InBiaseD: Inductive Bias Distillation to Improve Generalization and Robustness through Shape-awareness

Shruthi Gowda, Bahram Zonooz,[*] Elahe Arani[*]
Advanced Research Lab, NavInfo Europe, Eindhoven, The Netherlands
{shruthi.gowda,elahe.arani}@navinfo.eu, bahram.zonooz@gmail.com

Abstract

Humans rely less on spurious correlations and trivial cues, such as texture, compared to deep neural networks which lead to better generalization and robustness. It can be attributed to the prior knowledge or the high-level cognitive inductive bias present in the brain. Therefore, introducing meaningful inductive bias to neural networks can help learn more generic and high-level representations and alleviate some of the shortcomings. We propose InBiaseD to distill inductive bias and bring shape-awareness to the neural networks. Our method includes a bias alignment objective that enforces the networks to learn more generic representations that are less vulnerable to unintended cues in the data which results in improved generalization performance. InBiaseD is less susceptible to shortcut learning and also exhibits lower texture bias. The better representations also aid in improving robustness to adversarial attacks and we hence plugin InBiaseD seamlessly into the existing adversarial training schemes to show a better trade-off between generalization and robustness.[1]

1 Introduction

Deep neural networks (DNNs), in particular, the convolutional neural networks, which are loosely inspired by the primate visual system, are achieving superior performance in a multitude of perception tasks. The goal of DNNs is to learn higher-level abstractions and strike a balance between encompassing as much information as possible from the input data and maintaining invariance to the variations present in the distribution (Bengio, 2012). Furthermore, DNNs have been proposed as the computational models of human perception (Kubilius et al., 2016), where the networks are hypothesized to capture perceptually salient shape features similar to humans. However, studies have shown that networks are highly reliant on local textural information instead of the global shape semantics (Brendel & Bethge, 2019). Hence, there are still fundamental questions about the underlying working of the networks such as, What is the network exactly learning? Are they indeed capturing high-level abstractions?

The psychophysical experiment conducted by Geirhos et al. (2019) demonstrated the difference in biases existing between networks and humans. While the networks relied heavily on textures, humans relied more on the shape features. This indicates the tendency of the networks to focus on the trivial local cues, thus taking a "shortcut". As Occam's razor would theorize, 'why learn generic global features when the local attributes suffice?' This shortcut behavior of networks leads to lower generalization, as the shortcuts that exist in the current data, do not transfer to data from a different distribution. Jo & Bengio (2017) also quantitatively showed the tendency of the networks to learn the surface statistical irregularities present in the data, instead of the task relevant attributes. This fickle nature of networks to latch onto the spurious correlations and unintended cues present in the dataset instead of the high-level abstractions and task-relevant features weakens the generalization and robustness of neural networks.

The ability of humans to be less vulnerable to shortcut learning and to identify objects irrespective of textural or adversarial perturbations can be attributed to the high-level cognitive inductive bias in the brain. This prior-knowledge or pre-stored templates exist even at an earlier age, like in a child's brain (Pearl & Mackenzie, 2018). Although deep learning already has inductive bias inherent in its design, such as distributed representations and group equivariance (via convolutions), there is still scope to introduce meaningful biases that will enable the networks to achieve the original representation goal of learning high-level and meaningful representations while being invariant to the unintended cues in the training data. We focus on *"shape"* as one of the meaningful inductive biases as it is also observed that humans focus more on global shape semantics to make decisions (Geirhos et al., 2019; Ritter et al., 2017). We, therefore, strive to introduce an inductive bias to bring in more shape-awareness into neural networks.

[*] Shared last author. [1] We open source our code at https://github.com/NeurAI-Lab/InBiaseD.

To help the neural network learn more generic features, several works propose creating new samples to augment the training dataset (Geirhos et al., 2019; Li et al., 2020; Zhou et al., 2021; Chen et al., 2016). But creating new data comes with a cost and maybe with an unaccounted bias and moreover, the existing data already has under-utilized valuable information that can be exploited with minimal overhead. Thus, we forego any additional requirements in terms of data or meta information and propose *"InBiaseD" (Inductive Bias Distillation)*, to distill inductive bias into the network by utilizing the shape information already existing in the data. InBiaseD constitutes a setup of two networks, one receiving the original images and the other accessing the corresponding shape information. Along with the supervised learning for each network, a bias alignment objective is formulated to align both these modalities. The bias alignment is performed at latent space and the decision space and acts as a regularizer to reduce over-fitting to trivial solutions. Feature alignment in latent space forces the network to learn representations invariant to trivial attributes and be more generic. The decision boundary alignment incentivizes the supervision from shape to help make decisions that are less susceptible to shortcut cues. Thus, our method encourages the network to also focus on global shape semantics to encode more generic high-level abstractions.

We perform an extensive analysis to show the efficacy of our method. InBiaseD improves generalization performance over standard training on multiple different datasets of varying complexity. The generalization improvement also translates to out-of-distribution (OOD) data, thus displaying higher robustness to distribution shifts. We also conduct shortcut learning analysis and the results indicate that inducing shape-awareness into the networks makes them less vulnerable to spurious correlations and statistical irregularities that exist in the training data. The texture-bias analysis shows that our method is less prone to rely just on the local texture data. InBiaseD also makes the networks robust against adversarial perturbations thus further indicating that the network is learning more high-level representations. Real-world applications also need the networks to be reliable along with being accurate. To this end, we present a calibration analysis where we observe that InBiaseD shows better calibration and leans towards being cautious and prudent in contrast to the over-confident predictions by the standard training. Finally, we plugin InBiaseD into the standard adversarial training schemes to test the trade-off between generalization and robustness. Adding inductive bias proves beneficial and we report improved performance in both natural and adversarial accuracy. All our results highlight that distilling inductive bias and making DNNs more shape-aware has a positive impact and hence presents a compelling case for further exploration in incorporating higher-level cognitive biases. Our contributions are as follows,

- InBiaseD - Inductive Bias Distillation - a method to distill inductive bias into the networks and making them more shape-aware.
- Analysis to show the reduced susceptibility of InBiaseD to shortcut learning and texture bias.
- Seamless integration of InBiaseD to existing adversarial training schemes to measure the benefit of inductive bias in the trade-off between generalization and robustness.
- Extensive analysis on multiple datasets of varying complexity: identically and independently distributed (IID) and out-of-distribution (OOD) generalizations.

2 Related Work

Style transfer/augmentation based approaches: To improve generalization and reduce shortcut learning, multiple works have been proposed. To enable the network to learn more shape-biased representations, Geirhos et al. (2019) introduced a dataset, Stylized-ImageNet, by performing style-transfer of artistic paintings onto the ImageNet data using adaptive instance normalization (AdaIN) (Huang & Belongie, 2017). The results indicate that shape-biased representations may be beneficial for object recognition tasks. Li et al. (2020) also creates an augmented dataset using style transfer but the style image is chosen from the training data itself. The texture and shape information of two randomly chosen images are blended to create new training samples and labels. Zhou et al. (2021) uses AdaIN to mix styles of images present in the training set to create novel domains to improve domain generalization. Continuing with the solution trend of generating augmented images for training, InfoGan (Chen et al., 2016) generates training samples with disentangled features to synthesize de-biased samples. Synthesizing and generating new data is expensive and training a single network with both original and new data distributions leads to learning sub-optimal representations.

Debiasing approaches: Several other works identify the bias existing in the data (such as color, texture, gender) and try to reduce the model's reliance on these biases. Kim et al. (2019) train two networks, one to predict the label and the other to predict the bias, in an adversarial training process. A regularization loss based on mutual information is used to reduce the dependency of networks on biased instances. Other bias-supervision-based approaches try to disentangle bias-attributes from intrinsic-attributes in the data and put emphasis on learning the latter over the former. Lee et al. (2021) employs generalized cross-entropy to train a model to be biased by emphasizing the easier samples. Another network is trained on the relatively difficult samples which are assumed to have more intrinsic attributes. The latent

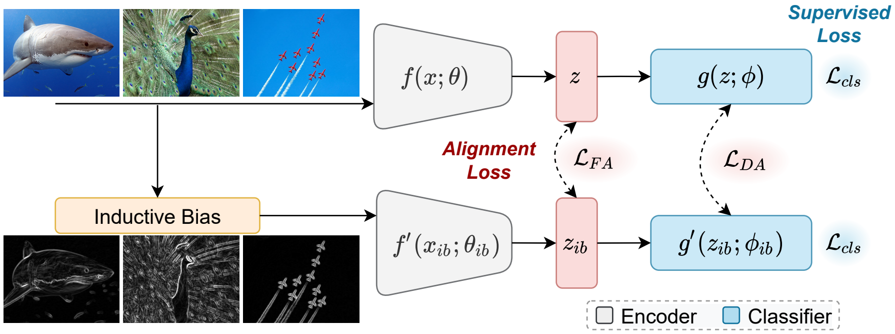

Figure 1: Schematic of our proposed ***Inductive Bias Distillation*** method. The InBiaseD network (f, g) receives the original data and the ShapeNet (f', g') receives the corresponding shape data. The networks are trained with a supervised loss and a bias alignment loss - an aggregation of *feature alignment* $\mathcal{L}_{FA}$ and *decision alignment* $\mathcal{L}_{DA}$ losses, to distill inductive bias into the InBiaseD network.

features are also swapped to allow for feature-level data augmentation. These approaches either require knowing the type of bias in advance or depend on a distinct correlation between bias attributes and intrinsic attributes in the data.

Ensemble approaches: To obtain better generalization across domains, Mancini et al. (2018) built an ensemble of multiple domain-specific classifiers, each attributing to a different cue. Jain et al. (2021) trained multiple networks separately, each with a different kind of bias. The ensemble of these biased networks was used to produce pseudo labels for unlabeled data. Lakshminarayanan et al. (2017) use ensembles of multiple networks trained on the same modality to improve generalization, referred to as DeepEnsemble. However, these approaches are expensive to use as all require an ensemble of networks at inference which doubles the resource cost. Moreover, our goal is to enable the network to learn global semantic information to reduce shortcut learning and texture-bias which differs from the objective of ensemble networks.

3 Methodology

Motivation: Learned representations play a pivotal role as they capture the features in the input distribution that is used by the networks to make the decisions. Improving in-distribution performance is good but understanding what features the network is learning to make these decisions deems crucial in improving the DNNs performance. A standard convolutional neural network tends to encode features encompassing more local information (such as texture and color) in the data. A model that learns to rely on this local solutions might achieve good accuracy on test data but might fail on a data distribution with different color schemes, backgrounds, or perturbations compared to the training data. Human perception, on the other hand, is more robust and is barely affected by color, textural changes, or any small perturbations in the data. The implicit template or cognitive bias allows the brain to look at more discriminative features such as shape for learning. The implicit inductive bias in CNNs can push the network to use better features but the biases are non-specific which allows the network to still stick to local patterns in the data. Inducing an additional prior knowledge or bias can drive the networks towards learning better representations and aid in improved generalization and robustness.

Therefore, we intend to use an implicit prior knowledge already existing in the data to enrich the learned representations. The shape information exists in the data, however with passive observation is not fully utilized by the neural networks. Making this knowledge explicit and biasing the network towards it can help to induce shape-awareness into the neural networks. To this end, we extract the shape information from the data by using edge-detection algorithms and encourage the network to look beyond the local attributes to also learn the global semantic information (see Section A.2.1 in Appendix for the algorithms and sample images).

3.1 Formulation

We propose InBiaseD to distill inductive-bias into the DNNs to encode better representations and enhance the generalization and robustness performance. We extract the shape information and enforce the network to focus on the shape attributes existing in the data. Using a single network to learn both original and shape data will result in sub-optimal representations as there is a distribution shift between them. Instead, we train two networks in synchrony and introduce a bias alignment objective to align the two networks in both representation space and decisions space (Figure 1).

The two networks, InBiaseD and ShapeNet, receive the original data and the shape correspondence, respectively. The bias alignment objective provides the flexibility to the network to learn on its own input but also align with its peer network. The bias alignment aggregates the information of two different spaces: feature-space and prediction-space. InBiaseD learns the original and the shape data in synchrony which helps explore more generic representations and reduce over-fitting to trivial cues in the data.

The input images x and their corresponding labels y are sampled from dataset D. The samples x are sent to the Sobel Algorithm 1 to extract the shape data, x_{ib}. InBiaseD network (with encoder f and classifier g) receives the original images x while ShapeNet (with encoder f' and classifier g') receives shape images x_{ib} as input. The latent representations $z = f(x)$ and $z_{ib} = f'(x_{ib})$ are used by the respective classifiers g and g' to perform the object recognition. Two networks learn in synchrony and inductive bias is instilled through a bias alignment objective which involves aggregating the information in two spaces. The decision alignment (DA) performed in the prediction space, aligns the networks' probability distributions. The decision boundary alignment is incentivized by the supervision from shape data hence, allowing the network to make decisions that are less susceptible to shortcut cues. We employ the Kullback-Leibler divergence as the objective for the DA,

$$\mathcal{L}_{DA} = \mathcal{D}_{KL}(\text{softmax}(g(z))||\text{softmax}(g'(z_{ib}))) \tag{1}$$

The DA happens in the final prediction stage, but we want to ensure the shape supervision is provided even at the earlier stages to ensure the encoding of shape attributes into the feature representations. To this end, we use the feature alignment (FA) to maintain consistency between the features of two networks in the latent space. We employ a more strict alignment using mean squared error (MSE) as the objective for the FA,

$$\mathcal{L}_{FA} = \underset{z \sim f(x), z_{ib} \sim f'(x_{ib})}{\mathbb{E}} \|z - z_{ib}\|_2^2 \tag{2}$$

The overall loss function of InBiaseD and ShapeNet networks are the sum of the classification loss and the two alignment losses:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{DA}(g(z), g'(z_{ib})) + \gamma \mathcal{L}_{FA}(z, z_{ib}) \tag{3}$$

$$\mathcal{L}_{ib} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{DA}(g'(z_{ib}), g(z)) + \gamma \mathcal{L}_{FA}(z_{ib}, z) \tag{4}$$

where λ and γ are the balancing factors. The detailed algorithm is provided in Algorithm 2 in Appendix.

4 Experiments

ResNet-18 architecture (He et al., 2016) is used for both the networks. We perform extensive analyses on multiple datasets of varying complexity, listed in Table 3 in Appendix. The baseline settings are a result of hyperparameter-tuning that get the highest baseline accuracy for each dataset and are provided in Table 4 in Appendix. The learning rate is set to 0.1. SGD optimizer is used with a momentum of 0.9 and a weight decay of $1e{-}4$. The same settings as for baselines are used for training InBiaseD. We apply random crop and random horizontal flip as the augmentations for all training. The same augmentation is applied to both networks, InBiaseD receives augmented inputs on original data while the ShapeNet receives them on the shape data. For extracting shape data, the original samples are up-sampled to twice their size, a Sobel filter is applied to extract the edges and finally, samples are then down-sampled to the original size. Only the InBiaseD network (with encoder f and classifier g in Figure 1) is used for inference purposes. For all the experiments, the mean and standard deviation of three runs with different random seeds are reported.

5 Shortcut Learning

Shortcuts are defined as decision rules that perform well on the current data but that do not transfer to data from a different distribution. DNNs are shown to rely on the spurious correlations or statistical irregularities present in the dataset, thus suffering from shortcut learning (Geirhos et al., 2020; Xiao et al., 2020). To measure the vulnerability of the models to shortcut learning, we perform two types of analyses.

5.1 Spurious Correlation Analysis

Spurious correlations are the unintended associations present in the training data that might not translate to the test settings. We investigate this effect on three different datasets. We create a synthetic dataset, Tinted-STL-10, by adding a class-specific tint to the original STL-10 data following Jain et al. (2021). This tint is only added to the training set (see samples in Figure 9) and not to the test set. The performance drops when tested on the original STL test set (without the tint) which reveals the susceptibility of DNNs to rely on these correlations. As the second dataset,

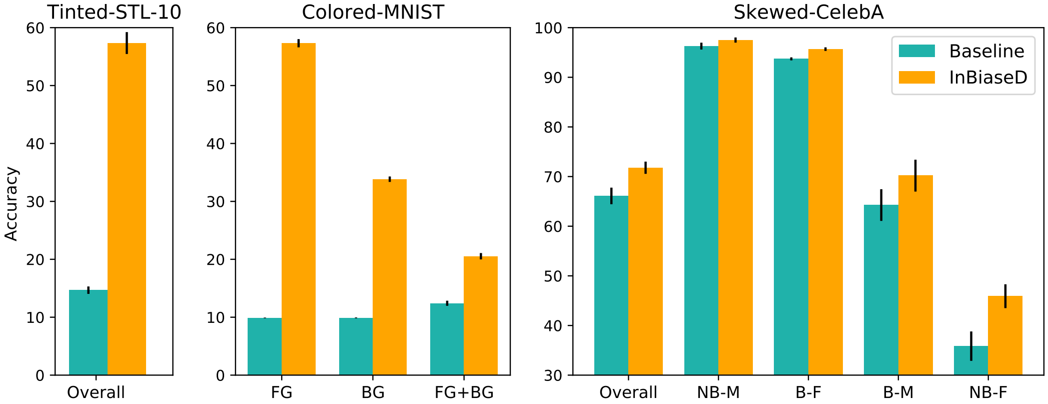

Figure 2: Shortcut learning experiments: models trained on Tinted-STL-10, variants of Colored-MNIST and Skewed-CelebA datasets and tested on the original test sets. The performance improvements indicate that InBiaseD is less vulnerable to spurious correlations added to the training data. See Table 10 in Appendix for numerical comparison against more techniques.

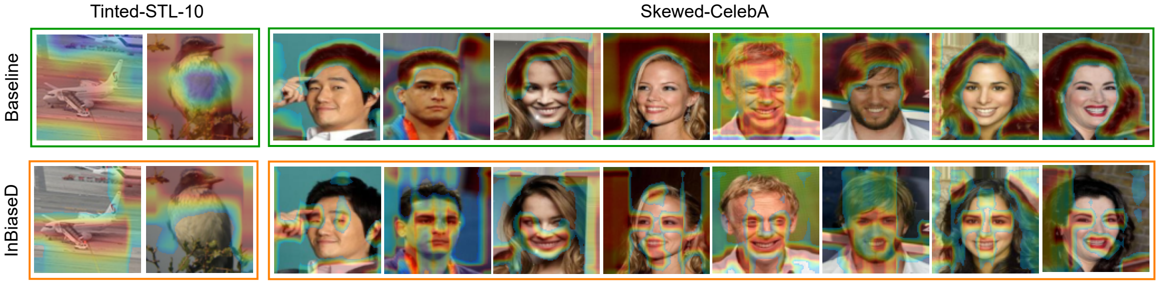

Figure 3: Visualizing the activations maps of the models on the Tinted-STL-10 and Skewed-CelebA datasets. InBiaseD looks at salient features of the object and face to make predictions while Baseline focuses more on the spurious background and hair color.

we inject spurious colors onto the MNIST dataset (Deng, 2012) to create Colored-MNIST (samples in Figure 10). These colors are added in three different ways to keep a correlation between: (1) digit color and its label (FG), (2) background color and the label (BG), or (3) both foreground-background color combinations and the label (FG+BG). The foreground and background colors of the test data are completely random and there is no spurious correlation between the image and its label. For the third dataset, we consider the gender classification task based on CelebA (Liu et al., 2015). We create a skewed-CelebA dataset that consists of only “blond-females” (B-F) and “non-blond-males” (NB-M) samples and use this as the training set following Jain et al. (2021). The test set contains all four combinations of hair color and gender.

Figure 2 shows that InBiaseD is less vulnerable to the induced correlations. InBiaseD gains a significant improvement on the Tinted-STL-10 and different variants of Colored-MNIST datasets. On the Skewed-CelebA dataset, we observe better generalization to blond-male (B-M) and non-blond-female (NB-F) samples (categories that were not present in the training set). The bias alignment loss supervises the network to look beyond the spurious attributes such as the tint-background, color-digit, and hair-gender correlation and instead focus on the semantic information of the objects to make the final decisions.

To precisely view what attributes the model is relying on to make the decisions, we use GRAD-CAM (Selvaraju et al., 2017) to visualize the attended image regions. We visualize the sensitivity of the final layer of the networks for the STL-10 dataset and the outputs to the activations of the penultimate layer of the networks for Celeb-A dataset. As seen in Figure 3, for STL-10 dataset, InBiaseD focused on the object-relevant features (such as the aircraft wings, the beak, and the head of the bird) while the standard network focuses more on the task-irrelevant background information to classify. On Skewed-CelebA data, the Baseline model relies more on the unintended cue from hair color while our method relies more on the salient facial features of the face (such as eyes and lips). Thus, the bias alignment in our method encourages the InBiaseD network to encode more task-relevant and higher-level semantic information.

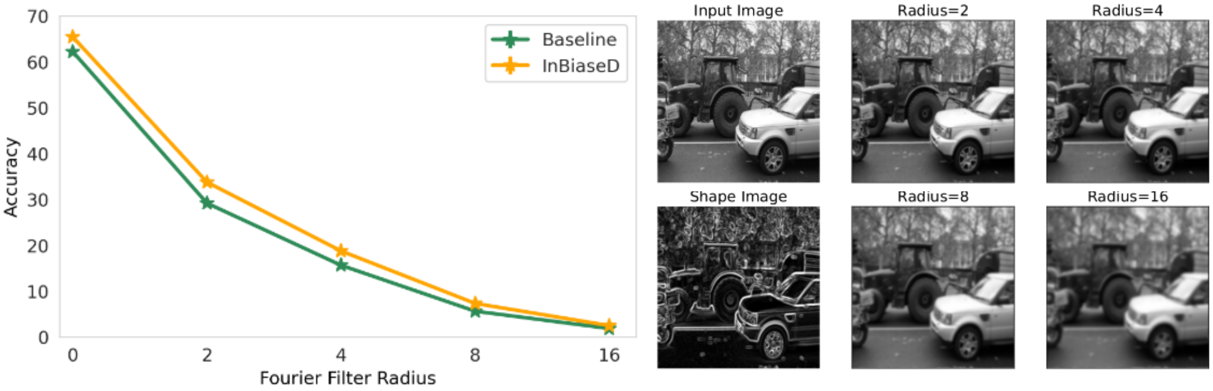

Figure 4: Fourier transform analysis to test the model's dependency on statistical irregularities in the dataset. InBiaseD fares better than the Baseline. Examples of radial low-pass Fourier filtered images are depicted (numerical results in Table 5 in Appendix).

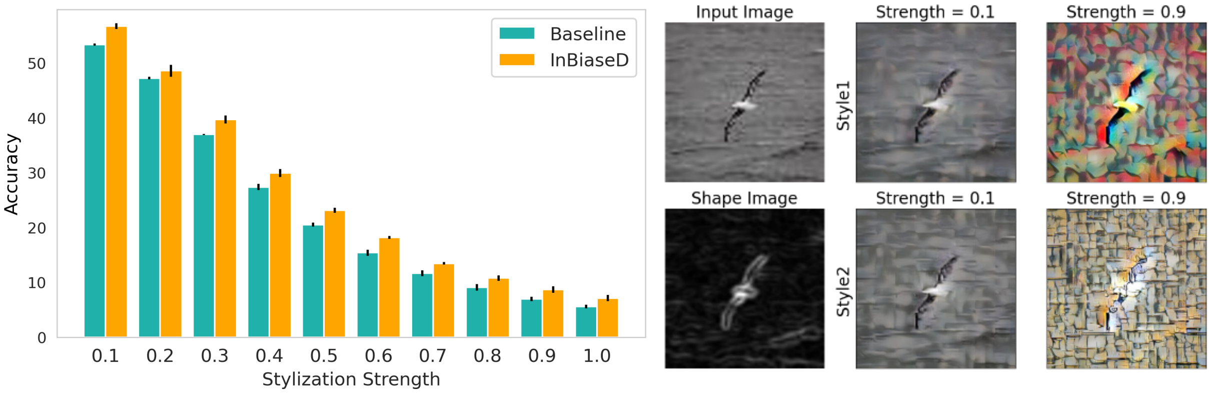

Figure 5: Evaluation of texture bias: models trained on TinyImagenet and tested on stylized images with four different styles and varying strengths. InBiaseD shows better generalization thus exhibiting lesser texture-bias and more shape-awareness (numerical results in Table 6 in Appendix).

5.2 Statistical Irregularity Analysis

Another test to show if models are indeed learning high-level abstractions is to apply a transform that changes the statistics of the data without altering its high-level semantic details. To this end, Jo & Bengio (2017) uses a Fourier transform that changes the statistics of the images while still preserving the recognizability of the objects from a human perspective. This ensures that the original image and the filtered image share the same high-level concepts but differ only in surface statistical cues. The Fourier filtered samples at varying radius strengths are depicted in Figure 4.

To quantitatively measure the tendency of DNNs to learn surface statistical irregularities in the data, we apply a radial low pass Fourier filter and evaluate the trained models on these samples to measure the generalization gap. Models trained on TinyImageNet are evaluated on these transformed images to check the accuracy. As shown in Figure 4, InBiaseD performs better and is less prone to learning the superficial attributes in the data than Baseline. Overall, distilling inductive bias shows promising results in the direction of tackling the challenges of shortcut learning.

6 Texture Bias

DNNs are more biased towards texture while humans rely more on the shape to form decisions (Geirhos et al., 2019). InBiaseD strives to make the model more shape-aware by distilling shape supervision. To test if our method has reduced the model's bias towards texture, we perform a texture bias analysis. We apply style-transfer (Huang & Belongie, 2017) of varying strengths on the TinyImageNet images and evaluate them using a model trained on the

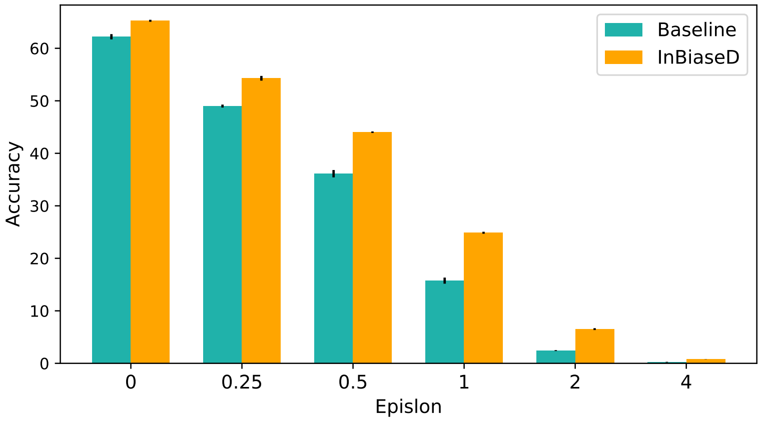

Figure 6: Evaluating PGD-10 adversarial attack at varying strengths on models trained on TinyImageNet dataset. InBiaseD is more robust to the perturbations indicating learning of more high-level representations (numerical results in Table 7 in Appendix).

original TinyImageNet dataset. We choose four different styles across the complete strength spectrum $\in [0.1, 1.0]$; see samples in Figure 11. The stylized images have different textures and hence the model needs to have learned more than just local texture cues to infer well on them. As shown in Figure 5, InBiaseD generalizes better on the stylized images compared to the Baseline. The bias alignment objective enables the model to learn more salient shape features instead of completely relying on local textural features. The performance stays significant and consistent at even higher strengths of stylization, thus proving beneficial in challenging cases where the texture is considerably different between training and testing domains. InBiaseD proves a successful step towards making DNNs more shape-aware and overcoming the challenges of texture-bias prevalent in the standard DNNs.

7 Robustness

7.1 Adversarial Attacks

DNNs are shown to be extremely sensitive to the adversarial perturbations (Szegedy et al., 2013), carefully crafted imperceptible noise added to the original data which results in a perceptually similar image to the original one. While humans can still make correct predictions, models are prone to making erroneous predictions that can have disastrous consequences in safety-critical applications. To analyze the adversarial robustness of the models, we perform a Projected Gradient Descent (PGD) attack (Madry et al., 2017) of varying strengths. We perform a PGD-10 attack on the models trained on the TinyImageNet dataset. As observed in Figure 6, InBiaseD shows more resistance to the attacks and has higher robustness across all the attack strengths. Jo & Bengio (2017) hypothesizes that if the models are learning high-level abstractions, then they should not be sensitive to small perturbations in the data. Hence, from this hypothesis and our results, we can infer that InBiaseD training is more likely to be helping in learning high-level representations compared to the standard training. These encouraging results prompt us to further explore the effect of adding inductive bias to the standard adversarial training schemes.

7.2 Adversarial Training

Adversarial training has proven to be an effective defense technique for improving the adversarial robustness of models. Robustness is at odds with the accuracy (Tsipras et al., 2018) and hence this trade-off between the natural accuracy and adversarial robustness is one of the prevalent challenges in adversarial training. Multiple works have been proposed to reduce this gap (Zhang et al., 2019; Arani et al., 2020; Borji, 2022). We employ two existing adversarial training schemes, Madry (Madry et al., 2017) and TRADES (Zhang et al., 2019) and plugin our inductive bias distillation into these frameworks seamlessly to perform adversarial training. The InBiaseD network sees the adversarial images while the ShapeNet only accesses the shape images of the natural images. The ShapeNet trains using just a self-supervised loss but offers bias supervision to the InBiaseD network. The detailed method and setup is explained in the Algorithm 3 and Section A.4 in Appendix.

Table 1: Evaluation of adversarially trained models on TinyImageNet dataset under PGD attack with ϵ=8. Adding InBiaseD to the adversarial training schemes provides a better trade-off in both generalization and robustness.

		Accuracy	Robustness		
			PGD-10	PGD-20	PGD-100
Madry	Baseline	40.33 ±0.24	24.58 ±0.28	19.69 ±0.10	18.96 ±0.11
	InBiaseD	**42.18** ±0.21	**26.26** ±0.23	**21.36** ±0.25	**20.79** ±0.28
TRADES	Baseline	45.90 ±0.26	24.82 ±0.30	18.80 ±0.18	18.12 ±0.19
	InBiaseD	**47.35** ±0.52	**26.59** ±0.38	**20.51** ±0.14	**19.86** ±0.06

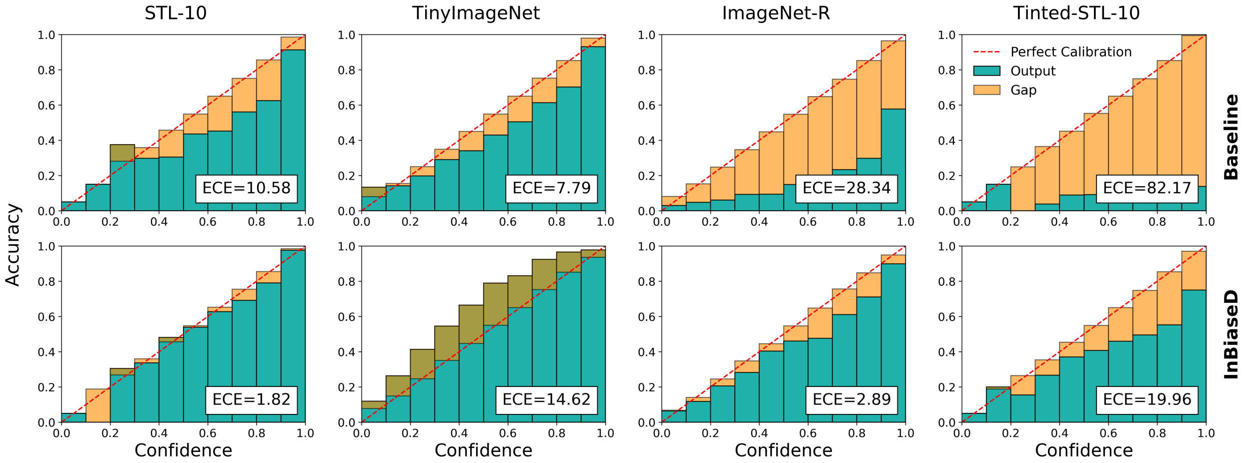

Figure 7: Reliability diagrams and ECE scores to measure calibration. InBiaseD reduces the error and is more underconfident and cautious while Baseline is overconfident with the predictions.

Integrating InBiaseD to adversarial training results in better generalization and robustness of both the training schemes (Table 1). With Madry, InBiaseD shows an improvement of around 4.5% in natural and 6.8% in adversarial accuracy. With TRADES, InBiaseD shows a 3% gain in natural and 7% in adversarial accuracy. Owing to the added shape knowledge, InBiased would require higher perturbations in the semantic regions to be fooled by the attack, thus rendering it more robust. Distilling inductive biases show promising results in achieving better-trade off between generalization and robustness of the models and open up new avenues for exploration.

8 Calibration Analysis

Many applications, especially safety-critical ones need the models to be highly accurate and reliable. Models must not only be accurate but should also indicate when they are likely to be incorrect. Model calibration refers to the accuracy with which the scores provided by the model reflect its predictive uncertainty. Most works focus on improving only predictive accuracy but it is essential to have a model that is also well-calibrated. Expected Calibration Error (ECE) (Guo et al. (2017)) signifies the deviation of the classification accuracy from the estimated confidence.

The ECE scores and reliability diagrams for four datasets are provided in Figure 7. While Baseline predictions are overconfident for all the datasets, InBiaseD either has significantly lower ECE and hence is better calibrated (e.g. for the challenging OOD dataset, ImageNet-R), or it generates more prudent predictions (i.e. for TinyImageNet). This indicates that standard training relying on local solutions makes decisions with high confidence but when shape-awareness is added, the decisions become more prudent which is imperative in safety-critical applications.

9 Comparative Evaluation

InBiaseD shows benefits in reducing the shortcoming of DNNs to shortcut learning, and texture bias and improves robustness. We provide an extensive evaluation to test the standard classification performance of InBiaseD against

Table 2: Image classification accuracy on different IID, OOD, and shortcut learning datasets. Comparison of InBiaseD against Baseline, SelfDistil (self distillation between two networks both training on RGB data). The highest accuracy is in bold. For the cases where higher inference resource is available, DeepEnsemble (ensemble of two networks trained on RGB data) is compared against InBiaseD$_{En}$ (ensemble of InBiaseD and ShapeNet). If the highest accuracy is in the ensemble-based method, it is underlined.

	IID		OOD		Shortcut Learning		
	STL-10	TinyImageNet	ImageNet-R	ImageNet-C	T-STL-10	C-MNIST	S-CelebA
Baseline	81.02±0.89	62.20±0.50	15.07±0.32	21.89±0.16	14.67±0.64	9.89±0.06	66.10±1.67
SelfDistil	82.54±0.12	64.10±0.45	15.17±0.30	21.66±0.17	13.65±0.24	9.92±0.19	68.36±1.31
DeepEnsemble	82.80±0.26	65.40±0.38	16.51±0.24	23.85±0.02	14.54±0.36	12.89±0.04	66.11±0.77
InBiaseD	**84.68**±0.32	**65.66**±0.14	**17.57**±0.11	**23.37**±0.18	**57.34**±1.89	**57.31**±0.71	**71.78**±1.22
InBiaseD$_{En}$	84.90±0.29	64.23±0.29	13.74±0.49	22.94±0.03	77.11±0.73	65.40±0.48	74.22±1.36

different techniques in Table 2. *Baseline* is the standard network trained and tested on RGB images. We also include the techniques that are used to improve generalization, namely ensembles and self-distillation. Hence, we also evaluate *DeepEnsemble* where two randomly initialized networks are independently trained on RGB images and their average predictions used at inference (after softmax). *SelfDistil* trains two networks simultaneously similar to InBiaseD but both networks take RGB images as the input to improve the performance. To compare with the DeepEnsemble, we also report the ensemble of InBiaseD and ShapeNet networks, referred to as InBiaseD$_{En}$. Note that the ensemble-based techniques require more resources as two (or more) networks are needed during inference.

We analyze the classification performance on IID data, OOD data, and three shortcut learning datasets. For the standard evaluation on IID data, we report the accuracy on STL-10 and TinyImageNet datasets, and a few more in Appendix in Table 8. For testing the out-of-distribution generalization, we use the networks trained on TinyImageNet and evaluate them on different variants of ImageNet: Imagenet-R (Hendrycks et al., 2021a) containing images from different renditions and ImageNet-C including 19 different corruptions applied on the ImageNet dataset (Hendrycks & Dietterich, 2019). Along with these, performance on ImageNet-B (Hendrycks et al., 2021a) and ImageNet-A (Hendrycks et al., 2021b) datasets, which contain blurry images and naturally occurring adversarial examples respectively, are reported in Table 9 in Appendix. SelfDistil performs better than Baseline in IID and is one of the shortcut learning datasets. DeepEnsemble achieves comparable performances to InBiaseD in IID and OOD scenarios as expected from the behavior of the ensemble, however, it significantly lags behind in the shortcut learning scenario. Note that if inference cost is not a bottleneck in an application, InBiaseD$_{En}$ can be a better candidate as it achieves comparable performance in IID and OOD scenarios compared to DeepEnsemble while outperforming in the shortcut learning scenario with a high margin. Overall, InBiaseD fares well on all the datasets and scenarios, as it significantly outperforms on shortcut learning datasets while also achieving comparable (or slightly higher) results on the IID and OOD scenarios against the commonly used generalization techniques. Therefore, InBiaseD offers a more generic and effective solution to tackle shortcut learning and improve generalization.

10 Conclusion

To tackle shortcut learning and texture bias present in CNNs, We introduce a method to distill inductive bias knowledge into the neural networks in terms of improved shape-awareness. Our proposed method, *InBiaseD*, is less vulnerable to shortcut learning and shows lesser texture bias. Furthermore, we observe a prominent improvement in robustness to adversarial perturbations. InBiaseD also results in a better trade-off in generalization and robustness when it is plugged into adversarial training schemes, hence opening new horizons for further exploring this path for an improved adversarial training scheme. Furthermore, InBiaseD achieves improved generalization performance on in-distribution as well as out-of-distribution data compared to the standard techniques. These findings indicate that making CNNs more shape-aware offers an effective and generic solution that reduces shortcut learning and texture bias behavior, and also improves generalization and robustness. In future work, we plan to extend the benefits of inductive bias to improve the transfer learning capability of neural networks to complex dense prediction tasks such as object detection and semantic segmentation. InBiaseD as a simple, yet effective and flexible method presents a promising avenue for incorporating different inductive biases in the future. Our results also highlight that distilling inductive bias has a positive impact and presents a compelling case for further exploration in incorporating higher-level cognitive biases.

REFERENCES

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Adversarial concurrent training: Optimizing robustness and accuracy trade-off of deep neural networks. *arXiv preprint arXiv:2008.07015*, 2020.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Ali Borji. Shape defense. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pp. 15–20. PMLR, 2022.

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

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2180–2188, 2016.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721–725, 2001.

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, 2019.

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021b.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Saachi Jain, Dimitris Tsipras, and Aleksander Madry. Combining diverse feature priors. *arXiv preprint arXiv:2110.08220*, 2021.

Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. 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.

Jonas Kubilius, Stefania Bracci, and Hans P Op de Beeck. Deep neural networks as a computational model for human shape sensitivity. *PLoS computational biology*, 12(4):e1004896, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and Cihang Xie. Shape-texture debiased neural network training. *arXiv preprint arXiv:2010.05981*, 2020.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE international conference on image processing (ICIP)*, pp. 1353–1357. IEEE, 2018.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Samuel Ritter, David GT Barrett, Adam Santoro, and Matt M Botvinick. Cognitive psychology for deep neural networks: A shape bias case study. In *International conference on machine learning*, pp. 2940–2949. PMLR, 2017.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 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.

Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pp. 271–272, 1968.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

A Appendix

A.1 Additional Related Works

Borji (2022) uses edge detection to enhance the adversarial robustness and introduce a adversarial training technique. Two techniques are proposed: (1) input-augmentation where the edge map is augmented as an additional channel for adversarial training. (2) using GANs to map edge maps to clean images. The methodology requires edge maps always during inference along with the generative network for method (2). There are requirements needed in the inference and also their results for TinyImageNet do not show improvement in the trade-off between natural and adversarial accuracy. Also, they are not compared with existing adversarial training techniques.

A.2 Methodology

The section provided more details about the algorithms used for shape detection, InBiaseD, and InBiaseD Adversatial Training.

A.2.1 Shape Detection Algorithm

We tried two different edge detection algorithms: Sobel (Sobel & Feldman, 1968) and Canny (Ding & Goshtasby, 2001). The Canny edge detector outputs a binary edge image while Sobel produces a softer output (see Figure 8). Sobel algorithm is chosen for this study.

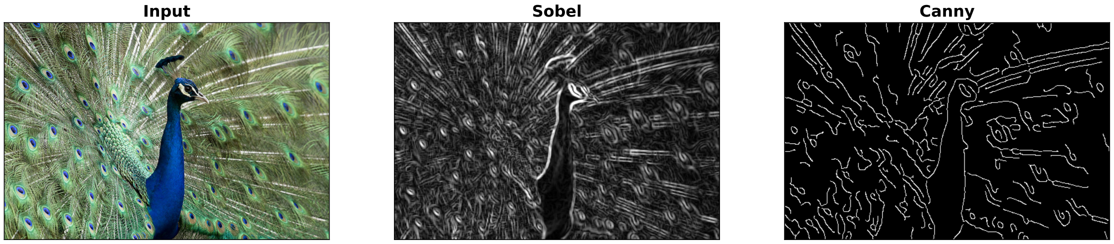

Figure 8: Comparison of different edge detection algorithms: Sobel and Canny.

Algorithm 1 Sobel Edge Detection Algorithm

```
Input: Input image $X$
Up-sample the images to twice the original size: $I$ = Upsample$(X)$
Apply Gaussian smoothing to reduce noisy edges: $I_b$ = Gaussian_Blur$(I, kernel\_size = 3)$
Get Sobel kernels: $G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$ and $G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$
Apply Sobel kernels: $I_{dx} = I_b * G_x$ and $I_{dy} = I_b * G_y$
where $*$ here denotes the 2-dimensional signal processing convolution operation
The edge magnitude: $I_{shape} = \sqrt{I_{dx}^2 + I_{dy}^2}$
Down-sample to original image size: $I_{shape}$ = Downsample$(I_{shape})$
```

Algorithm 2 InBiaseD Algorithm

```
**Input:** Dataset $D$, Batch size $m$
**Initialize:** Two networks: encoders parameterized by $\theta$ and $\theta_{ib}$ and classifiers parameterized by $\phi$ and $\phi_{ib}$
**while** Not Converged **do**
    Sample mini-batch: $(x_1, y_1), ..., (x_m, y_m) \sim D$
    Apply Sobel algorithm to extract shape images: $(x_{ib1}, y_1), ..., (x_{ibm}, y_m)$;
    where $x_{ib} = Sobel(x)$ (Algorithm 1)
    Get the encoded features $z = f(x;\theta)$ and $z_{ib}$ = $f'(x_{ib};\theta_{ib})$, and logits $g(z;\phi)$ and $g'(z_{ib};\phi_{ib})$
    Calculate the Decision Alignment loss $\mathcal{L}_{DA}$ (Equation 1)
    Calculate the Feature Alignment loss $\mathcal{L}_{FA}$ (Equation 2)
    Compute overall loss for each network $\mathcal{L}$ (Equation 3 and 4)
    Compute stochastic gradients and update the parameters $\theta$, $\phi$ and $\theta_{ib}$ , $\phi_{ib}$
**Return:** InBiaseD network ($\theta$, $\phi$) for inference
```

Algorithm 3 InBiaseD Adversarial Training

```
**Input:** Dataset $D$, Batch size $m$
**Initialize:** Two networks: Encoders $f$ and $f'$ parameterized by $\theta$ and $\theta_{ib}$ ; Classifiers $g$ and $g'$ by $\phi$ and $\phi_{ib}$
**while** Not Converged **do**
    Sample mini-batch: $(x_1, y_1), ..., (x_m, y_m) \sim D$
    Extract shape images: $x_{ib,j} = Sobel(x_j), j \in 1, ..., m$ (Algorithm 1)
    Sample perturbation $\delta$ from a set of allowed perturbations $S$ bounded by $\epsilon$
    **if** Madry **then**
        $\delta^* = \arg\max_{\delta \in S} \mathcal{L}_{cls}(\theta, \phi, \delta)$                ▷ Adversarial perturbation
        $\mathcal{L}_{adv} = \mathcal{L}_{cls}(x + \delta^*; \theta, \phi)$
    **else if** TRADES **then**
        $\delta^* = \arg\max_{\delta \in S} \Big[\mathcal{L}_{cls}(x + \delta; \theta, \phi) + \alpha\mathcal{D}_{KL}\Big(f(g(x))||f(g(x + \delta))\Big)\Big]$   ▷ Adversarial perturbation
        $\mathcal{L}_{adv} = \mathcal{L}_{cls}(x + \delta^*; \theta, \phi) + \alpha\mathcal{D}_{KL}\Big(f(g(x))||f(g(x + \delta^*))\Big)$
    Compute overall loss for each network:
    $\mathcal{L} = \mathcal{L}_{adv} + \lambda\mathcal{L}_{DA} + \gamma\mathcal{L}_{FA}$
    $\mathcal{L}_{ib} = \mathcal{L}_{cls}(\theta_{ib}, \phi_{ib}, x_{ib})$
    Compute stochastic gradients and update the parameters $\theta$, $\phi$ and $\theta_{ib}$ , $\phi_{ib}$
**return** InBiaseD network $(f, g)$
```

A.3 EXPERIMENTAL SETUP

The datasets are shown in Table 3 and the hyperparameters used in all the experiments for each dataset are provided in Table 4. For InBiaseD, we uesd the same architecture and settings as for Baseline. The learning rate is set to 0.1, weight decay to $5e-4$ and batch size is 128 (except for PartImageNet, where it is 64). InBiased also has four additional hyperparameters in terms of the loss balancing weights between the two networks (λ and γ).

Table 3: Different datasets used in this study.

IID	Shortcut Learning	Texture Bias	OOD
CIFAR-10	Tinted-STL-10	Stylized-TinyImageNet	ImageNet_C
CIFAR-100	Skewed-CelebA		ImageNet_R
STL-10	Colored-MNIST		ImageNet_B
TinyImageNet			ImageNet_A
Part-ImageNet			

Table 4: Experimental setup for all the experiments. The learning rate is set to 0.1 (except for C-MNIST where it is 0.01). SGD optimizer is used with a momentum of 0.9 and a weight decay of $1e-4$. Resnet-18* refers to the CIFAR-version in which the first convolutional layer has 3x3 kernel and the maxpool operation is removed.

Dataset	Architecture	# Epochs	Scheduler	$\lambda_{f \rightarrow f'}$	$\lambda_{f' \rightarrow f}$	$\gamma_{f \rightarrow f'}$	$\gamma_{f' \rightarrow f}$
Tinted-STL-10	ResNet-18*	200	Cosine	5	1	1	5
Colored-MNIST	MLP	100	Cosine	50	1	1	50
Skewed-CelebA	ResNet-18*	200	Cosine	1	1	1	5
STL-10	ResNet-18*	200	Cosine	1	1	1	5
CIFAR-10	ResNet-18*	200	Cosine	1	1	1	5
CIFAR-100	ResNet-18*	200	Cosine	1	1	1	5
TinyImageNet	ResNet-18*	250	Cosine	1	1	1	5
PartImageNet	ResNet-18	100	MultiStep	1	1	1	5

A.4 Experimental Setup: Adversarial Training

For adversarial training, we use the standard Madry and TRADES adversarial training schemes and add InBiaseD as the plugin. ResNet-18 is used as the architecture and the training is performed on the TinyImageNet dataset for 100 epochs with a learning rate of 0.1 using SGD optimizer and CosineLR scheduler. We use PGD with $\epsilon = 8$ and $step\_size = 0.03$. The additional regularization hyperparameter in TRADES is set to 5.0. For InBiased, we use similar setup and parameters and the additional loss balancing hyperparameters are set to $\lambda_{f-f'}$=1, $\lambda_{f'-f}$=1, $\gamma_{f-f'} = 1$ and $\gamma_{f'-f} = 5$, respectively.

A.5 Numerical Results

Table 5: Statistical irregularity analysis (numerical results of Figure 4).

Fourier filter radius	0	1	2	3	4
Baseline	62.20±0.50	29.20±0.49	15.66±0.34	5.65±0.68	1.88±0.35
InBiaseD	**65.66**±0.14	**33.78**±0.63	**18.78**±0.19	**7.34**±0.37	**2.51**±0.25

Table 6: Evaluation of texture bias (numerical results of Figure 5): models trained on TinyImagenet and tested on stylized images with different styles and varying strengths.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Baseline	53.40±0.17	47.27±0.23	37.02±0.11	27.43±0.55	20.55±0.35	15.43±0.54	11.70±0.55	9.14±0.60	7.01±0.37	5.64±0.32
InBiaseD	**56.79**±0.20	**48.62**±0.44	**39.74**±0.32	**29.97**±0.48	**23.18**±0.66	**18.23**±0.49	**13.49**±0.80	**10.82**±0.77	**8.75**±0.53	**7.15**±0.51

Table 7: Evaluating PGD-10 adversarial attack at varying strengths on models trained on TinyImageNet dataset (numerical results of Figure 6).

	0	0.25	0.5	1.0	2.0	4.0
Baseline	62.20±0.50	49.00±0.28	36.12±0.70	15.75±0.58	2.43±0.09	0.23±0.06
InBiaseD	**65.66**±0.14	**54.29**±0.45	**44.02**±0.15	**24.87**±0.19	**6.52**±0.18	**0.79**±0.02

We present additional results on different datasets and techniques to compare the benefits of InBiaseD. Tables 8, 9 and 10 show the additional results on the IID, OOD and the shortcut learning datasets, respectively. *Baseline* refers to the individual network trained and tested on RGB images. Similarly to test the efficacy of training only on shape data, we provide the *Baseline (Shape)* that is trained only on the shape and also tested on the extracted shape data. The resultant accuracy on IID data is significantly lower compared to the RGB Baseline. Hence, just texture or shape alone is not the optimal solution. We also consider other generalization techniques like *SelfDistil*, which is a self-distillation technique trained using two networks similar to InBiaseD but the input to the second network is RGB instead of shape. Our proposed method constitutes of two networks: *InBiaseD* receiving original images while *ShapeNet* receiving the corresponding shape images. In this section, we present the results of both networks. ShapeNet is inferred on the shape images of the test data. Another way of combining two modalities is via the ensembles. We report the results on *DeepEnsemble* technique, which consists of two networks, each randomly initialized and trained on RGB images only. For comparison, we also report *InBiaseD*$_{En}$, which is an ensemble of *InBiaseD* and *ShapeNet* networks. But the resource consumption is double as two networks are used during inference.

InBiaseD shows superior overall performance against multiple techniques across all datasets, with the biggest benefit being on shortcut learning analyses 10. An important observation is that on shortcut learning datasets, Baseline(Shape) does considerably well as this network only sees shape images and not the original RGB images with the spurious cues. However, together with the results on IID and OOD generalization implies that shape alone is not enough. On the other side, InBiaseD offers an optimal and generic solution for all the scenarios when IID, OOD generalization, and shortcut learning are considered.

Table 8: Comparison: IID performance. Comparison of InBiaseD against Baseline(RGB), Baseline(Shape), and SelfDistil. The highest accuracy is in bold. If inference resource is available to use ensembles, DeepEnsemble (ensemble of two networks trained on RGB data) is reported to compare against InBiased$_{En}$ (ensemble of InBiaseD and shapeNet). If the highest accuracy is in the ensemble-based method, it is underlined.

	CIFAR-10	CIFAR-100	STL-10	TinyImageNet	PartImageNet
Baseline	95.43±0.05	78.35±0.62	81.02±0.89	62.20±0.50	82.32±0.09
Baseline (Shape)	85.19±0.39	61.23±0.60	77.76±0.95	48.98±0.33	74.07±0.14
SelfDistil	**95.50**±0.18	79.38±0.60	82.54±0.12	64.10±0.45	80.06±0.58
DeepEnsemble	95.95±0.04	79.97±0.21	82.80±0.26	65.40±0.38	81.76±0.11
ShapeNet	89.23±0.12	63.68±0.43	80.85±0.49	52.73±0.35	78.12±0.15
InBiaseD	95.45±0.13	**80.20**±0.03	**84.68**±0.32	**65.66**±0.14	**85.16**±0.20
InBiaseD$_{En}$	94.61±0.14	77.78±0.12	84.90±0.29	64.23±0.29	83.50±0.20

Table 9: Comparison: OOD performance.Comparison of InBiaseD against Baseline(RGB), Baseline(Shape), and SelfDistil. The highest accuracy is in bold. If inference resource is available to use ensembles, DeepEnsemble (ensemble of two networks trained on RGB data) is reported to compare against InBiased$_{En}$ (ensemble of InBiaseD and ShapeNet). If the highest accuracy is in the ensemble-based method, it is underlined.

	ImageNet-R	ImageNet-C	ImageNet-B	ImageNet-A
Baseline	15.07±0.32	21.89±0.16	34.29±1.41	2.31±0.18
Baseline (Shape)	5.18±0.26	16.59±0.08	12.50±1.27	1.72±0.23
SelfDistil	15.17±0.30	21.66±0.17	**34.54**±1.14	2.43±0.15
DeepEnsemble	16.51±0.24	23.85±0.02	36.30±0.24	2.53±0.24
ShapeNet	4.47±0.27	17.49±0.05	12.90±0.50	1.66±0.16
InBiaseD	**17.52**±0.55	**23.37**±0.18	33.65±1.68	**2.74**±0.30
InBiaseD$_{En}$	13.74±0.49	22.94±0.03	30.53±0.86	2.34±0.16

Table 10: Shortcut learning analyses (numerical results of Figure 2).

	T-STL-10	C_MNIST			S-CelebA				
		FG	BG	FG+BG	OV	NB-M	B-F	B-M	NB-F
Baseline	14.67±0.64	9.89±0.06	9.88±0.07	12.41±0.44	66.10±1.67	96.28±0.69	93.71±0.30	64.26±3.19	35.84±2.96
Baseline (Shape)	**77.75**±0.83	**65.73**±0.22	**66.50**±0.25	**66.86**±0.40	55.90±0.35	41.15±0.10	71.45±0.30	36.11±6.24	**63.70**±6.24
SelfDistil	13.65±0.24	9.92±0.19	10.31±0.89	13.21±0.45	68.36±1.31	95.94±0.66	70.84±0.30	64.44±4.76	31.16±2.14
DeepEnsemble	14.54±0.36	12.89±0.04	11.06±0.12	14.21±0.34	66.11±0.78	96.89±0.33	94.89±0.05	63.70±4.02	35.12±1.41
ShapeNet	65.94±0.67	65.28±0.32	65.57±0.88	60.26±1.45	54.22±0.34	65.56±4.90	61.49±7.90	56.38±7.60	43.57±6.80
InBiaseD	57.34±1.89	57.31±0.71	33.83±0.47	20.53±0.55	**71.78**±1.23	**97.50**±0.52	**95.67**±0.36	**70.19**±3.21	45.90±2.40
InBiaseD$_{En}$	77.11±0.73	65.40±0.48	54.00±0.90	63.59±0.14	74.22±1.36	95.82±0.58	92.84±2.59	69.72±3.53	43.81±2.20

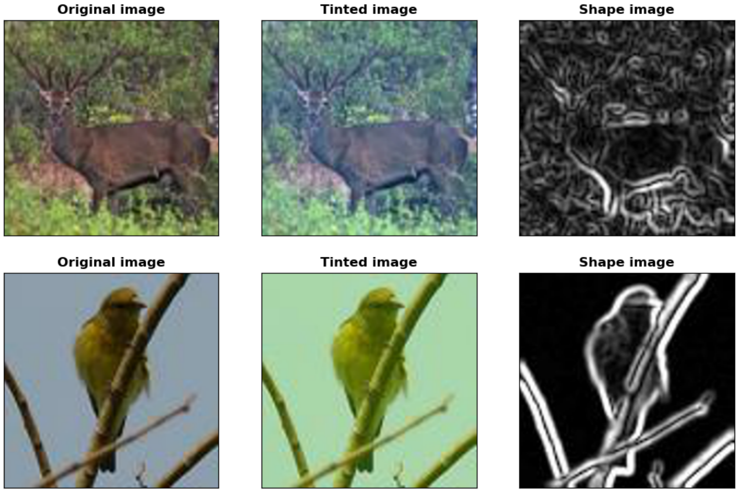

Figure 9: Shortcut Learning: examples of Tinted-STL-10 dataset where a class-specific tint is added to the original STL-10 images as a spurious correlation.

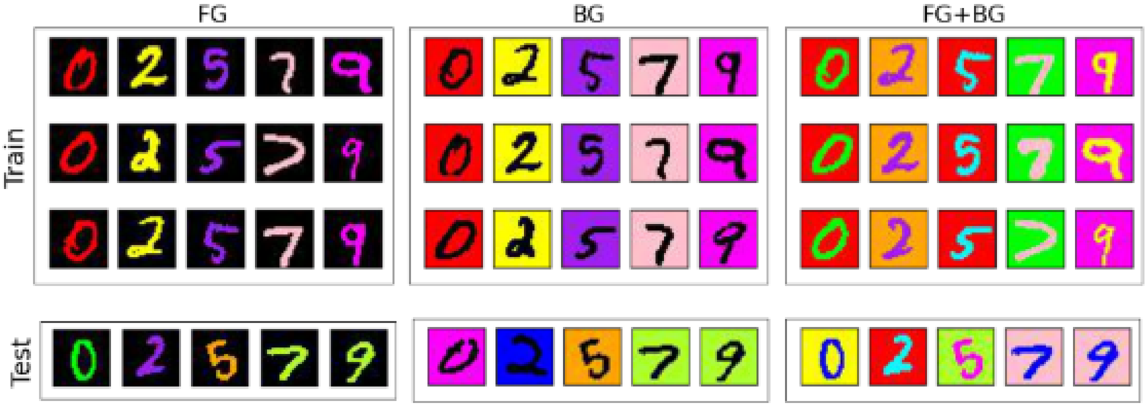

Figure 10: Shortcut Learning: examples of Colored-MNIST dataset. FG : a class-specific color is added to the foreground digits, BG : a class-specific color is added to the background, FG+BG: a class-specific color combination is added to both the foreground-background of the MNIST dataset.

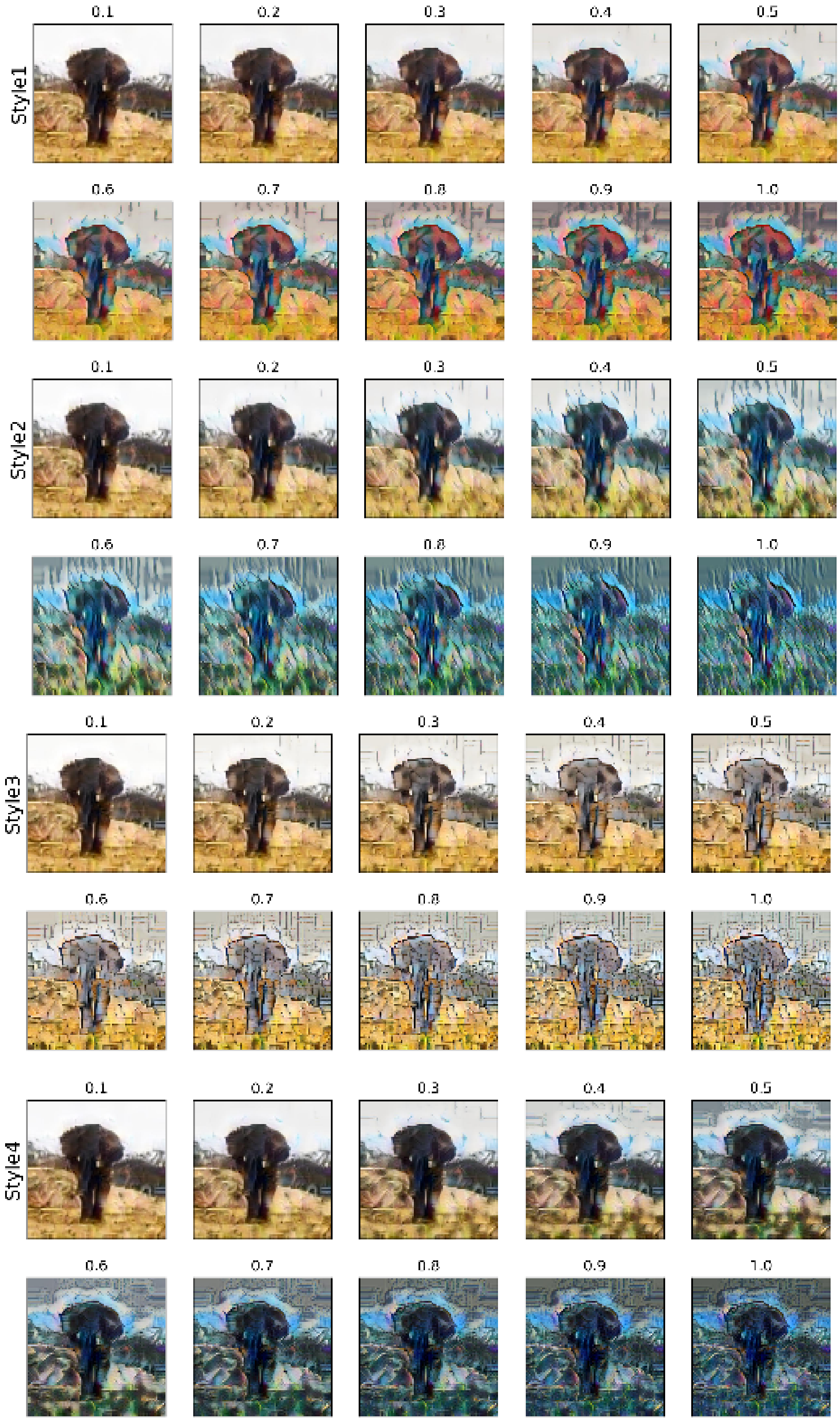

Figure 11: Texture Bias: Examples of four different styles applied on TinyImageNet on strengths ranging from 0.1 to 1.0