Overcoming Simplicity Bias in Deep Networks using a Feature Sieve

Rishabh Tiwari [1] **Pradeep Shenoy** [1]

Abstract

Simplicity bias is the concerning tendency of deep networks to over-depend on simple, weakly predictive features, to the exclusion of stronger, more complex features. This is exacerbated in real-world applications by limited training data and spurious feature-label correlations, leading to biased, incorrect predictions. We propose a direct, interventional method for addressing simplicity bias in DNNs, which we call the *feature sieve*. We aim to automatically identify and suppress easily-computable spurious features in lower layers of the network, thereby allowing the higher network levels to extract and utilize richer, more meaningful representations. We provide concrete evidence of this differential suppression & enhancement of *relevant* features on both controlled datasets and real-world images, and report substantial gains on many real-world debiasing benchmarks (11.4% relative gain on Imagenet-A; 3.2% on BAR, etc). Crucially, we do not depend on prior knowledge of spurious attributes or features, and in fact outperform many baselines that explicitly incorporate such information. We believe that our *feature sieve* work opens up exciting new research directions in automated adversarial feature extraction and representation learning for deep networks.

1. Introduction

Deep networks are known to be vulnerable to a number of failure modes; in particular, *simplicity bias* is the tendency of DNNs to prioritize weak predictive features over stronger, more difficult-to-extract features (Shah et al., 2020). This bias has been studied analytically (Pezeshki et al., 2021) as well as empirically using natural images (texture bias (Geirhos et al., 2018)) and carefully controlled synthetic datasets (Hermann & Lampinen, 2020) that independently manipulate feature complexity and predictive power. Such learning biases have significant real-world consequences too, resulting for instance in biased decision-making in AI-assisted workflows for face recognition, healthcare, credit rating, etc. Figure 1 illustrates the idea behind simplicity bias, and some real-world consequences. As a result, much recent work aims to *debias* neural network models via a variety of approaches to achieve more equitable outcomes (Mehrabi et al., 2021; Zafar et al., 2017; Dwork et al., 2012; Russell et al., 2017).

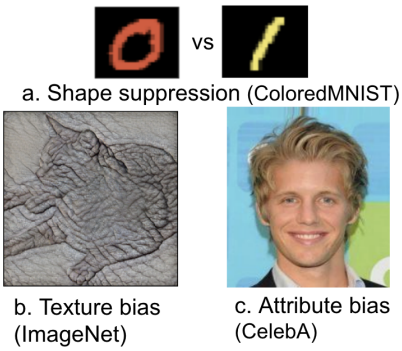

Figure 1. Simplicity bias and spurious features. a) DNNs focus on color to the exclusion of shape when both are predictive. b) Image misclassified as elephant due to overdependence on texture features (adapted from (Geirhos et al., 2018)). c) Classifiers mislabel blond-haired male faces as female.

Previous approaches towards debiasing DNNs include data manipulation via augmentation & adversarial training (Duboudin et al., 2022; Niu et al., 2022), data reweighting (Nam et al., 2020), multiple training environments (Arjovsky et al., 2019; Zhou et al., 2022), robust learning (Pezeshki et al., 2021), and fairness objectives (Li et al., 2022). Other researchers have proposed diversity-enhanced ensembles (Kim et al., 2022; Teney et al., 2022; Niu et al., 2022) and architecture optimization (Bai et al., 2021b).

We propose a novel, direct approach towards addressing simplicity bias in neural networks: an adversarial *learning challenge* that forces the network to learn sophisticated feature representations. We refer to this learning challenge as a *feature sieve*, and enforce it through the use of an auxiliary

[1]Google Research India. Correspondence to: Pradeep Shenoy <shenoypradeep@google.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

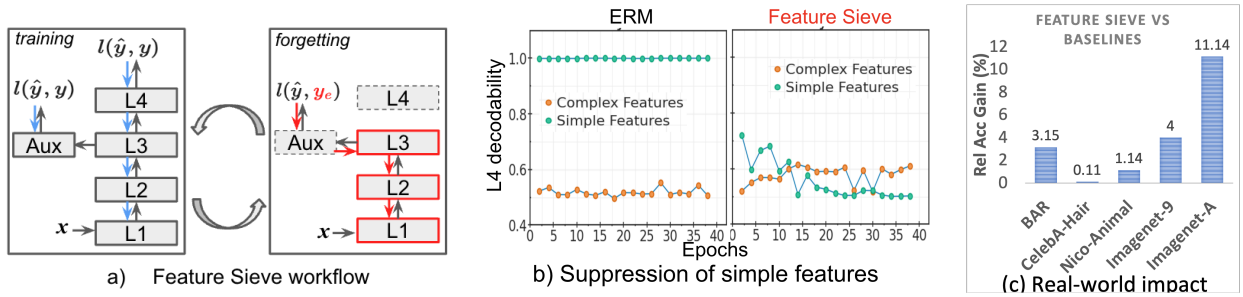

Figure 2. SIFER workflow and results. a) We use an auxiliary network to alternately identify predictive features and erase them *only at lower network layers*. By positioning the auxiliary network at different depths, we control the complexity of erased features. See Section 3 for details. b) Our approach successfully suppresses digit and *enhances CIFAR decodability* at higher layers for CIFAR_MNIST dataset. c) We show significant gains over other approaches on many real-world debiasing benchmarks.

network (Figure 2a;b). Our primary intuition is that simple features are computable early in the neural network, and proliferate throughout the deeper layers, thereby hindering the learning of complex features. We therefore propose to use the auxiliary network to alternately *predict* labels using available features at some intermediate level (i.e., identify simple predictive features), and *erase* those features from the early layers of the network, using a "forgetting loss" (see Section 3 for details). Critically, our proposal does not depend on any specific definition or complexity class of "simple features", and instead automatically adapts to data characteristics using generalization error estimates.

We explicate our approach and its inner workings using experiments on controlled datasets (CMNIST, CIFAR_MINST), and demonstrate its practical value on real-world debiasing benchmarks including BAR (Nam et al., 2020), CelebA (Liu et al., 2018), NICO (He et al., 2021), ImageNet-9 (Xiao et al., 2020) and ImageNet-A (Hendrycks et al., 2021); in nearly all experiments we show substantial gains over other competitive approaches. Figure 2c provides a quick visual summary of our findings.

Summing up, we propose SIFER: **Si**eving **Fe**atures for **R**obust learning, a novel approach towards mitigating simplicity bias, thereby debiasing neural networks from spurious correlations in data. Our contributions are listed below.

- We propose and formalize the idea of a feature sieve for mitigating simplicity bias, and provide an automated learning recipe to control feature complexity based on validation set.
- We show, using controlled datasets, the effectiveness of our approach in enhancing the decodability of complex features. We also demonstrate the customizability of our approach–our work is not restricted only to suppressing "simple" features, but is more broadly a controllable feature tradeoff tool.
- We show significant gains in debiasing classifiers on real-world datasets: 3.2%, 4%, 11.1% relative gains over baselines on BAR, ImageNet-9, ImageNet-A (Figure 2c). Crucially, we **do not use foreknowledge of biased features / input dimensions** in obtaining these results, unlike many of the baselines we outperform.
- Finally, we show using feature importance visualizations that SIFER is able to correctly identify important visual features of a scene, while suppressing irrelevant but spuriously-label-correlated background features (Figure 4); this underscores the relevance of SIFER to real-world feature understanding.

We hope that our work with SIFER[1] encourages further work in designing interesting computational barriers for neural networks; by automating the extraction and combination of diverse features ordered by complexity and predictive power, we could make significant progress towards the debiasing of machine learning models.

2. Related Work

2.1. Simplicity Bias

Shah et al. (2020) showed that neural networks trained with SGD are biased to learn the simplest predictive features in the data while ignoring others. Numerous studies have attempted to investigate the correlation and impact of such shortcuts, yielding a wealth of intriguing findings (Nagarajan et al., 2020; Hermann & Lampinen, 2020).

2.2. Debiasing Spurious Correlations

Unlike our work, the majority of previous work on mitigating simplicity bias uses explicit biased-attribute labels (Kim

[1]Code available at https://github.com/google-research/google-research/sifer

et al., 2019; Li & Vasconcelos, 2019; Sagawa et al., 2019; Teney et al., 2020; Krueger et al., 2021; Bai et al., 2021a) in their debiasing recipes. This reduces their practicality since both identifying, and manually labeling biased instances and dimensions in real-life data are significant barriers. Only recently, the focus has shifted towards debiasing without using explicit attribute labels (Teney et al., 2022; Kim et al., 2022; Niu et al., 2022; Shrestha et al., 2022; Nam et al., 2020). Here we discuss different technical approaches used by previous work in both of the above directions:

Alternate Networks: LfF (Nam et al., 2020) and LWBC (Kim et al., 2022) initially train a prejudiced network and try to debias the second network by focusing on samples that go against the bias.
Ensemble: LWBC (Kim et al., 2022) and ESB (Teney et al., 2022) both create a classifier ensemble; the former enforces debiasing via reweighting of training instances, and the latter incorporates a diversity constraint in the ensemble.
Architecture Design: NAS-OoD (Bai et al., 2021b) adds an OOD generalization criterion to network architecture search training to select inherently more robust network architectures. OccamNet (Shrestha et al., 2022) adds a few inductive biases in the network–for instance, explaining the dataset with simple hypotheses and bounded network depth, and applying spatial localization assumptions about unbiased (visual) features in order to filter spurious features.
Multiple Environments: IRM (Arjovsky et al., 2019) uses the theory of causal bayesian networks to find an invariant feature representation using multiple training environments with different bias correlations. REx (Krueger et al., 2021) tries to improve on the worst linear combinations of risks from different training environments. CaaM (Wang et al., 2021) learns causal attention by partitioning the data on-the-go to break correlation with bias.
Augmentations: DecAug (Bai et al., 2021a) proposed a semantic augmentation and feature decomposition approach to disentangle context features from category related features. Niu et al. (2022) adds adversarial augmentations to the image while training to avoid over-reliance on spurious visual cues. This work is conceptually closest to our work, in that it builds an ensemble where previous components compete with a new classifier to encourage it to learn diverse hypotheses. Our approach directly addresses the competitive development of features within a network (the "heart" of the simplicity bias challenge); we also outperform them on the BAR dataset (Nam et al., 2020) (Table 3), while being more computationally parsimonious.

3. SIFER: a Feature Sieve for Bias Mitigation

3.1. Preliminaries & Intuition

We start from the assumption that simple features are (by definition) quickly learned, available early in the neural network stack (i.e., in layers closer to the input), and more easily proliferate throughout the subsequent layers (see e.g., Hermann & Lampinen (2020) for substantial supportive evidence for these assumptions). Further, the ubiquitous presence of simple features actively prevents acquisition of more complex hypotheses by subsequent NN layers, due to the so-called simplicity bias inherent in NN training methods–see e.g., Shah et al. (2020); Pezeshki et al. (2021) for theoretical results supporting these claims.

Thus, our primary goal is to *identify and actively suppress* simple / spurious predictive features, so as to create room for the learning of complex predictive features at higher layers of the NN–an approach we refer to as a "feature sieve".

We include another key consideration in the design of our approach: do not leverage any *a priori* information of simple features, or even the function class / degree of complexity of simple features. To support this design goal, we a) build into our design the knobs that control tradeoffs between simpler- and more complex-to-compute features, and b) focus on reducing generalization error as the objective in setting these knobs. This allows us to not only automatically discover useful tradeoffs, but also to ensure that our trained classifiers are overall more accurate than standard baselines.

As a final remark, we note that the distinctions between simple / complex, spurious / accurate, early-layer / late-layer, and early-acquisition / late-acquisition are likely substantially more nuanced than a simple one-to-one correspondence, even though they are often used interchangeably for ease of exposition. For instance, depending on the dataset, a "simple" feature may in fact be the best / most unbiased predictive feature. For this reason, too, depending upon generalization error for controlling the feature sieve is strongly preferred to the use of any stronger inductive bias along the dimensions mentioned above.

3.2. The Alternating Identify-and-Erase Workflow

Figure 2(a) provides an overview of SIFER. Briefly, we use an *auxiliary network*, working at an intermediate level of representation in the neural network, to identify predictive features (simple / spurious) in the representation, and subsequently to erase them at the lower layers of the primary network. This is a direct operationalization of our primary goal stated above.

Identifying simple features: The training of the primary network proceeds in conventional fashion via forward- and back-propagation (Figure 2(a), left panel, black & blue arrows respectively), with an additional auxiliary layer that learns to predict the label from an intermediate representation. Note that feedback from the auxiliary layer does not back-propogate to the main network. This is a conscious decision choice to force the auxiliary layer to learn from

already-available features rather than create or reinforce them in the main network. By controlling the auxiliary network's capacity and the layer of the primary network to which it is attached, we can control the complexity of the predictive features it can identify.

Applying the feature sieve: We aim to *erase* the identified features in the early layers of the neural network, by the combination of the following steps: a) The parameters of auxiliary layer ($\mathcal{A}$) are frozen, and only that portion of the main network ($\mathcal{M}_d$) which is before the auxiliary layer is kept trainable– this is the region where we wish to "forget" the simple features, and b) We apply a *forgetting loss* ($\mathcal{L}_f$) at the output layer of the auxiliary network.

$$\hat{\mathbf{y}}_{aux} = \mathcal{A}(\mathcal{M}_d(\mathbf{x})) \quad (1)$$

$$\mathbf{y}_{ep} = [\frac{1}{n}, \frac{1}{n}, ...] \quad \textit{(n entries)} \quad (2)$$

$$\mathcal{L}_f = \texttt{CE}(\hat{\mathbf{y}}_{aux}, \mathbf{y}_{ep}) \quad (3)$$

where $\mathbf{x}$, $\mathbf{y}_{ep}$, $\hat{\mathbf{y}}_{aux}$ and n represent input images, a pseudo-label with uniform probability across classes, the prediction from auxiliary layer, and number of classes respectively.

Iterative optimization: A challenge is that this process of identification and sieving is dynamic in nature; in particular, the two steps may interfere with each other. In order to handle this challenge, we *interleave* the two steps such that each forgetting step happens after regular intervals of some mini-batch iterations ($\mathcal{F}$) which is treated as a hyperparameter selected using the validation set.

The entire learning recipe is summed up in Algorithm 1

Algorithm 1 SIFER: Mitigating simplicity bias

Input : Pretrained Model Weights $\mathbf{W}$;
training data $\mathcal{D}$; training iters N
Hparams : Aux Depth $\mathcal{A}_D$; Aux Position $\mathcal{A}_P$
main_lr_weight α_1; aux_lr_weight α_2
aux_forget_weight α_3; forget_after_iters $\mathcal{F}$
Output : robust model weights $\mathbf{W}$

```
for k = 1 ... N do
    (x, y) ← sample(D)
    ŷ, ŷ_aux ← Forward_with_aux(x, A_D, A_P, W)
    L_1 ← CE(ŷ, y)
    L_2 ← CE(ŷ_aux, y)
    L_f ← CE(ŷ_aux, U)
    L ← α_1 L_1 + α_2 L_2
    if k % F == 0 then
        L ← L + α_3 L_f
    end
    ∇W ← Backward(L)
    W ← OptimizeStep(∇W)
end
```

3.3. Controllability of the Feature Sieve

As remarked earlier, we aim to automatically discover notions of and tradeoffs between so-called simple and complex features, as relevant for the specific dataset at hand. The feature sieve approach described here allows for many mechanisms to control this discovery & tradeoff. The primary parameters are the position & depth of the auxiliary network ($\mathcal{A}_P, \mathcal{A}_D$) which implicitly control the function complexity of the features available for discovery by the auxiliary network; and the auxiliary forgetting weight α_3, which controls the degree to which the discovered features are suppressed. The interleaving of the feature identifying & feature sieving steps is controlled by the parameter $\mathcal{F}$–again, based on the specific dataset and the nature of the features contained, this controls the dynamics of the training procedure.

Finally, we set these hyperparameters based on the goal of minimizing validation error–this ensures not only that the parameters are chosen using unbiased estimates of generalization, but also that at a minimum, we perform better than the standard training baseline (which, as the trivial solution of not-forgetting, is included in the search space for the feature sieve).

4. Experiment Setup

4.1. Datasets for Studying Simplicity Bias

CMNIST: Colored-MNIST is a 2-class synthetic dataset used to study simplicity bias. We use digits 0 & 1 respectively from the MNIST dataset, with an added color channel (red for 0 images, green for 1).

CIFAR-MNIST: Similar to CMNIST, this binary classification dataset has paired-composite images–Class 0 pairing MNIST 0s with CIFAR automobiles, and Class 1 pairing MNIST 1s with CIFAR truck images.

Both datasets contain perfectly predictive simple *and* complex features; by training a classifier and manipulating the test set to break one of these correlations, one can examine which features are being used by the trained classifier.

4.2. Real-World Debiasing Benchmarks

BAR: Biased Activity Recognition (Nam et al., 2020) is a real-world image benchmark for classifying human actions (images) into 6 classes; each training image contains spurious correlations with background features (e.g., rocks with climbing). The test set contains the same set of actions but with different backgrounds (e.g., ice with climbing). The training data has no bias-conflicting examples (i.e., examples which violate the spurious correlation), which makes this a challenging benchmark.

CelebA: CelebA (Liu et al., 2018) contains human faces,

each labelled with 40 attributes. Following Kim et al. (2022), we focus on predicting `HairColor`, an attribute heavily correlated with `Gender` in the dataset. Specifically, most CelebA images with `blond-hair` (more than 99% of them in the training set) are women.

NICO: NICO (He et al., 2021) is a real-world benchmark for out-of-distribution robustness. Following Bai et al. (2021a), we used its Animal subset containing 10 object classes and 10 context labels. The training set only contains 7 contexts for each object class while the validation and test set contains 3 extra unseen contexts (total 10). Unlike the majority of the baselines, we don't use context label attributes in train, validation, or test.

ImageNet-9: ImageNet-9 (Xiao et al., 2020) is a subset of ImageNet (Deng et al., 2009) containing 9 super classes. It has been established that this subset has a spurious correlation between object labels and image texture. We followed the setting used by (Kim et al., 2022) and (Bahng et al., 2020) for creating train and val split. We report the average accuracy on the validation split.

ImageNet-A: ImageNet-A (Hendrycks et al., 2021) contains handpicked real-world images misclassified by models trained on ImageNet. Since these misclassifications are due to over-reliance on spurious features like color&texture, we use this dataset for evaluating models trained on ImageNet-9 as a robustness challenge (i.e., OOD test set).

4.3. Training Procedure & Metrics

For all our real-world experiments we consistently used ResNet-18, an auxiliary layer that uses the same layer structure as of BasicBlock of ResNet with varying depth, optimized using SGD optimizer with a fixed learning rate of 0.001. For real-world experiments the model is loaded with ImageNet pre-trained weights. We repeat the experiments with 5 different random seeds and report the mean and std deviation of results. Refer Appendix A.1 for more details.

Choice of Validation Set: For BAR, since there is no validation data provided, we study it under two settings. In the first, we use 20% of images from the test set and call it OOD-validation. In the second setting, which is harder and more realistic, we use 20% images from the train set, calling it In-Domain (ID) validation. For NICO-Animal, CelebA Hair and ImageNet-9, we use the already supplied validation data. Table 1 shows the percentage of "bias-conflicting" examples, i.e., examples that violate the spurious feature correlation or training domain, for each portion of each dataset. Note that BAR-ID val setting, NICO and ImageNet-9/ImageNet-A experiments do not have any bias conflicting examples in the train set, and methods that rely on attribute labels and/or reweighting of training data will perform poorly on them.

Evaluation Metrics: `Accuracy` means average accuracy on all examples. `Unbiased` means accuracy averaged over each label-context group. This metric is more fair when there is a huge imbalance between the groups. `Conflicting` means accuracy only on the bias conflicting examples.

Table 1. Composition of conflicting examples in different datasets.

Dataset	% Conflict Examples			hparam goal
	Train	Val	Test	
BAR-ID val	0	0	100	Avg Acc
BAR-OOD val	0	100	100	Avg Acc
CelebA	0.8	0.9	0.9	Unbiased Acc
NICO	0	10	10	Avg Acc
IN-9/IN-A	0	0	100	Avg Acc

We used `Accuracy` for BAR, NICO and Imagenet-9 / ImageNet-A and `Unbiased` for CelebA-Hair dataset as the performance metric on validation data for hyperparameter search and early stopping (Table 1).

Feature decodability: To measure the "decodability" of a chosen feature in a classifier at a given layer, we freeze the classifier and train a linear decoder on its network representation at the specified layer. The decoder is trained on validation data, with each instance being assigned as label the value of its feature. For example, to check decodability of shape in a CMNIST classifier, input instances are assigned the label of the shape they contain, while ignoring their color. This decoder's accuracy is then reported on the test set.

5. Results

5.1. Suppressing Simple Features

We first studied the effectiveness of SIFER on targeted suppression of specific features. To do this, we experimented with the CIFAR_MNIST dataset, which consists of composite pairings of MNIST & CIFAR images, each fully predictive of the assigned label (see Section 4 for more details). DNNs are known to entirely ignore the CIFAR feature on this training dataset–when the CIFAR component is randomized at test-time, accuracy is unaffected, but when the MNIST component is randomized, accuracy drops to chance. We refer to the MNIST component as the simple feature, and CIFAR as the complex feature.

Figure 3 shows layerwise decodability (Section 4.3) of simple and complex features, tracked across epochs in the training process. We contrast standard training (top row) against SIFER (bottom row). Standard training overemphasizes the simple feature at higher layers, and ignores the complex

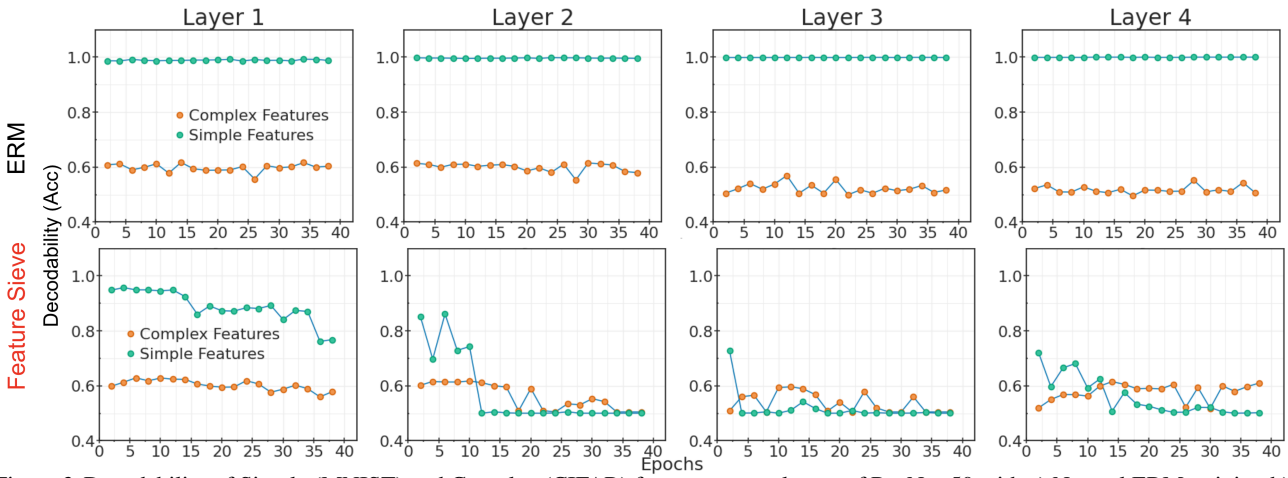

Figure 3. Decodability of Simple (MNIST) and Complex (CIFAR) features across layers of ResNet-50 with a) Normal ERM training b) with SIFER

Table 2. Feature Controllability.

DataSet	Target Feature	SIFER (Ours)		ERM	
		SR	CR	SR	CR
CMNIST	Complex (Digit)	99.54±0.19	58.14±10.69	56.96±6.59	92.21±3.92
	Simple (Color)	52.44±1.22	99.64±1.30	49.20±2.60	96.27±0.99
CIFAR_MNIST	Complex (CIFAR)	62.37±4.62	48.93±1.92	58.14±1.60	100
	Simple (Digit)	47.17±0.14	99.83±0.29	49.20±2.60	100

feature. The complex CIFAR feature is in fact decodable to some extent in earlier layers of the ERM classifier, but is suppressed in later layers, due to the preponderance of the simple feature. In contrast, the auxiliary forgetting loss in SIFER effectively *suppresses the simple feature* in the earlier layers, and thereby enhances the decodability of the complex feature in the higher layers. This shows that removing the availability of spurious simple features is a direct method of overcoming simplicity bias.

These findings are more remarkable given that no prior knowledge of "simple"/"complex" features was used in any way whatsoever during training. SIFER organically discovers and suppresses the by-design simple feature purely through the use of the strategically placed auxiliary network, and its configuration via the training recipe.

5.2. Feature Controllability using SIFER

We described in Section 3.3 the various degrees of freedom in SIFER for identifying and suppressing features, and choosing them by the use of a validation set (i.e., using generalization error). This gives us a simple, powerful method for certain kinds of domain generalization, by simply using domain-shift data as the validation set. We demonstrate this capability by conducting studies on controlled datasets CMNIST & CIFAR_MNIST. In each, the training data were designed to have pairings of simple and complex features where both simple and complex features were fully predictive. We then trained a SIFER classifier for *different choices of validation set*, representing which feature we actually wanted the classifier to focus on. This was achieved by randomizing the "non-relevant" feature in the validation dataset, and choosing all our hyperparameters based on that validation dataset. This represents a real-world scenario where small amounts of vetted data are available for optimization of a model, but (re)-labeling or manipulating all training data is infeasible. Table 2 shows the results comparing SIFER against ERM. SIFER shows higher accuracy for chosen features (higher diagonal terms) than for spurious features (off-diagonal terms), driven by the choice of the validation data. In contrast, ERM primarily focuses on the simple features, irrespective of the choice of validation set (higher second column numbers). Thus, our method is in fact able to focus on the relevant feature–be it simple or complex–in an easily controllable manner.

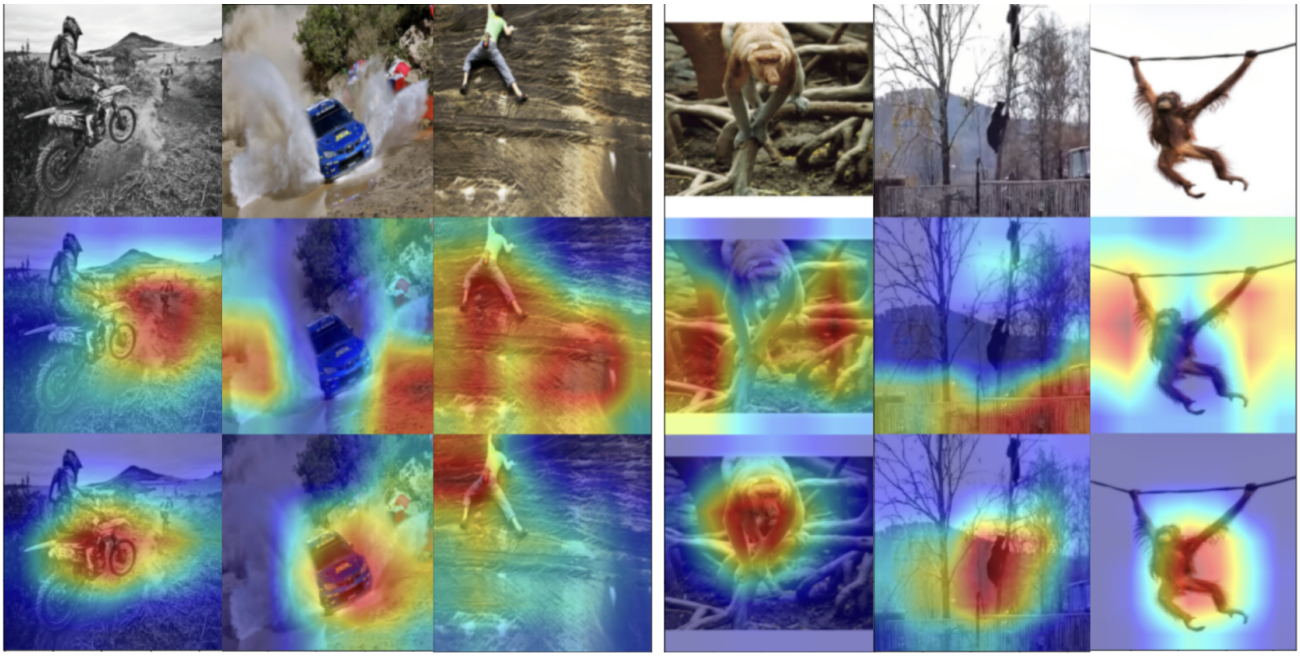

Figure 4. Examples of SIFER's focus on *relevant features* while suppressing irrelevant background information. Top row: Input images; Middle row: GRAD-CAM-derived feature importance visualizations for the ERM classifier; Bottom row: feature importance for SIFER. First 3 columns from BAR (Nam et al., 2020), and last 3 columns from NICO-Animal (He et al., 2021).

5.3. Debiasing Real-World Datasets

Our method outperforms baselines on four different real world datasets–BAR (Nam et al., 2020), CelebA Hair (Liu et al., 2018), NICO (He et al., 2021) and Imagenet-A (Hendrycks et al., 2021), by large margins (upto 11%, see Figure 2c for a quick summary). Critically, we chose in all our experiments to *not use any knowledge* of which attribute labels are considered spurious in each dataset–this is because in real-world scenarios, it is difficult to know in advance which attributes may end up containing biased information, or to label data according to those attributes in order to do targeted debiasing of models. Nevertheless, we outperform the other baselines, including many that do use attribute labels as part of their training procedure.

Mitigating Spurious Correlations: Biased Activity Recognition (BAR) and CelebA Hair Dataset represent background and gender bias in real life. In the BAR training set, human activity (image categories) is spuriously correlated with the background in which those activities are performed; in CelebA Hair, hair color is strongly correlated with gender. Both BAR and CelebA are heavily biased and contain no or very few conflicting examples–eg. CelebA Hair has only 1% of men with blond hair in the train set.

For BAR, since no validation set is provided, we show results both using in-distribution and out-distribution validation sets to compare against both sets of baselines: those that do and do not require conflicting examples in the validation set[4]. We outperform baselines in both settings by more than 1-2% absolute accuracy, refer Table 3. Table 4 shows results on CelebA dataset, we get almost the same unbiased accuracy as LWBC and improve upon conflicting accuracy.

Domain-shift Generalization: NICO introduces three new contexts in which object classes appear in the validation and test set, that are absent in the training set. Table 5 shows that SIFER beats all baselines on classification accuracy for the test set; this is despite our not using context information, unlike a majority of the baselines. Thus, SIFER is valuable for zero-shot domain generalization.

Robustness to Texture bias: Table 6 shows results on ImageNet-9, which is known to be biased towards texture, and ImageNet-A, which consists of natural images that have bias-conflicting features. This setting is closest to real-world scenarios for texture bias. We improve the previous best baseline by 3% absolute on ImageNet-9 validation set and by 4% absolute on ImageNet-A test set. Thus, SIFER en-

[2]ESB uses R50 architecture unlike other baseline which uses R18.

[3]Architecture design optimization based method, hence unfair to compare directly against other methods.

[4]For instance, methods that work on the principle of reweighting conflicting samples in the trainset (eg LWBC (Kim et al., 2022)) typically add 1% of conflicting samples from the test set to the training data

Table 3. Classification Accuracy (%) on test set of BAR Dataset.

Method	Used OOD Val	Accuracy
ERM	✓	51.85± 5.92
BiaSwap (Kim et al., 2021)	✓	52.44
LfF (Nam et al., 2020)	✓	62.98± 2.76
PGI (Ahmed et al., 2021)	✓	65.19± 1.32
EIIL (Creager et al., 2021)	✓	65.44± 1.17
ESB^2 (Teney et al., 2022)	✓	67.10± 0.30
Roadblock (Niu et al., 2022)	✓	69.51± 2.43
Debian (Li et al., 2022)	✓	69.88± 2.92
SIFER (Ours)	✓	**72.08± 0.38**
ERM	✗	35.32± 0.46
ReBias (Bahng et al., 2020)	✗	37.02± 0.26
LfF (Nam et al., 2020)	✗	48.15± 0.93
SSL+ERM (Kim et al., 2022)	✗	60.88± 0.80
LWBC (Kim et al., 2022)	✗	62.03± 0.74
ESB (Teney et al., 2022)	✗	64.40± 0.20
SIFER (Ours)	✗	**65.75± 1.84**

Table 4. Unbiased and Conflicting Accuracy metrics (%) on Test set of CelebA Hair Dataset

Method	Spurious Attribs	Unbiased	Conflict
DRO (Sagawa et al., 2019)	✓	85.43± 0.53	83.40± 0.67
EnD (Tartaglione et al., 2021)	✓	91.21± 0.22	87.45± 1.06
CSAD (Zhu et al., 2021)	✓	89.36	87.53
ERM	✗	70.25± 0.35	52.52± 0.19
LfF (Nam et al., 2020)	✗	84.24± 0.37	81.24± 1.38
SSL+ERM (Kim et al., 2022)	✗	80.48± 0.91	66.79± 2.20
LWBC (Kim et al., 2022)	✗	88.90± 1.55	87.22± 1.14
SIFER (Ours)	✗	**89.00± 0.92**	**88.04± 1.25**

courages learning features robust to texture bias, improving performance on both the in-distribution validation set as well as bias-conflicting test set. Two critical findings here are a) that SIFER did not sacrifice in-distribution accuracy through the process of sieving simple features, and b) the learned classifier robustly transfers over to a novel test set, where it provides even larger gains.

5.4. SIFER: Debiasing without Extra Information

Table 3 (BAR dataset, bottom half) shows that SIFER with only in-distribution validation data (cf. Table 1) outperforms most baselines that leverage an additional OOD validation set (top half). Further, without using either attribute knowledge or conflicting examples in the validation set, we show

Table 5. Classification Accuracy (%) on test set of NICO Dataset. Most of the baselines (DecAug, DRO, etc) use spurious attribute labels for training, while we do not.

Method	Accuracy
ERM	75.87
IRM (Arjovsky et al., 2019)	59.17
REx (Krueger et al., 2021)	74.31
JiGen (Carlucci et al., 2019)	84.95
Mixup (Zhang et al., 2017)	80.27
Cumix (Mancini et al., 2020)	76.78
MTL (Blanchard et al., 2021)	78.89
DANN (Ganin et al., 2016)	75.59
CORAL (Sun & Saenko, 2016)	80.27
MMD (Li et al., 2018)	70.91
DRO (Sagawa et al., 2019)	77.61
CNBB (He et al., 2021)	78.16
DecAug (Bai et al., 2021a)	85.23
SIFER (Ours)	**86.20± 0.85**
NAS-OoD^3 (Bai et al., 2021b)	88.72

huge gains over ERM (65.7% accuracy vs 35.3%), demonstrating that SIFER *does not critically depend on* such additional information for debiasing, although we can certainly leverage such information for additional gains (72.8% vs 65.7% accuracy when using OOD validation data).

5.5. Feature Decodability in Real-world Datasets

Figure 5 shows feature decodability (Section 4.3) on the real world dataset CelebA, where the target label is hair color, and previous work has shown that gender is a spuriously correlated attribute. Results show that SIFER suppresses gender decodability, particularly in upper layers, with color feature achieving stronger decodability (unlike ERM where gender is more easily decodable than color). This mirrors the results on synthetic datasets that were presented in Figure 3, and shows that SIFER can *automatically identify and suppress* featural information related to abstract concepts such as gender, in support of better generalization accuracy.

5.6. SIFER focuses on Relevant Information

We visualize the information in an image that is relevant to a given classifier (Selvaraju et al., 2017), in order to verify whether our feature sieving results in semantically relevant modifications to learned classifiers. Figure 4 shows this evaluation, contrasting ERM classifier's regions of focus (middle row) and SIFER's regions of focus (bottom row) on a range of input images (top row, drawn from BAR & NICO). Interestingly, not only does SIFER correctly focus

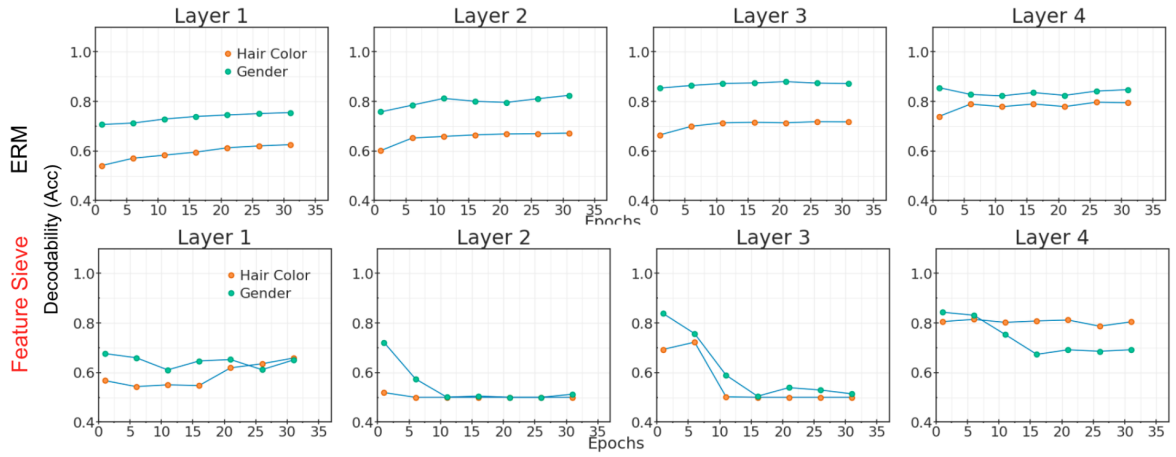

Figure 5. Decodability of Spurious (Gender) and Target (Hair Color) features across layers of ResNet-18 while training on CelebA with a) Normal ERM training and b) with SIFER.

Table 6. Classification Accuracy (%) on Validation set of ImageNet-9 and test set of ImageNet-A.

Method	Spurious Attribs	ImageNet-9	ImageNet-A
		Accuracy	Accuracy
StylisedIN (Geirhos et al., 2018)	✓	88.4± 0.5	24.6± 1.4
LearnedMixin (Clark et al., 2019)	✓	64.1± 4.0	15.0± 1.6
RUBi (Cadene et al., 2019)	✓	90.5± 0.3	27.7± 2.1
ERM	✗	90.8± 0.6	24.9± 1.1
BagNet18 (Brendel & Bethge, 2019)	✗	67.7± 0.3	18.8± 1.15
ReBias (Bahng et al., 2020)	✗	91.9± 1.7	29.6± 1.6
LfF (Nam et al., 2020)	✗	86.00	24.60
CaaM (Wang et al., 2021)	✗	95.70	32.80
SSL+ERM (Kim et al., 2022)	✗	94.18± 0.07	34.21± 0.49
LWBC (Kim et al., 2022)	✗	94.03± 0.23	35.97± 0.49
SIFER	✗	**97.78± 0.12**	**39.98± 0.81**

on the central object of interest, but also it is able to effectively suppress the (spuriously label-correlated) background information, which is highly valued by the ERM classifier. This undercores SIFER's ability to carefully differentiate between relevant and irrelevant features, rather than some notion of simple vs complex features alone.

6. Discussion & Conclusion

We proposed SIFER–a novel *feature sieve* approach towards addressing simplicity bias and spurious correlations in deep neural networks. Our proposal introduces an auxiliary network attached to the deep network which alternately identifies and suppresses predictive features. The approach is controllable through the use of configuration parameters optimized using validation data; thus, it requires no foreknowledge or hand-coding of the notion of "simple features". We demonstrated on controlled datasets the ability of SIFER to automatically identify and suppress features; further, we showed that, strictly speaking, SIFER *rebalances* the role of various features in a controllable manner driven by the needs of generalization. We showed using extensive experiments on real-world data that our approach provides significant gains–3-11% relative accuracy improvements on BAR, NICO, and Imagenet-A. We believe our work is a small, important first step in a fruitful new direction of research. We hope that follow-up work will build on the notion of the feature sieve, developing effective computational barriers that encourage deep networks to discover and utilize richer, more powerful featural representations. Our current approach strikes a balance between various competing features, guided by generalization error estimates (validation error). One could potentially extract even more value if different feature classes could be isolated into (relatively) independent predictors, then combined effectively. This is, for instance, the approach taken by Niu et al. (2022). Thus, a straightforward next step we aim to explore is the study of ensembling approaches to combine a range of features of varying complexity & predictive power, and methods for efficiently learning them. We also hope to develop a systematic theoretical understanding of feature sieve approaches and their role in supervised learning using DNNs.

References

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In International Conference on Learning Representations, 2021.

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

Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. Learning de-biased representations with biased representations. In International Conference on Machine Learning, pp. 528–539. PMLR, 2020.

Bai, H., Sun, R., Hong, L., Zhou, F., Ye, N., Ye, H.-J., Chan, S.-H. G., and Li, Z. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. AAAI, 2021a.

Bai, H., Zhou, F., Hong, L., Ye, N., Chan, S.-H. G., and Li, Z. Nas-ood: Neural architecture search for out-of-distribution generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8320–8329, 2021b.

Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G., and Scott, C. Domain generalization by marginal transfer learning. The Journal of Machine Learning Research, 22 (1):46–100, 2021.

Brendel, W. and Bethge, M. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. arXiv preprint arXiv:1904.00760, 2019.

Cadene, R., Dancette, C., Cord, M., Parikh, D., et al. Rubi: Reducing unimodal biases for visual question answering. Advances in neural information processing systems, 32, 2019.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain generalization by solving jigsaw puzzles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2229–2238, 2019.

Clark, C., Yatskar, M., and Zettlemoyer, L. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. arXiv preprint arXiv:1909.03683, 2019.

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In International Conference on Machine Learning, pp. 2189–2200. PMLR, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

Duboudin, T., Dellandréa, E., Abgrall, C., Hénaff, G., and Chen, L. Look beyond bias with entropic adversarial data augmentation. In 2022 26th International Conference on Pattern Recognition (ICPR), pp. 2142–2148. IEEE, 2022.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In Proceedings of the 3rd innovations in theoretical computer science conference, pp. 214–226, 2012.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. The journal of machine learning research, 17(1):2096–2030, 2016.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In International Conference on Learning Representations, 2018.

He, Y., Shen, Z., and Cui, P. Towards non-iid image classification: A dataset and baselines. Pattern Recognition, 110:107383, 2021.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. CVPR, 2021.

Hermann, K. and Lampinen, A. What shapes feature representations? exploring datasets, architectures, and training. Advances in Neural Information Processing Systems, 33: 9995–10006, 2020.

Kim, B., Kim, H., Kim, K., Kim, S., and Kim, J. Learning not to learn: Training deep neural networks with biased data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9012–9020, 2019.

Kim, E., Lee, J., and Choo, J. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14992–15001, 2021.

Kim, N., Hwang, S., Ahn, S., Park, J., and Kwak, S. Learning debiased classifier with biased committee. arXiv preprint arXiv:2206.10843, 2022.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In International Conference on Machine Learning, pp. 5815–5826. PMLR, 2021.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5400–5409, 2018.

Li, Y. and Vasconcelos, N. Repair: Removing representation bias by dataset resampling. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9572–9581, 2019.

Li, Z., Hoogs, A., and Xu, C. Discover and mitigate unknown biases with debiasing alternate networks. In 17th European Conference on Computer Vision, pp. 270–288. Springer, 2022.

Liu, Z., Luo, P., Wang, X., and Tang, X. Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15 (2018):11, 2018.

Mancini, M., Akata, Z., Ricci, E., and Caputo, B. Towards recognizing unseen categories in unseen domains. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16, pp. 466–483. Springer, 2020.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), 54(6):1–35, 2021.

Nagarajan, V., Andreassen, A., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. arXiv preprint arXiv:2010.15775, 2020.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: Training debiased classifier from biased classifier. In Advances in Neural Information Processing Systems, 2020.

Niu, H., Li, H., Zhao, F., and Li, B. Roadblocks for temporarily disabling shortcuts and learning new knowledge. In Advances in Neural Information Processing Systems, 2022.

Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. Advances in Neural Information Processing Systems, 34:1256–1272, 2021.

Russell, C., Kusner, M. J., Loftus, J., and Silva, R. When worlds collide: integrating different counterfactual assumptions in fairness. Advances in neural information processing systems, 30, 2017.

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

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pp. 618–626, 2017.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. Advances in Neural Information Processing Systems, 33: 9573–9585, 2020.

Shrestha, R., Kafle, K., and Kanan, C. Occamnets: Mitigating dataset bias by favoring simpler hypotheses. arXiv preprint arXiv:2204.02426, 2022.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14, pp. 443–450. Springer, 2016.

Tartaglione, E., Barbano, C. A., and Grangetto, M. End: Entangling and disentangling deep representations for bias correction. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13508–13517, 2021.

Teney, D., Abbasnejad, E., and Hengel, A. v. d. Unshuffling data for improved generalization. arXiv preprint arXiv:2002.11894, 2020.

Teney, D., Abbasnejad, E., Lucey, S., and van den Hengel, A. 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, pp. 16761–16772, 2022.

Wang, T., Zhou, C., Sun, Q., and Zhang, H. Causal attention for unbiased visual recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3091–3100, 2021.

Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition. ArXiv preprint arXiv:2006.09994, 2020.

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Proceedings of the 26th international conference on world wide web, pp. 1171–1180, 2017.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.

Zhou, X., Lin, Y., Zhang, W., and Zhang, T. Sparse invariant risk minimization. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 27222–27244. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhou22e.html.

Zhu, W., Zheng, H., Liao, H., Li, W., and Luo, J. Learning bias-invariant representation by cross-sample mutual information minimization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15002–15012, 2021.

Appendix

A. Additional Details

A.1. Training Details

For all experiments we consistently used ResNet-18, an auxiliary layer that uses the same layer structure as of BasicBlock of ResNet with varying depth. The ResNet network is composed of 4 layers modules (each itself is made up of 2 BasicBlocks). We apply auxiliary layer only at the end of the of layers except layer 4, this gives us 3 different choice for aux position($\mathcal{A}_P$) which we treat as hyperparameter. The network is optimized using SGD optimizer with a fixed learning rate of 0.001. For real-world experiments the model is loaded with ImageNet pre-trained weights. We repeat the experiments with 5 different random seeds and report the mean and std deviation of results. Table 7 shows the hyperparamters search space for all the hyperparameters that we tune on the basis of validation set. To reduce hyperparameter search space we fixed the value of $\alpha 1$ to 10. Table 8 shows the hyperparameter values obtained from the hyperparamter tuning.

Table 7. Range for hyperparameters search.

Hparam	**Range**
$\mathcal{A}_D$	$[1, 9]$
$\mathcal{A}_P$	$[1, 3]$
$\alpha 2$	$\text{loguniform}(10^{-1}, 10^2)$
$\alpha 3$	$\text{loguniform}(10^{-1}, 10^2)$
$\mathcal{F}$	$[1, 9] * 10$

Table 8. Hyperparameter values obtained from the tuning.

Dataset	$\mathcal{A}_D$	$\mathcal{A}_P$	$\alpha 2$	$\alpha 3$	$\mathcal{F}$
BAR - ID val	4	2	2	4.5	70
BAR - OD val	2	2	1	3	30
CelebA	2	2	25	15	50
NICO	2	1	1	75	70
IN-9/IN-A	4	3	1	4.5	70

B. Baselines

Here we list and briefly explain all the baselines that we compare against on real world datasets:
BiaSwap (Kim et al., 2021) proposes a bias-tailored augmentation-based approach for learning debiased representation without requiring supervision on the bias type. they divide the data into bias-guiding and bias-conflicting groups and then swaps the bias in bias guiding group.
LfF (Nam et al., 2020) uses generalized cross-entropy initially trains a prejudiced net-work and tries to debias the second network by focusing weighing on samples that go against the bias.
IRM (Arjovsky et al., 2019) uses theory of causal bayesian networks to find an invariant feature representation using multiple training environments with different bias correlations.
REx (Krueger et al., 2021) proposed a min-max algorithm to optimize for the worst linear combination of risks on different environments.
EIIL (Creager et al., 2021) optimizes for bias group assignment to automatically identify the bias groups to maximize IRM.
PGI (Ahmed et al., 2021) follows EIIL to identify bias groups by training a small neural network.
Evading Simplicity Bias (ESB) (Teney et al., 2022) creates a ensemble of diverse classifiers by incorporating a diversity regularizer between the gradients while training.
Roadblock (Niu et al., 2022) adds adversarial augmentations to the image while training to avoid over-reliance on spurious visual cues.
Debian (Li et al., 2022) trains two networks in alternate manner namely discoverer and classifier, the discoverer tries to find

multiple unknown biases of the classifier without any annotations of biases, and the classifier aims at unlearning the biases identified by the discoverer.

ReBias (Bahng et al., 2020) propose a novel framework to train a de-biased representation by encouraging it to be different from a set of representations that are biased by design.

LWBC (Kim et al., 2022) employs a committee of classifiers as an auxiliary module that identifies bias-conflicting data and assigns large weights to them when training the main classifier.

Group-DRO (Sagawa et al., 2019) minimizes for worst-case training loss over a set of pre-defined groups.

EnD (Tartaglione et al., 2021) proposes a regularization technique that uses the bias attributes to prevent deep models from learning spurious biases by inserting an information bottleneck.

CSAD (Zhu et al., 2021), given the bias attributes, explicitly extracts target and bias features disentangled from the latent representation generated by a feature extractor and then learns to discover and remove the correlation between the target and bias features.

JiGen (Carlucci et al., 2019) jointly classifies objects and solves unsupervised jigsaw tasks.

Cumix (Mancini et al., 2020) mixes up data and labels from different domains to be able to recognize unseen categories in unseen domains.

MTL (Blanchard et al., 2021) argue that problem of Domain Generalization can be viewed as a kind of supervised learning problem by augmenting the original feature space with the marginal distribution of feature vectors.

DANN (Ganin et al., 2016) proposes a representation learning approach such that features are not predictive of the domain from which the model is being trained on.

CORAL (Sun & Saenko, 2016) proposes an unsupervised domain adaptation method that aligns the second-order statistics of the source and target distributions with a linear transformation.

MMD (Li et al., 2018) extend adversarial autoencoders by imposing the Maximum Mean Discrepancy measure to align the distributions among different domains, and matching the aligned distribution to an arbitrary prior distribution via adversarial feature learning.

CNBB (He et al., 2021) is an OoD learning method that based on sample reweighting inspired by causal inference.

DecAug (Bai et al., 2021a) proposed a semantic augmentation and feature decomposition approach to distangle context features from category related features.

NAS-OoD (Bai et al., 2021b) adds an OOD generalization criterion to network architecture search training to construct inherently more robust network architectures.

StylisedIN (Geirhos et al., 2018) showed that ImageNet is texture biased and works on improving shape bias.

LearnedMixin (Clark et al., 2019) trains a robust model as part of an ensemble with the naive one in order to encourage it to focus on other patterns in the data that are more likely to generalize.

CaaM (Wang et al., 2021) learns causal attention by partitioning the data on-the-go to break correlation with bias.