear model is perfectly predictive, but the predictor using both types of features attains a much better margin, and generalizes better under fixed ℓ_1-norm perturbations to the features. Fig. 5 shows the decision boundary using ERM and the proposed approach. We see that CMID encourages the model to use both features and attain better margin.

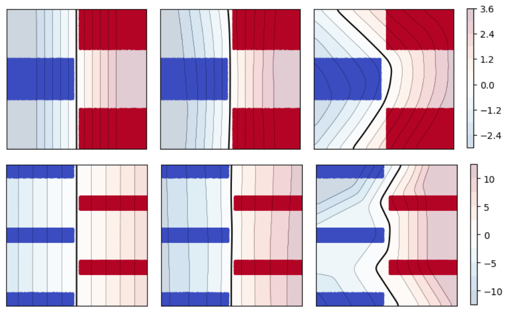

Figure 5: Results on the slab dataset. Left: Linear model, Center: 1-hidden-layer NN-ERM, Right: 1-hidden-layer NN-CMID.

Texture vs Shape Bias on ImageNet-9. [GRM+22] showed that CNNs trained on ImageNet tend to make predictions based on image texture rather than image shape. To quantify this phenomenon, the authors designed the GST dataset, which contains synthetic images with conflicting shape and texture (e.g., image of a cat modified with elephant skin texture as a conflicting cue). The *shape bias* of a model on the GST dataset is defined as the number of datapoints for which the model correctly identifies shape compared to the total number of datapoints for which the model correctly identifies either shape or texture.

We consider ImageNet-9, a subset of ImageNet organized into nine classes by [XEIM20]. Each of the nine condensed classes consists of images from multiple ImageNet classes. These include dog, bird, wheeled vehicle, reptile, carnivore, insect, musical instrument, primate, and fish. To interpret a prediction from a model trained on ImageNet-9 as a GST dataset prediction, we consider a subset of classes from both and use the mapping listed in Table 4. Specifically, to determine whether a model trained on ImageNet-9 predicts correctly on a GST image, we first determine which of the 5 ImageNet-9 classes from Table 4 has the highest probability based on the model's output, and then use the mapping to obtain the predicted GST class label. Thus, the

ImageNet-9 Class	GST Dataset Classes
dog	dog
bird	bird
wheeled vehicle	bicycle, car, truck
carnivore	bear, cat
musical instrument	keyboard

Table 4: ImageNet-9 classes mapped to corresponding GST dataset classes.

[3]Our code is available at https://github.com/estija/CMID.

shape bias is calculated as[4]:

$$\text{shape bias} = \frac{\text{number of correct shape predictions}}{\text{number of correct shape predictions} + \text{number of correct texture predictions}} \times 100.$$

A shape bias of 100 indicates that the model always uses shape when shape conflicts with texture, whereas 0 indicates that the model always uses texture instead of shape.

We train a ResNet50 model using ERM and CMID. In Table 5, we show that compared to ERM, training with CMID mitigates texture bias and encourages the model to rely on shape for predictions.

Method	Shape Bias
ERM	38.6
JTT [LHC+21a]	45.8
CMID	51.8

Table 5: Shape bias of ResNet-50 trained on ImageNet-9.

5.2 Better OOD Generalization

In this section, we show that CMID leads to improved OOD generalization on several datasets. We first evaluate our approach on the widely used CMNIST data [ABGLP20], where color is a spurious feature, as well as a recent variant with an additional spurious feature in the form of patch, CPMNIST [BCL22]. We also consider the Camelyon17, a real-world medical image dataset from the WILDS benchmark [KSM+21], and the Adult-Confounded dataset [CJZ21], a semi-sythetic variant of the UCI-Adult dataset, which is a well-known fairness benchmark.

Synthetic: CMNIST and Color+Patch MNIST. We present results on two variants of the MNIST dataset [Den12], which contains images of handwritten digits, using a binary digit classification task (< 5 or not). The colored-MNIST data was proposed in [ABGLP20], where color (red/green) is injected as a spurious feature, with unstable correlation $1 - p_e$ with the label across environments. The train data has two environments with $p_e = 0.1, 0.2$ while the test data has $p_e = 0.9$, to test OOD performance. Further, it contains 25% label noise to reduce the predictive power of the task-relevant feature: digit shape. We also consider the color+patch MNIST data proposed in [BCL22], where an additional spurious feature is injected into the data, in the form of a 3×3 patch. The position of the patch (top left/bottom right) is correlated with the label, with the same p_e, but independent of the color.

Method	Group labels	Bias: Color			Bias: Color & Patch		
		Test (i.i.d) $p_e = 0.1$	Test (OOD) $p_e = 0.9$	δ_{gap}	Test (i.i.d) $p_e = 0.1$	Test (OOD) $p_e = 0.9$	δ_{gap}
ERM	No	88.6 ± 0.3	16.4 ± 0.8	-72.2	93.7 ± 0.3	14.0 ± 0.5	-79.7
IRM [ABGLP20]	Yes	71.4 ± 0.9	66.9 ± 2.5	-4.5	93.5 ± 0.2	13.4 ± 0.3	-80.1
GDRO [SKHL20]	Yes	89.2 ± 0.9	13.6 ± 3.8	-75.6	92.3 ± 0.3	14.1 ± 0.8	-78.2
PI [BCB21]	Yes	70.3 ± 0.3	70.2 ± 0.9	-0.1	85.4 ± 0.9	15.3 ± 2.7	-70.1
BLOOD [BCL22]	Yes	70.5 ± 1.1	70.7 ± 1.4	0.2	68.3 ± 2.3	62.3 ± 3.3	-6.0
EIIL [CJZ21]	No	71.7 ± 1.6	62.8 ± 5.0	-8.9	65.3 ± 8.4	53.0 ± 5.6	-12.3
JTT [LHC+21a]	No	72.2 ± 1.1	64.6 ± 0.56	-7.6	64.0 ± 2.7	56.2 ± 2.7	-7.8
CMID	No	69.2 ± 0.9	68.9 ± 0.9	-0.3	60.3 ± 2.7	59.4 ± 1.0	-0.9
Optimal	-	75	75	0	75	75	0

Table 6: Comparison on Colored MNIST and Color+Patch MNIST.

Following [BCL22], we evaluate on i.i.d test data with $p_e = 0.1$ and OOD data with $p_e = 0.9$ for both the cases (details in the Appendix). Table 6 shows that CMID gets competitive OOD performance with methods that require group knowledge, and has the lowest gap δ_{gap} between test performance on i.i.d and OOD samples, even in the presence of multiple spurious features.

Medical: Camelyon17-WILDS. Camelyon17-WILDS is a real-world medical image dataset of data collected from five hospitals [BGM+19, KSM+21]. Three hospitals comprise the training set, one is the validation set and the third is the OOD test set. Images from different hospitals vary visually. The task is to predict whether or not the image contains tumor tissue, and the dataset is a well-known OOD generalization benchmark [BCL22, KSM+21]. Table 7 shows that CMID leads to higher average accuracies than existing group-based methods when evaluated on images from the test hospital.

[4]We use the code by [GRM+22] available at https://github.com/rgeirhos/texture-vs-shape.

Method	ERM	IRM [ABGLP20]	GDRO [SKHL20]	PI [BCB21]	JTT [LHC+21a]	BLOOD [BCL22]	CMID
Train Acc	97.3 ± 0.1	97.1 ± 0.1	96.5 ± 1.4	93.2 ± 0.2	85.2 ± 6.6	93.0 ± 1.8	92.9 ± 1.4
OOD Test Acc	66.5 ± 4.2	59.4 ± 3.7	70.2 ± 7.3	71.7 ± 7.5	74.0 ± 16.4	74.9 ± 5.0	76.1 ± 6.3

Table 7: Average train and OOD test accuracies over three trials on Camelyon17-WILDS.

Fairness: Adult-Confounded. The Adult-Confounded dataset is a semi-synthetic variant of the UCI Adult dataset [NHBM98, LD21], developed by [CJZ21]. The UCI Adult dataset contains attributes based on demographics and employment information and the target label a binarized income indicator (thresholded at $\$50,000$). The task is commonly used as an algorithmic fairness benchmark.

[LBC+20, CJZ21] define four sensitive subgroups based on binarized sex (Male/Female) and race (Black/non-Black) labels: Non-Black Males (G1), Non-Black Females (G2), Black Males (G3), and Black Females (G4). They observe that each subgroup has a different correlation strength with the target label ($p(y = 1|G)$), and thus, in some cases, subgroup membership alone can be used to achieve low error rate in prediction. Based on this observation, [CJZ21] create a semi-synthetic variant of the UCI Adult data, known as Adult-Confounded, where they exaggerate the spurious correlations in the original data. As G1 and G3 have higher values of $p(y = 1|\mathrm{G})$ across both the splits, compared to the other subgroups (see [CJZ21] for exact values), these values are increased to 0.94, while they are set to 0.06 for the remaining two subgroups, to generate the Adult-Confounded dataset. In the test set, these are reversed, so that it serves as a worst-case audit to ensure that the model is not relying on subgroup membership alone in its predictions.

Table 8 shows that compared to other methods CMID achieves superior OOD test performance with the least gap between train and test performance, indicating its low reliance on sensitive subgroup information.

Method	ERM	ARL [LBC+20]	JTT [LHC+21a]	EIIL [CJZ21]	CMID
Train Acc	92.7 ± 0.5	72.1 ± 3.6	80.2 ± 1.7	69.7 ± 1.6	76.2 ± 2.2
OOD Test Acc	31.1 ± 4.4	61.3 ± 1.7	71.8 ± 5.3	78.8 ± 1.4	78.8 ± 0.7
δ_{gap}	-61.6	-10.8	-8.4	9.1	2.6

Table 8: Comparison on Adult-Confounded dataset.

5.3 Subgroup Robustness

We evaluate our approach on four benchmark classification tasks for robustness to spurious correlations, namely on Waterbirds, CelebA, MultiNLI and CivilComments-WILDS datasets (Table 1). We follow the setup in [SKHL20] for the first three and [KSM+21] for CivilComments-WILDS.

Waterbirds. Waterbirds is a synthetic dataset created by [SKHL20] consisting of bird images over backgrounds. The task is to classify whether a bird is a *landbird* or a *waterbird*. The background of the image *land background* or *water background*, acts a spurious correlation.

CelebA. CelebA is a synthetic dataset created by [LLWT15] containing images of celebrity faces. We classify the hair color as *blonde* or *not blonde*, which is spuriously correlated with the gender of the celebrity *male* or *female*, as done in [SKHL20, LHC+21a].

MultiNLI. MultiNLI [WNB18] is a dataset of sentence pairs consisting of three classes: entailment, neutral, contradiction. Pairs are labeled based on whether the second sentence entails, is neutral with, or contradicts the first sentence, which is correlated with the presence of negation words in the second sentence [SKHL20, LHC+21a].

CivilComments-WILDS. CivilComments-WILDS is a dataset of online comments proposed by [BDS+19]. The goal is to classify whether a comment is *toxic* or *non-toxic*, which is spuriously correlated with the mention of one or more of the following demographic attributes: male, female, White, Black, LGBTQ, Muslim, Christian, and other religion [PSF18, DLS+18]. Similar to previous work [KSM+21, LHC+21a], we evaluate over 16 overlapping groups, one for each potential label-demographic pair.

Although CMID does not require group knowledge for training, following [LHC+21a], we use a validation set with group labels for model selection. Table 9 shows the average and worst-group accuracies for CMID and comparison with other methods [DHN19, NCA+20, LHC+21a] which do not use group information. GDRO [SKHL20], which uses group information, acts as a benchmark. We see that on three of these datasets, CMID competes with state-of-the-art algorithms that improve subgroup robustness. Interestingly, CMID seems paricularly effective on the two language-based datasets.

Method	Group labels	Waterbirds		CelebA		MultiNLI		CivilComments-WILDS	
		Average	Worst-group	Average	Worst-group	Average	Worst-group	Average	Worst-group
ERM	No	97.3	72.6	95.6	47.2	82.4	67.9	92.6	57.4
CVaRDRO [DHN19]	No	96.0	75.9	82.5	64.4	82.0	68.0	92.5	60.5
LfF [NCA+20]	No	91.2	78.0	85.1	77.2	80.8	70.2	92.5	58.8
JTT [LHC+21a]	No	93.3	86.7	88.0	81.1	78.6	72.6	91.1	69.3
CMID	No	88.6	84.3	84.5	75.3	81.4	71.5	84.2	74.8
GDRO [SKHL20]	Yes	93.5	91.4	92.9	88.9	81.4	77.7	88.9	69.9

Table 9: Average and worst-group test accuracies on benchmark datasets for subgroup robustness.

We also note that CMID is not very effective on CelebA images. We believe that this is because both the spurious feature (gender) and the invariant feature (hair color) for CelebA are of similar complexity. In Section 6.2, we explore this further with a similar experiment as in Section 2 for CelebA.

5.4 Fairness Application: Bias in Occupation Prediction

The Bios dataset [DARW+19, CDAMK23] is a large-scale dataset of more than 300k biographies scrapped from the internet. The goal is to predict a person's occupation based on their bio. Based on this task, [CDAMK23] formalizes a notion of social norm bias (SNoB). SNoB captures the extent to which predictions align with gender norms associated with specific occupations. In addition to gender-specific pronouns, these norms encompass other characteristics mentioned in the bios. They represent implicit expectations of how specific groups are expected to behave. [CDAMK23] characterizes SNoB as a form of algorithmic unfairness arising from the associations between an algorithm's predictions and individuals' adherence to inferred social norms. They also show that that adherence to or deviations from social norms can result in harm in many contexts and that SNoB can persist even after the application of some fairness interventions.

To quantify SNoB, the authors utilize the Spearman rank correlation coefficient $\rho(p_c, r_c)$, where p_c represents the fraction of bios associated with occupation c that mention the pronoun 'she', and r_c measures the correlation between occupation predictions and gender predictions. The authors employ separate one-vs-all classifiers for each occupation and obtain the occupation prediction for a given bio using these classifiers. For gender predictions, they train occupation-specific models to determine the gender-based group membership (female or not) based on a person's bio, and use the predictions from these models. A higher value of $\rho(p_c, r_c)$ represents a larger social norm bias, which indicates that in male-dominated occupations, the algorithm achieves higher accuracy on bios that align with inferred masculine norms, and vice-versa.

Table 10 shows results on the Bios data. We compare with a group fairness approach, Decoupled [DIKL18] that trains separate models for each gender, in order to mitigate gender bias. We see that CMID address SNoB bias better than ERM and Decoupled, achieving a lower $\rho(p_c, r_c)$ and improved accuracy.

Method	Accuracy	$\rho(p_c, r_c)$
ERM	0.95	0.66
Decoupled [DIKL18]	0.94	0.60
CMID	0.96	0.38

Table 10: Comparison on Bios data.

6 Additional Experiments

6.1 Effect of the Simple Model Architecture

In this section, we analyze the effect of using various simple model architectures on the performance of the model learned by CMID. As shown in Fig. 6, we consider four models for the Waterbirds dataset: a linear model, the shallow CNN (`2DConvNet1`) used in Table 2, a ResNet-18 pretrained on ImageNet and the ResNet-50 pretrained on ImageNet, which is also the architecture of

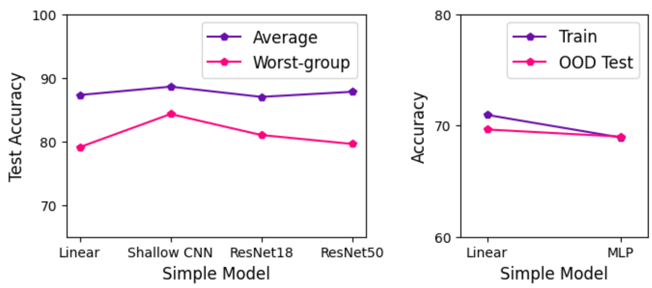

Figure 6: Performance of CMID using different simple model architectures for Waterbirds (Left) and CMNIST (Right) datasets.

the benchmark model. We also compare the results when using a linear model and an MLP as the simple model for the CMNIST dataset. We note that in these experiments, we only tune the learning rate (LR) and weight decay (λ_2) for training the simple model and keep the rest of the hyperparameter values for training the final model consistent across each dataset (details in the Appendix).

For the Waterbirds dataset, we observe that using a shallow CNN as the simple model is most effective for CMID. We believe that this is because a linear model is too simple to capture the surrogate feature, i.e. the background in this case. On the other hand, deep models like ResNet are complex enough to learn both types of features, while relying more strongly on the surrogate feature. As a result, regularizing the CMI with respect to a model that is too simple or too complex may not be effective in reducing reliance on spurious features, while improving reliance on task-relevant features.

6.2 CelebA: Invariant and Spurious Features have Similar Complexity

In our subgroup robustness experiment for CelebA (Table 9 in Section 5.3), we found that our method did not yield a significant improvement in worst-group accuracy. We investigate this further in this section. We show that for the CelebA dataset, the complexity of the invariant and surrogate features are actually quite similar. The experiment is similar to the experiments we did for CMNIST and Waterbirds in Section 2. We create a subset of the CelebA dataset by sampling an equal number of samples from all four subgroups. Table 11 presents a comparison of the results when predicting the invariant feature (hair color) and the surrogate feature (gender) using the simple model (`2DConvNet2`). We observe that the performance for both tasks is comparable, suggesting that the features exhibit similar complexity.

We contrast these results with those for CMNIST and Waterbirds in Table 2. For CMNIST and Waterbirds, there was a significant difference in the accuracy to which the simple model could predict the invariant and surrogate feature. For CelebA, the difference is much smaller which suggests that spurious features are *not* simpler than invariant features for this dataset—explaining why our method is not as effective for it.

Predict invariant feature		Predict surrogate feature	
Train	Test	Train	Test
89.1 ± 1.5	84.3 ± 0.6	92.4 ± 2.4	88.3 ± 1.6

Table 11: Comparison between performance for predicting the simple feature and the complex feature on CelebA dataset.

7 Discussion and Conclusion

We proposed a new framework (CMID) to mitigate simplicity bias, and showed that it yields improvements over ERM and many other previous approaches across a number of OOD generalization, robustness and fairness benchmarks. We conclude with some discussion on differences with related work, limitations of our approach, and directions for future work.

Differences with Related Work. In general, prior work focuses on one or two of the applications that we consider, while our approach proves effective across several datasets for all cases.

Among the methods that we compare with, methods such as IRM [ABGLP20], GDRO [SKHL20], PI [BCB21] and BLOOD [BCL22] require knowledge of group or environment labels and consequently may not be useful in cases where this knowledge is not available. Approaches such as CVaRDRO [DHN19] upweigh samples with higher loss values for DRO, and may lead to overfitting in the presence of label noise. Further, in cases where both the simple feature and the complex feature are close to fully predictive, methods such as [LHC+21a, CJZ21, DRL+23] may not be effective as these methods rely on incorrect predictions [LHC+21a, CJZ21] or high misclassification probability [DRL+23] of one model to train another model by upweighting such samples [LHC+21a, DRL+23] or learning features that generalize across such samples.

Our approach is much simpler than debiasing/feature diversification methods like [NCA+20, BCY+20, LHX22, ZLPB22, TALvdH22]. Many of these involve training two complex models simultaneously [NCA+20, BCY+20] or alternatively [LHX22], which can make these methods computationally more expensive. [ZLPB22, TALvdH22] involve training multiple models and returning the average or selecting the best one among them, respectively, while our results show that the simple approach to train a single model with the CMI regularization can prove effective across several settings. [PJFK23, LYF22] require access to additional data from a target domain to encourage disagreement between two models for feature diversification. In contrast, our approach does not have such a requirement and allows for diversification directly on the training data.

In addition, our approach is theoretically grounded and explicit in its assumptions, which could make it easier for a practitioner to evaluate and use. In particular, our definition of spurious features gives an explicit characterization of features/biases that we regard as undesirable and our approach seeks to reduce. It allows the user to understand the situations when the approach might be effective, and indeed we observe that this characterization proves effective in several settings.

Limitations. We note that like any other regularization or inductive bias, CMID may not be effective for *every* task. For certain tasks, spurious features as defined by us in Section 2 may not actually be spurious, and we may not always want to reduce reliance on features that are simple and highly predictive of the label. This seems to contradict the Occam's razor principle and the widely held notion that simple solutions generalize better. However, it is worth noting here that no inductive bias can work well in all cases. This is established by the No Free Lunch theorem [Wol96, SSBD14]: it is impossible for a single algorithm to perform well on all tasks. Given a finite amount of data, only algorithms with an appropriate inductive bias that is suitable for a given task can generalize well on that task. Consequently, while simplicity bias seems to explain in-distribution generalization of NNs in several cases, there are cases where it can prevent the model from learning more complex, task-relevant features, and in such cases, alleviating simplicity bias can be useful to encourage the learning of task-relevant features for better generalization under distributional shift. This is consistent with our experimental evaluations on several datasets from various modalities and domains.

Even in situations where mitigating simplicity bias is useful, there can be cases where the separation between the feature complexity of spurious and invariant features may not be very large. This can make the selection of an appropriate simple model class for our approach challenging. Indeed, in our experiments, we found that for the CelebA dataset, the spurious and task-relevant features are not significantly different in terms of complexity. Consequently, our approach does not lead to much improvement in the worst-group accuracy on this dataset.

We also note that in general, use of methods designed to improve OOD generalization and subgroup robustness can lead to a drop in the accuracy on the training set or the i.i.d. test set, compared to ERM. This is seen in some of our experimental results as well. This is because a model trained with the ERM objective relies strongly on the spurious feature(s), which is predictive on the train set as well as the i.i.d. test set. In contrast, improvement on OOD test sets is achieved by leveraging other features, which may be less predictive on the train set but generalize better.

Future Work. A natural direction of future work is to further explore the capabilities and limitations of our approach, and to also further understand its theoretical properties. More broadly, our work suggests auditing large models with respect to much simpler models can lead to improved properties of the larger models along certain robustness and fairness axes. It would be interesting to explore the power of similar approaches for other desiderata, and to understand its capabilities for fine-tuning large pre-trained models.

Acknowledgements

This work was supported by NSF CAREER Award CCF-2239265 and a Amazon Research Award. The authors acknowledge use of the Amazon Elastic Compute Cloud and USC Advanced Research Computing's Discovery cluster. BV thanks Puneesh Deora for helpful discussions and feedback, and Deqing Fu for discussion on simplicity bias.

References

[ABGLP20] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020. 1, 2, 3, 8, 9, 11, 12, 14

[AJB+17] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017. 3

[BCB21] Yujia Bao, Shiyu Chang, and Regina Barzilay. Predict then interpolate: A simple algorithm to learn stable classifiers. In *International Conference on Machine Learning*. PMLR, 2021. 1, 3, 11, 12, 14

[BCL22] Jun-Hyun Bae, Inchul Choi, and Minho Lee. BLOOD: Bi-level learning framework for out-of-distribution generalization, 2022. 1, 3, 11, 12, 14, 25, 26

[BCY+20] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 528–539. PMLR, 13–18 Jul 2020. 4, 14

[BDS+19] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, 2019. 2, 12

[BGM+19] Péter Bándi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, Quanzheng Li, Farhad Ghazvinian Zanjani, Svitlana Zinger, Keisuke Fukuta, Daisuke Komura, Vlado Ovtcharov, Shenghua Cheng, Shaoqun Zeng, Jeppe Thagaard, Anders B. Dahl, Huangjing Lin, Hao Chen, Ludwig Jacobsson, Martin Hedlund, Melih Çetin, Eren Halıcı, Hunter Jackson, Richard Chen, Fabian Both, Jörg Franke, Heidi Küsters-Vandevelde, Willem Vreuls, Peter Bult, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. From detection of individual metastases to classification of lymph node status at the patient level: The camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2019. 2, 11

[CDAMK23] Myra Cheng, Maria De-Arteaga, Lester Mackey, and Adam Tauman Kalai. Social norm bias: residual harms of fairness-aware algorithms. *Data Mining and Knowledge Discovery*, pages 1–27, 2023. 13, 27

[CJZ21] Elliot Creager, Joern-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2189–2200. PMLR, 18–24 Jul 2021. 1, 2, 3, 9, 11, 12, 14, 25, 26

[CZYJ20] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1448–1458. PMLR, 13–18 Jul 2020. 9

[DARW+19] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 120–128, New York, NY, USA, 2019. Association for Computing Machinery. 13, 27

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 27

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 11

[DHN19] John C. Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses against mixture covariate shifts, 2019. 3, 12, 13, 14

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery. 4

[DIKL18] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 119–133. PMLR, 23–24 Feb 2018. 4, 13, 27

[DLS+18] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. 2018. 12

[DRL+23] Nikolay Dagaev, Brett D. Roads, Xiaoliang Luo, Daniel N. Barry, Kaustubh R. Patil, and Bradley C. Love. A too-good-to-be-true prior to reduce shortcut reliance. *Pattern Recogn. Lett.*, 166(C):164–171, feb 2023. 4, 14

[FC19] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. 1

[GJM+20] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 3

[GRM+22] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2022. 3, 4, 10, 11

[GSK+22] Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-distribution robustness via targeted augmentations. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. 3

[GSL+18] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. 4

[HJKRR18] Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1939–1948. PMLR, 10–15 Jul 2018. 4

[HPS16] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA, 2016. Curran Associates Inc. 4

[HSNL18] Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, 2018. 4

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4, 27

[JMZC20] Shengyu Jia, Tao Meng, Jieyu Zhao, and Kai-Wei Chang. Mitigating gender bias amplification in distribution by posterior regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2936–2942, Online, July 2020. Association for Computational Linguistics. 4

[KC12] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, oct 2012. 4

[KIW22] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Machine Learning*, 2022. 3

[KSM+21] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 11, 12, 25, 26

[LBC+20] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 4, 12, 26

[LD21] Friedrich Leisch and Evgenia Dimitriadou. *mlbench: Machine Learning Benchmark Problems*, 2021. R package version 2.1-3.1. 12, 26

[LHC+21a] 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 Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–24 Jul 2021. 3, 11, 12, 13, 14, 25, 26, 27

[LHC+21b] Jiashuo Liu, Zheyuan Hu, Peng Cui, B. Li, and Zheyan Shen. Heterogeneous risk minimization. In *International Conference on Machine Learning*, 2021. 8, 9

[LHX22] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover andnbsp;mitigate unknown biases withnbsp;debiasing alternate networks. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, page 270–288, Berlin, Heidelberg, 2022. Springer-Verlag. 4, 14

[LLWT15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 2, 12

[LYF22] Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Learning from underspecified data. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. 4, 14

[MBJN23] Depen Morwani, Jatin Batra, Prateek Jain, and Praneeth Netrapalli. Simplicity bias in 1-hidden layer neural networks, 2023. 1, 3

[NCA+20] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20673–20684. Curran Associates, Inc., 2020. 3, 12, 13, 14

[NHBM98] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. Uci repository of machine learning databases, 1998. 12, 26

[NKB+20] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. 1

[NKK+19] Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. *SGD on Neural Networks Learns Functions of Increasing Complexity*. Curran Associates Inc., Red Hook, NY, USA, 2019. 1, 3

[PJFK23] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. In *The Eleventh International Conference on Learning Representations*, 2023. 4, 14

[PKB+21] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 3

[PSF18] Ji Ho Park, Jamin Shin, and Pascale Fung. Reducing gender bias in abusive language detection, 2018. 12

[QPIW23] Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. Simple and Fast Group Robustness by Automatic Feature Reweighting. *International Conference on Machine Learning (ICML)*, 2023. 3

[RRR21] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021. 7, 9

[SDA+20] Nimit Sohoni, Jared A. Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 3

[SDE+23] Amrith Setlur, Don Dennis, Benjamin Eysenbach, Aditi Raghunathan, Chelsea Finn, Virginia Smith, and Sergey Levine. Bitrate-constrained dro: Beyond worst case robustness to unknown group shifts, 2023. 3

[SKHL20] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 11, 12, 13, 14, 27

[SRKL20] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 7

[SSB+22] Nimit S. Sohoni, Maziar Sanjabi, Nicolas Ballas, Aditya Grover, Shaoliang Nie, Hamed Firooz, and Christopher Ré. Barack: Partially supervised group robustness with guarantees, 2022. 3

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014. 1, 15

[STR+20] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 1, 3, 10, 24

[TALvdH22] Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16761–16772, June 2022. 4, 14

[UMG20] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online, July 2020. Association for Computational Linguistics. 4

[VPCL19] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *International Conference on Learning Representations*, 2019. 3

[WBW+11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[WNB18] 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. 2, 12

[Wol96] David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, 1996. 1, 15

[WZY+19] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *International Conference on Computer Vision (ICCV)*, October 2019. 4

[XEIM20] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition, 2020. 10

[ZBH+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. 1

[ZLK+17] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 4

[ZLPB22] Jianyu Zhang, David Lopez-Paz, and Leon Bottou. Rich feature construction for the optimization-generalization dilemma. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26397–26411. PMLR, 17–23 Jul 2022. 4, 14

A Evaluating the Estimated CMI

In this section, we evaluate the reliability and scalability of the CMI estimate in (1) compared to the original CMI, which is computed with discretized model outputs. We consider CMNIST data with the hyperparameter values as mentioned in Section C.2.1 for the results in this section.

Table 12 compares the times (in milliseconds) to compute the original CMI and the estimated CMI, using batch size 64, for 10 classes and 200 classes. We see that the computation time for the estimated CMI does not increase significantly as the number of classes increases.

Method	10 classes	200 classes
$I(M, M_s\|Y)$	14.1 ± 3.3	23.3 ± 2.7
$\hat{I}_n(M, M_s\|Y)$	20.17 ± 4.2	24.3 ± 3.3

Table 12: Comparison of computation times (in milliseconds) for the original CMI and the estimated CMI in (1).

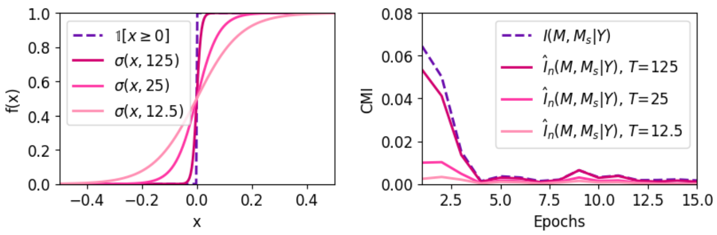

Figure 7: (Left) Sigmoid functions for different values of the temperature parameter T. (Right) Comparison of the original CMI with the estimated CMI using different values of T.

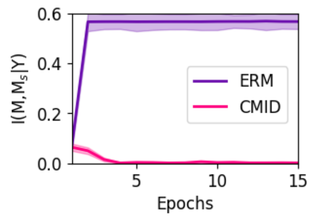

Figure 8: Comparison of the original CMI for models trained with ERM and CMID as a function of training time.

Fig. 7 compares the estimated CMI for different values of the temperature parameter T with the original CMI. The values shown in the figure are computed with batch size 1000, while for training, we use the estimated CMI with $T = 12.5$ and batch size 64. We see that as T increases, the sigmoid approximates the indicator function more closely and the estimated CMI tends closer to the original CMI.

Fig. 8 compares the CMI computed with discretized outputs using batch size 1000, when the model is trained with ERM or CMID. We see that regularizing the estimated CMI causes the original CMI to decrease over time, whereas training with ERM leads to an increase in the CMI. This shows that the solution learned by ERM makes predictions aligned with those of the simple model, whereas our approach CMID learns a model which attains low CMI with the simple model.

B Proofs for Section 4

In this section, we present the proofs for the theoretical results in Section 4.

B.1 Proof of Proposition 1

We first restate Proposition 1:

Proposition 1. *ERM($\mathcal{M}$) satisfies* $\frac{w_1}{w_2} = \frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1^2}$. *When* $\frac{\mu_1}{\mu_2'} \frac{\sigma_2'^2}{\sigma_1^2} < 1$, *ERM($\mathcal{M}_s$)=* $\left[0, \frac{\mu_2'}{\sigma_2^2}\right]$ *(upto scaling).*

Proof. We have:

$$\mathbb{E}\left(w_1 X_1 + w_2 X_2 - y\right)^2 = w_1^2(\sigma_1^2 + \mu_1^2) + w_2^2(\sigma_2^2 + \mu_2^2) + 1 - 2w_1\mu_1 - 2w_2\mu_2' + 2w_1 w_2 \mu_1 \mu_2'.$$

Since $\arg\min_{\boldsymbol{w} \in \mathcal{M}} \mathbb{E}\left(w_1 X_1 + w_2 X_2 - y\right)^2$ is a convex problem, to find the minimizer we set gradients with respect to w_1 and w_2 to be 0. Subsequently, we solve the resulting set of equations. By taking the gradient and setting it to 0, we obtain the following set of equations:

$$2w_1(\sigma_1^2 + \mu_1^2) - 2\mu_1 + 2w_2\mu_1\mu_2' = 0, \tag{6}$$

$$2w_2(\sigma_2^2 + \mu_2^2) - 2\mu_2' + 2w_1\mu_1\mu_2' = 0. \tag{7}$$

From (7), $w_2 = \mu_2' \dfrac{1 - w_1\mu_1}{\sigma_2^2 + \mu_2^2}$. Substituting this in (6) and solving for w_1, we get:

$$w_1 = \frac{\mu_1}{\sigma_1^2}\left(\frac{1}{\frac{\mu_1^2}{\sigma_1^2}+\frac{\mu_2'^2}{\sigma_2'^2}+1}\right),$$

where $\sigma_2'^2 = \sigma_2^2 + \mu_2^2 - \mu_2'^2$. Using this, we get the expression for w_2:

$$w_2 = \frac{\mu_2'}{\sigma_2'^2}\left(\frac{1}{\frac{\mu_1^2}{\sigma_1^2}+\frac{\mu_2'^2}{\sigma_2'^2}+1}\right).$$

Thus, for ERM($\mathcal{M}$), $\frac{w_1}{w_2} = \frac{\mu_1}{\mu_2'}\frac{\sigma_2'^2}{\sigma_1^2}$.

Next, consider ERM over $\mathcal{M}_s$. If $w_2 = 0$, we use (6) and get $\left[\frac{\mu_1}{\sigma_1^2+\mu_1^2}, 0\right]$ as the solution, for which the loss value is $\frac{\sigma_1^2}{\sigma_1^2+\mu_1^2}$. If $w_1 = 0$, we use (7) and get $\left[0, \frac{\mu_2'}{\sigma_2^2+\mu_2^2}\right]$ as the solution, for which the loss is $\frac{\sigma_2'^2}{\sigma_2'^2+\mu_2'^2}$. When $\frac{\mu_1^2}{\mu_2'^2}\frac{\sigma_2'^2}{\sigma_1^2} < 1$, the latter has a smaller loss and thus, ERM($\mathcal{M}_s$) is $\left[0, \frac{\mu_2'}{\sigma_2^2}\frac{1}{1+\frac{\mu_2^2}{\sigma_2^2}}\right]$.

□

B.2 Proof of Theorem 1

Theorem 1. *Let data be generated as per Assumption 1. For $\nu = 0.5\log(1 + c^2)$ for some c:*

1. *When $\frac{\mu_1}{\mu_2'}\frac{\sigma_2'^2}{\sigma_1\sigma_2} > \frac{1}{c}$, the solution to (2) is the same as ERM($\mathcal{M}$), so $\frac{w_1}{w_2} = \frac{\mu_1}{\mu_2'}\frac{\sigma_2'^2}{\sigma_1^2}$.*
2. *Otherwise, w_1 is upweighted and the solution to (2) satisfies $\frac{|w_1|}{|w_2|} = \frac{1}{c}\frac{\sigma_2}{\sigma_1}$.*

Proof. Consider the constraint $I(w_1X_1 + w_2X_2; w_2^*X_2|y) \leq \nu$. As we are working with continuous random variables in this setup, we employ differential entropy for our entropy computations. The entropy of a Gaussian random variable X with variance σ^2 is given by $H(X) = 0.5(\log(2\pi\sigma^2) + 1)$. Using this and the definitions of CMI and conditional entropy, we have:

$$\begin{aligned} I(w_1X_1 + w_2X_2, w_2^*X_2|y) &= H(w_1X_1 + w_2X_2|y) - H(w_1X_1 + w_2X_2|y, w_2^*X_2) \\ &= H(w_1X_1 + w_2X_2|y) - H(w_1X_1|y) \\ &= \tfrac{1}{2}\log\left(\tfrac{w_1^2\sigma_1^2 + w_2^2\sigma_2^2}{w_1^2\sigma_1^2}\right). \end{aligned}$$

Using $\nu = 0.5\log(1 + c^2)$, the constraint becomes: $\frac{w_2^2\sigma_2^2}{w_1^2\sigma_1^2} \leq c^2$. Thus, (2) reduces to solving:

$$\min_{w_1,w_2} \mathbb{E}\left(w_1X_1 + w_2X_2 - y\right)^2 \text{ s.t. } \tfrac{|w_2|\sigma_2}{|w_1|\sigma_1} \leq c.$$

If $\frac{\mu_2'}{\mu_1}\frac{\sigma_1\sigma_2}{\sigma_2'^2} < c$, ERM($\mathcal{M}$) satisfies the constraint and serves as the solution to (2). Otherwise, since this is a convex optimization problem with an affine constraint, the constraint must be tight. Therefore, we determine the solution by finding ERM($\mathcal{M}$) subject to $\frac{|w_2|\sigma_1}{|w_1|\sigma_2} = c$. The solution is given by:

$$w_1 = \frac{\mu_1}{\sigma_1^2}\left(\frac{\left(1+\frac{\mu_2'\sigma_1}{\mu_1\sigma_2}c\right)}{\frac{\mu_1^2}{\sigma_1^2}+1+c^2\left(\frac{\mu_2^2}{\sigma_2^2}+1\right)+\frac{2c\mu_1\mu_2'}{\sigma_1\sigma_2}}\right), \quad w_2 = c\frac{\sigma_1}{\sigma_2}\frac{\mu_1}{\sigma_1^2}\left(\frac{\left(1+\frac{\mu_2'\sigma_1}{\mu_1\sigma_2}c\right)}{\frac{\mu_1^2}{\sigma_1^2}+1+c^2\left(\frac{\mu_2^2}{\sigma_2^2}+1\right)+\frac{2c\mu_1\mu_2'}{\sigma_1\sigma_2}}\right).$$

□

B.3 Proof of Theorem 2

Theorem 2. *Let the data be generated as per Assumption 1 and (3). Let $\nu = 0.5\log(1 + c^2)$ for some sufficiently small c, and $\frac{\sigma_2^2}{\sigma_2'^2} = \frac{\sigma_3^2}{\sigma_3'^2}$. Then the solution to (4) satisfies: $\frac{|w_2\mu_2' + w_3\mu_3'|}{|w_1|\sigma_1} = c'$, where $c' = 2c\sqrt{\frac{\mu_2'^2}{\sigma_2^2} + \frac{\mu_3'^2}{\sigma_3^2}}$. Moreover, $w_2, w_3 \propto c'w_1$.*

Proof. Let $Z_1 = w_1X_1 + w_2X_2 + w_3X_3$ and $Z_2 = w_2^*X_2 + w_3^*X_3$. Then,

$$I(Z_1,\, Z_2|y) = H(Z_1|y) + H(Z_2|y) - H(Z_1, Z_2|y).$$

Since all features are Gaussian, we have:

$$\begin{aligned}
H(Z_1|y) &= 0.5(\log(w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + w_3^2\sigma_3^2) + \log(2\pi) + 1),\\
H(Z_2|y) &= 0.5(\log((w_2^*)^2\sigma_2^2 + (w_3^*)^2\sigma_3^2) + \log(2\pi) + 1),\\
H(Z_1, Z_2|y) &= 0.5\log|K| + \log(2\pi) + 1,
\end{aligned}$$

where K is the covariance matrix of Z_1 and Z_2 conditioned on y. We can calculate K as follows. Since all features are conditionally independent, we have:

- $\mathbb{E}(Z_1 - \mathbb{E}(Z_1))^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + w_3^2\sigma_3^2$.
- $\mathbb{E}(Z_1 - \mathbb{E}(Z_1))(Z_2 - \mathbb{E}(Z_2)) = w_2w_2^*\sigma_2^2 + w_3w_3^*\sigma_3^2$.
- $\mathbb{E}(Z_2 - \mathbb{E}(Z_2))^2 = (w_2^*)^2\sigma_2^2 + (w_3^*)^2\sigma_3^2$.

Thus, $|K| = (w_1^2\sigma_1^2+w_2^2\sigma_2^2+w_3^2\sigma_3^2)((w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2)-(w_2w_2^*\sigma_2^2+w_3w_3^*\sigma_3^2)^2$. Using these, the constraint becomes:

$$\begin{aligned}
&\log\left(\frac{(w_1^2\sigma_1^2+w_2^2\sigma_2^2+w_3^2\sigma_3^2)((w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2)}{(w_1^2\sigma_1^2+w_2^2\sigma_2^2+w_3^2\sigma_3^2)((w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2)-(w_2w_2^*\sigma_2^2+w_3w_3^*\sigma_3^2)^2}\right) \le \log(1+c^2)\\
&\implies \frac{(w_2w_2^*\sigma_2^2+w_3w_3^*\sigma_3^2)^2}{w_1^2\sigma_1^2((w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2)+\sigma_2^2\sigma_3^2(w_2^*w_3-w_3^*w_2)^2} \le c^2\\
&\implies \frac{(w_2w_2^*\sigma_2^2+w_3w_3^*\sigma_3^2)^2-c^2\sigma_2^2\sigma_3^2(w_2^*w_3-w_3^*w_2)^2}{w_1^2\sigma_1^2((w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2)} \le c^2.
\end{aligned}$$

Assume that c is sufficiently small, *i.e.*, $c \le \frac{\sqrt{3}|w_2w_2^*\sigma_2^2+w_3w_3^*\sigma_3^2|}{2\sigma_2\sigma_3|w_2w_3^*-w_3w_2^*|}$. Then, we get the condition:

$$\begin{aligned}
&\frac{(w_2w_2^*\sigma_2^2+w_3w_3^*\sigma_3^2)^2}{w_1^2\sigma_1^2((w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2)} \le 4c^2\\
&\implies \frac{\left|w_2\mu_2'\frac{w_2^*\sigma_2^2}{\mu_2'}+w_3\mu_3'\frac{w_3^*\sigma_3^2}{\mu_3'}\right|}{|w_1|\sigma_1} \le 2c\sqrt{(w_2^*)^2\sigma_2^2+(w_3^*)^2\sigma_3^2}.
\end{aligned}$$

When $[w_2^*, w_3^*] = v\left[\mu_2'/\sigma_2'^2, \mu_3'/\sigma_3'^2\right]$, where v is some constant, we get:

$$\frac{\left|w_2\mu_2'\frac{\sigma_2^2}{\sigma_2'^2}+w_3\mu_3'\frac{\sigma_3^2}{\sigma_3'^2}\right|}{|w_1|\sigma_1} \le 2c\sqrt{\frac{\mu_2'^2\sigma_2^2}{\sigma_2'^4}+\frac{\mu_3'^2\sigma_3^2}{\sigma_3'^4}}.$$

Since $\frac{\sigma_2^2}{\sigma_2'^2} = \frac{\sigma_3^2}{\sigma_3'^2}$, we get:

$$\frac{|w_2\mu_2'+w_3\mu_3'|}{|w_1|\sigma_1} \le 2c\sqrt{\frac{\mu_2'^2}{\sigma_2^2}+\frac{\mu_3'^2}{\sigma_3^2}} = c'.$$

When $\frac{\frac{\mu_2'^2}{\sigma_2'^2}+\frac{\mu_3'^2}{\sigma_3'^2}}{\frac{\mu_1}{\sigma_1}} \le c'$, ERM($\mathcal{M}$), *i.e.*, $[w_1, w_2, w_3] \propto \left[\frac{\mu_1}{\sigma_1^2}, \frac{\mu_2'}{\sigma_2'^2}, \frac{\mu_3'}{\sigma_3'^2}\right]$ satisfies the constraint, and thus is the solution to (4). Otherwise, since (4) is a convex problem with an affine constraint, the constraint must be tight. Therefore, we find ERM($\mathcal{M}$) subject to the equality constraint $|w_2\mu_2' + w_3\mu_3'| = c'\sigma_1|w_1|$. The solution is given by:

$$w_1 = \frac{\mu_1}{\sigma_1^2}\frac{1+c'\frac{\sigma_1}{\mu_1}}{1+\frac{\mu_1^2}{\sigma_1^2}+2c'\frac{\mu_1}{\sigma_1}+(c')^2\frac{1-\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}}{\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}+\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}-2\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}}},$$

$$w_2 = c'\sigma_1w_1\frac{\frac{\mu_2'}{\sigma_2^2+\mu_2^2}\left(1-\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}\right)}{\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}+\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}-2\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}},\quad w_3 = c'\sigma_1w_1\frac{\frac{\mu_3'}{\sigma_3^2+\mu_3^2}\left(1-\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}\right)}{\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}+\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}-2\frac{\mu_2'^2}{\sigma_2^2+\mu_2^2}\frac{\mu_3'^2}{\sigma_3^2+\mu_3^2}}.$$

□

B.4 Proof of Proposition 2

Proposition 2. *Let ERM*($\mathcal{M}_s$) $= M_s^*$*. Under Assumptions 2 and 3, the solution to the problem:*

$$\arg\min_{M \in \mathcal{M}} \mathbb{E}\, \ell_M(Z) \; s.t. \; I(M; M_s^* | Y) = 0 \tag{5}$$

is $M = \Phi^*$*, the maximal invariant predictor.*

Proof. Using Assumption 2, the class of simple models only contains variant predictors, so ERM($\mathcal{M}_s$) $= \Psi_s$. Consequently, the constraint in (5) can be written as $I(M; \Psi_s | Y) = 0$.

Considering the set of candidate predictors for M, namely $\{0, \Phi_c, \Psi_s, \Psi_c\}$, we examine the CMI constraint for each. Using the definition of mutual information, we have $I(0, \Psi_s|Y) = 0$ and $I(\Psi_s, \Psi_s|Y) = H(\Psi_s|Y)$. According to the definition of *variant* predictor, $H(\Psi_s|Y) > H(\Psi_s|Y, E) \geq 0$.

From Assumption 3, which states that the invariant and variant predictors are conditionally independent, we can deduce that $I(\Phi_c, \Psi_s|Y) = 0$. From Assumption 3, we also have $I(\Psi_c, \Psi_s|Y) > I(\Psi_c, \Psi_s|Y, E) \geq 0$.

Using these results, the feasible set is $[0, \Phi_c]$, which corresponds to the invariance set $\mathcal{I}_E(\mathcal{M})$. Consequently, problem (5) is equivalent to finding $\arg\max_{\Phi \in \mathcal{I}_E(\mathcal{M})} I(Y; \Phi)$. The solution to this problem is Φ_c, which represents the MIP Φ^*.

□

C Experimental Settings

We begin by describing some common details and notation that we use throughout this section. As in the main text, we use λ to represent the regularization strength for CMI. To ensure effective regularization, we adopt an epoch-dependent approach by scaling the regularization strength using the parameter S. Specifically, we set $\lambda = \lambda_c (1 + t/S)$ at epoch t. The temperature parameter T is set as 12.5 throughout the experiments. Additionally, we use LR to denote the learning rate, BS to denote the batch size, and λ_2 to denote the weight decay parameter, which represents the strength of ℓ_2-regularization. When using the Adam optimizer, we employ the default values for momentum.

The experiments on Slab data, CMNIST and CPMNIST data and Adult-Confounded data were implemented on Google Colab. The ImageNet-9 experiments were run on an AWS G4dn instance with one NVIDIA T4 GPU. For experiments on the subgroup robustness datasets and the Camelyon17-WILDS data, we used two NVIDIA V100 GPUs with 32 GB memory each. We only used CPU cores for the Bios data experiments.

C.1 Mitigating Simplicity Bias Experiments

This section includes the details for the experiments showing that CMID mitigates simplicity bias where we use the Slab data and the ImageNet-9 data.

C.1.1 Slab Data

Dataset. All the features in the 3-Slab and 5-Slab data are in the range $[-1, 1]$. The features are generated by defining the range of the slabs along each direction and then sampling points in that range uniformly at random. The base code for data generation came from the official implementation of [STR+20] available at https://github.com/harshays/simplicitybiaspitfalls. We consider 10^5 training samples and 5×10^4 test samples. In both the cases, the linear margin is set as 0.05. The 3-Slab data is 10-dimensional, where the remaining 8 coordinates are standard Gaussians, and are not predictive of the label. The slab margin is set as 0.075. The 5-Slab data is only 2-dimensional, and the slab margin is set as 0.14.

Training. We consider a linear model for the simple model and following [STR+20], a 1-hidden layer NN with 100 hidden units as the final model (for both ERM and CMID). Throughout, we use SGD with BS $= 500$, $\lambda_2 = 5 \times 10^{-4}$ for training. The linear model is trained with LR $= 0.05$, while the NN is trained with LR $= 0.005$.

For the 3-Slab data, the models are trained for 300 epochs. We consider $\lambda_c \in \{100, 150, 200\}$ and choose $\lambda_c = 150$ for the final result. For the 5-Slab data, the models are trained for 200 epochs and we use a 0.99 momentum in this case. We consider $\lambda_c \in \{1000, 2000, 2500, 3000\}$ and choose $\lambda_c = 3000$ for the final result. Note that we consider significantly high values of λ_c for this dataset compared to the rest because the simple model is perfectly predictive of the label in this case. This implies that its CMI with the final model is very small, and the regularization strength needs to be large in order for this term to contribute to the loss.

C.1.2 Texture vs Shape Bias on ImageNet-9

We use ResNet50 pretrained on ImageNet data as the simple model, and train it on ImageNet-9 using Adam with $\text{LR} = 0.001, \text{BS} = 32, \lambda_2 = 10^{-4}$ for 10 epochs. We do not consider a simpler architecture and training from scratch since this pre-trained model already exhibits texture bias. For the final model, we consider the same model and parameters, except we use SGD with 0.9 momentum as the optimizer and $\lambda_2 = 0.001$ for both ERM and CMID and train for 10 epochs. Values of CMID specific parameters were $\lambda_c = 15, S = 10$. For tuning, we consider $\text{LR} \in \{10^{-5}, 10^{-4}, 10^{-3}\}$ for both the models and $\text{BS} \in \{16, 32\}, \lambda_2 \in \{0.0001, 0.001, 0.01\}$ and $\lambda_c \in \{0.5, 15, 25, 50\}$.

C.2 Better OOD Generalization Experiments

This section includes the details for the experiments showing that our approach leads to better OOD generalization. For this, we used CMNIST and CPMNIST, Camelyon17-WILDS and Adult-Confounded datasets.

C.2.1 CMNIST and CPMNIST

Dataset. Following [BCL22], we use $25,000$ MNIST images (from the official train split) for each of the training environments, and the remaining $10,000$ images to construct a validation set. For both the test sets, we use the $10,000$ images from the official test split.

Training. The details about model architecture and parameters for training the simple model with ERM and the final model with CMID, for both the datasets, are listed in Table 13. Following [BCL22], the MLP has one hidden layer with 390 units and ReLU activation function. In both the cases, the simple model is trained for 4 epochs, while the final model is trained for 20 epochs. We choose the model with the smallest accuracy gap between the training and validation sets. For tuning, we consider the following values for each parameter: for the simple model, $\text{LR} \in \{0.005, 0.01, 0.05\}$, $\lambda_2 \in \{0.001, 0.005, 0.01\}$, and for the final model, $\text{LR} \in \{0.001, 0.005\}$, $\lambda_c \in [3, 8]$ and $S \in [3, 6]$, where lower values of S were tried for higher values of λ_c and vice-versa.

For the final results, we report the mean and standard deviation by averaging over 4 runs. For comparison, we consider the results reported by [BCL22] for all methods, except EIIL [CJZ21] and JTT [LHC+21a]. Results for EIIL are obtained by using their publicly available implementation for CMNIST data (available at https://github.com/ecreager/eiil), and incorporating the CPMNIST data into their implementation. The hyperparameter values in their implementation are kept the same. We implement JTT to obtain the results. We consider $\text{LR} \in \{0.001, 0.005, 0.01\}$ and the parameter for upweighting minority groups [LHC+21a] $\lambda_{up} \in \{5, 10, 15, 20, 25\}$ for tuning.

Dataset	Simple Model	Optimizer	LR	BS	λ_2	Final Model	Optimizer	LR	BS	λ_c	S
CMNIST	Linear	SGD	0.01	64	0.005	MLP	SGD	0.001	64	4	4
CPMNIST	Linear	SGD	0.01	64	0.001	MLP	SGD	0.005	64	5	3

Table 13: Training details for CMNIST and CPMNIST.

C.2.2 Camelyon17-WILDS

Dataset. Camelyon17-WILDS [KSM+21] contains 96×96 image patches which may or may not display tumor tissue in the central region. We use the same dataset as Bae et al. [BCL22], which includes $302,436$ training patches, $34,904$ OOD validation patches, and $85,054$ OOD test patches, where no two data-splits contain images from overlapping hospitals. We use the WILDS package, available at https://github.com/p-lambda/wilds for dataloading.

Training. For the simple model, we train a `2DConvNet1` model (see Section C.4.1 for details) for 10 epochs. We use the Adam optimizer with $\text{LR} = 10^{-4}, \text{BS} = 32, \lambda_2 = 10^{-4}$. For the final model, we train a DenseNet121 (randomly initialized, no pretraining) for 5 epochs using SGD with 0.9 momentum with $\text{LR} = 10^{-4}, \text{BS} = 32, \lambda_2 = 0.01$ and $\lambda_c = 0.5, S = 10$. We use the same BS and λ_2 values as [BCL22] for consistency. For tuning, we consider $\text{LR} \in \{10^{-5}, 10^{-4}, 10^{-3}\}$ for both the models and $\lambda_c \in \{0.5, 2, 5, 15\}$ for CMID. While [BCL22] and [KSM+21] use learning rates 10^{-5} and 0.001, respectively, we found a learning rate of 10^{-4} was most suited for our approach. We select the model with the highest average accuracy on the validation set to report the final results. For comparison, we use the results reported by [BCL22].

C.3 Adult-Confounded

Dataset. The UCI Adult dataset [NHBM98, LD21], comprises 48,842 census records collected from the USA in 1994. Following [CJZ21], we use the original train/test splits from UCI Adult as well as the same subgroup sizes, but individual examples are under/over-sampled using importance weights based on the correlation on the original data and the desired correlation for the Adult-Confounded dataset.

Training. We use a linear model (with a bias term) as the simple model, and following [CJZ21] use an Adagrad optimizer throughout. We use BS = 50. The simple model is trained for 50 epochs, with $\text{LR} = 0.05, \lambda_2 = 0.001$. Following [CJZ21, LBC+20], we use a two-hidden-layer MLP architecture for the final model, with 64 and 32 hidden units, respectively. It is trained with $\text{LR} = 0.04, \lambda_c = 4, S = 4$ for 10 epochs. We also construct a small validation set from the train split by randomly selecting a small fraction of samples $(5 - 50)$ from each subgroup (depending on its size) and then upsampling these samples to get balanced subgroups of size 50. We choose the model with lowest accuracy gap between the train and validation sets. For tuning, we consider $\text{LR} \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\lambda_c, S \in \{3, 4, 5\}$. For the final results, we report the mean and accuracy by averaging over 4 runs. For comparison, we reproduced the results for ERM from [CJZ21], and thus, consider the values reported in [CJZ21] for ERM, ARL and EIIL. We implement JTT to obtain the results. We use $LR = 0.05$ and $\lambda_2 = 0.001$ to train the final model and tune the parameter for upweighting minority groups [LHC+21a] λ_{up} over $\{10, 20, 25, 30\}$. The remaining parameters are kept the same as for our approach.

C.4 Subgroup Robustness Experiments

This section includes the details for the experiments showing that CMID enhances subgroup robustness, where we use four benchmark datasets: Waterbirds, CelebA, MultiNLI and CivilComments.

C.4.1 Model Architectures

In this section, we discuss the architectures we consider for the simple models for this task. For the two image datasets, a shallow 2D CNN is a natural choice for the simple model as 2D CNNs can capture local patterns and spatial dependencies in grid-like data. On the other hand, for the two text datasets with tokenized representations, we consider a shallow MLP or 1D CNN for the simple model. MLPs can capture high-level relationships between tokens by treating each token as a separate feature, while 1D CNNs can capture local patterns and dependencies in sequential data.

Figure 9: Left: `2DConvNet1` and Right: `2DConvNet2` architectures.

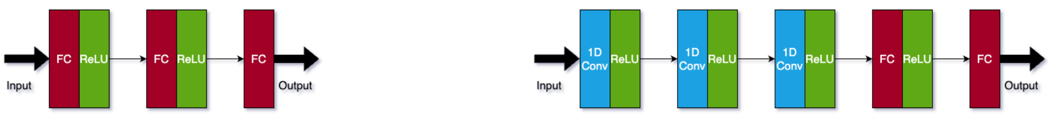

Figure 10: Left: `2MLP` and Right: `1DConvNet` architectures.

Next, we describe the details for the model architectures. Let F denote the filter size and C denote the number of output channels (for convolutional layers) or the output dimension (for linear/FC layers). Throughout, we use $F = 2$ for the average pooling layers. Fig. 9 shows the `2DConvNet1` and the `2DConvNet2` architecture, which were used as simple models for Waterbirds and CelebA, respectively. These were the only two architectures we considered for the 2DCNN on these datasets. In `2DConvNet1`, the 2D convolutional layers use $F = 7, C = 10$ and $F = 4, C = 20$, respectively, while $C = 2000$ for the FC layer. In the `2DConvNet2` architecture, $F = 5, C = 10$ for the 2D convolutional layer and $C = 500$ for the FC layer. Fig. 10 shows the `2MLP` and the `1DConvNet` architecture, which were used as simple models for MultiNLI and CivilComments-WILDS, respectively. For tuning, we considered these models as well as a 1DCNN with one less 1D convolutional layer than `1DConvNet` for both the datasets. In `2MLP`, the fully connected (FC) layers use $C = 100$ and $C = 25$, respectively. In the `1DConvNet` architecture, the 1D convolutional layers use $F = 7, C = 10$, $F = 5, C = 32$ and $F = 5, C = 64$, respectively, while $C = 500$ for the FC layer.

C.4.2 Training Details

We utilize the official implementation of [SKHL20] available at https://github.com/kohpangwei/group_DRO as baseline code and integrate our approach into it. Most hyperparameter values are kept unchanged, and we list the the important parameters along with model architectures for all the datasets in Table 14. For the simple models, we consider shallow 2D CNNs for the image datasets, and MLP and 1D CNN for the text data, as discussed in the previous section. In all cases, the simple model is trained for 20 epochs. For the final model, following [SKHL20], we use the Pytorch `torchvision` implementation of ResNet50 [HZRS16] with pretrained weights on ImageNet data for the image datasets, and the Hugging Face `pytorch-transformers` implementation of the BERT `bert-base-uncased` model, with pretrained weights [DCLT19] for the language-based datasets.

Dataset	Simple Model	Optimizer	LR	BS	λ_2	Final Model	Optimizer	LR	BS	λ_2	λ_c	S	# epochs
Waterbirds	`2DConvNet1`	Adam	10^{-5}	32	10^{-4}	ResNet50	SGD	5×10^{-4}	128	10^{-4}	20	4	100
CelebA	`2DConvNet2`	Adam	10^{-5}	32	5×10^{-4}	ResNet50	SGD	3×10^{-4}	128	0.001	10	5	50
MultiNLI	`2MLP`	Adam	0.005	16	10^{-4}	BERT	AdamW	5×10^{-5}	32	0	75	10	5
CivilComments	`1DConvNet`	Adam	10^{-4}	16	10^{-4}	BERT	AdamW	10^{-5}	32	0.001	25	10	10

Table 14: Training details for subgroup robustness datasets.

Dataset	LR	λ_c	S
Waterbirds, CelebA	$[1, 5] \times 10^{-4}$	$\{10, 15, 20, 25, 50, 75\}$	$\{4, 5, 6, 8, 10\}$
MultiNLI, CivilComments	$\{1, 2, 5\} \times 10^{-5}$	$\{10, 25, 50, 75\}$	$\{10\}$

Table 15: Values considered for tuning the hyperparameters for training the final model for the four subgroup robustness datasets.

Table 15 shows the values of LR, λ_c and S we consider for tuning for the final model. Following [SKHL20], we keep $\lambda_2 = 0$ for MultiNLI. For the rest, we consider $\lambda_2 \in \{0.0001, 0.0005, 0.001\}$. For results, we choose the model with the best worst-group accuracy on the validation set. For comparison, we consider the values reported in [LHC+21a].

C.5 Bias in Occupation Prediction Experiment

We use a version of the Bios data shared by the authors of [DARW+19]. We used the official implementation of [CDAMK23], available at https://github.com/pinkvelvet9/snobpaper, to obtain results and for comparison purposes. In this implementation, they consider 25 occupations and train separate one-vs-all linear classifiers for each occupation based on word embeddings to make predictions. We directly used their implementation to obtain results for ERM and Decoupled [DIKL18] on the data.

For our approach, we employed linear models for both the simple model and the final model. We directly regularized the CMI with respect to the ERM from their implementation. The final model was trained for 5 epochs using SGD with LR $= 0.1$, BS $= 128$, $\lambda_c = 5$, $S = 5$. We only tuned the LR for this case, considering values of 0.05 and 0.1.

C.6 Effect of Simple Model Architecture Experiment

We tune the learning rate (LR) and weight decay (λ_2) for training the simple model (as listed in Table 16). The rest of the hyperparameter values for training the final model are kept consistent across each dataset (as listed in Table 13 for CMNIST and Table 15 for Waterbirds).

	Waterbirds				CMNIST	
	Linear	Shallow CNN	ResNet18	ResNet50	Linear	MLP
LR, λ_2						

Table 16: Training details for the simple models.