AugMax: Adversarial Composition of Random Augmentations for Robust Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data augmentation is a simple yet effective way to improve the robustness of deep neural networks (DNNs). Diversity and hardness are two complementary dimensions of data augmentation to achieve robustness. For example, AugMix explores random compositions of a diverse set of augmentations, while adversarial training generates adversarially hard samples. Motivated by this, we propose a data augmentation framework, termed AugMax, to unify the two concepts: AugMax first randomly samples multiple augmentation operators and then learns an adversarial mixture of the selected operators. Being a stronger form of data augmentation, AugMax leads to a significantly augmented input distribution which makes model training more challenging. To solve this problem, we further design a disentangled normalization module, termed DuBIN (Dual-Batch-and-Instance Normalization), that disentangles the instance-wise feature heterogeneity arising from AugMax. Experiments show that AugMax-DuBIN leads to significantly improved out-of-distribution robustness, outperforming prior arts by 2.01%, 2.52%, 1.84% and 2.13% on CIFAR10-C, CIFAR100-C, Tiny ImageNet-C and ImageNet-C.

1 Introduction

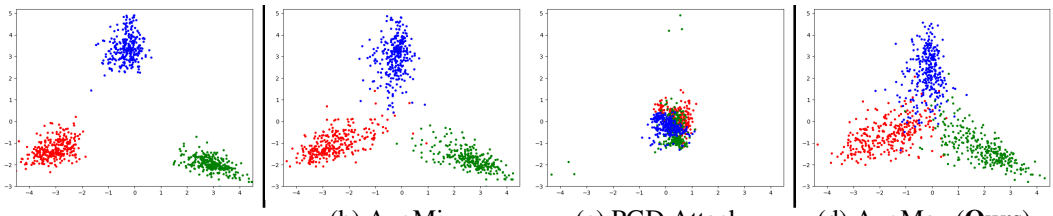
Out-of-distribution (OOD) samples present a challenge when deploying AI models in the real world. Examples include natural corruptions (*e.g.*, due to camera blurs or noise, snow, rain, or fog in image data), sensory perturbations (*e.g.*, sensor transient error, electromagnetic interference) and domain shifts (*e.g.*, summer $\rightarrow$ winter). However, deep neural networks are often trained on limited amounts of data which may not contain these scenarios. As a result, they are vulnerable to unforeseen distributional changes despite achieving high performance on standard benchmarks [1–3]. This jeopardizes the trustworthiness of those models and their safe deployment in real-world environments. Thus it is critical to develop techniques that improve robustness even when training with relatively clean datasets.

Several techniques have been proposed to consolidate the robustness against unforeseen corruptions, including robust data augmentation [5, 3, 8, 6], Lipschitz continuity [9–11], stability training [12], pre-training [13–15], and robust network structures [16–18], to name a few. Among these techniques, data augmentation is of particular interest due to its empirical effectiveness, ease of implementation, low computational overhead, and plug-and-play nature.

There are mainly *two categories* of data augmentation approaches.

The first category aims to increase *diversity* of the training data by composing multiple random transformations [19–21, 5, 3]. While standard data augmentation methods (*e.g.*, random flipping and translation) lead to poor robustness [5], more aggressive combinations of multiple augmentations have shown promise. The most successful example of such augmentations is possibly AugMix [5],

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) Standard (b) AugMix (Diversity) (c) PGD Attack (Hardness) (d) AugMax (**Ours**) (Diversity & Hardness)

Figure 1: **The effects of diversity and hardness in data augmentations.** We visualize features of augmented images fed to the network during training. The features are from the penultimate layer of a ResNeXt29 trained on CIFAR10 training data using only standard data augmentation (random flipping and translation) following [4]. For visualization, we randomly selected 300 fixed images from 3 fixed classes (denoted by different colors) to which we apply different augmentation methods: (a) standard augmentation (random flipping and translation); (b) AugMix [5]; (c) PGD Attack [6, 7]; (d) AugMax (ours). In order to achieve good model robustness, the augmented training data should both be diverse and also contain enough hard cases. As can be seen, PGD Attack generates hard cases (which the network cannot separate) but are not diverse enough (they are all clustered together) while AugMix creates diverse but not hard samples (they are well separated). By contrast, our approach (AugMax) achieves a unification between hard and diverse samples.

which stochastically samples from diverse augmentation operations and randomly mixing them to produce a high diversity of augmented images. AugMix can increase sample diversity compared to standard augmentations, as shown in Figure 1 (a) and (b).

The second category aims to boost the *hardness* of the training data by sampling from the worst-case augmentations that tries to fool the model into misclassifying the samples. A common technique to achieve this goal is adversarial perturbation [6, 22, 8]. Training over such worst-case samples allows a model to actively fix its generalization weaknesses and empirically leads to improved robustness against corruptions [23, 24, 7]. Due to the extra complexity to generate adversarial perturbations, the improved robustness usually comes at the cost of largely increased training time compared with non-adversarial training methods [6, 25]. An example is the PGD attack [6] which falls into this category, as is shown in Figure 1 (c).

Previous work has focused on leveraging one of these category to improve robustness. In this paper, we *unify* both approaches in a single framework. In particular, we show that diversity and hardness are *complementary* and that a unification between the two is necessary to achieve robustness. We propose a new strategy to achieve this unification and successfully increase robustness.

Summary of contributions:

- We propose **AugMax**, a novel augmentation framework which achieves robustness through a unification between diversity and hardness, by searching for the worst-case mixing strategy.
- Being a stronger form of data augmentation, AugMax leads to a significantly augmented input distribution which makes model training more challenging. To solve this problem, we design a new normalization strategy, termed **DuBIN**, to disentangle the instance-wise feature heterogeneity of AugMax samples.
- We show that combination of AugMax and DuBIN (**AugMax-DuBIN**) achieves state-of-the-art robustness against corruptions and improves robustness against other common distribution shifts.

AugMax achieves a good unification between sample diversity and hard corner-case generation during data augmentation. AugMax is built on top of the AugMix framework [5] which mixes multiple data augmentation operators in a multi-branch and layered pipeline. However, different from AugMix where augmentation operators and mixing weights are both randomly sampled, operators are first randomly sampled, followed by adversarially trained mixture of the selected operators in Augmax. This simple change from AugMix to AugMax results in considerable difference in their feature distributions. From the visualizations (in Figure 1), AugMax generates more adversarially "hard samples", while still keeping a good amount of diversity compared to AugMix and PGD Attack. Searching for adversarial mixing strategies in AugMax is slightly more expensive since it involves adversarial mixing. To make this efficient, we adopt an efficient adversarial training strategy [26]. As a result, AugMax adds only a reasonable amount of extra complexity compared to non-adversarial

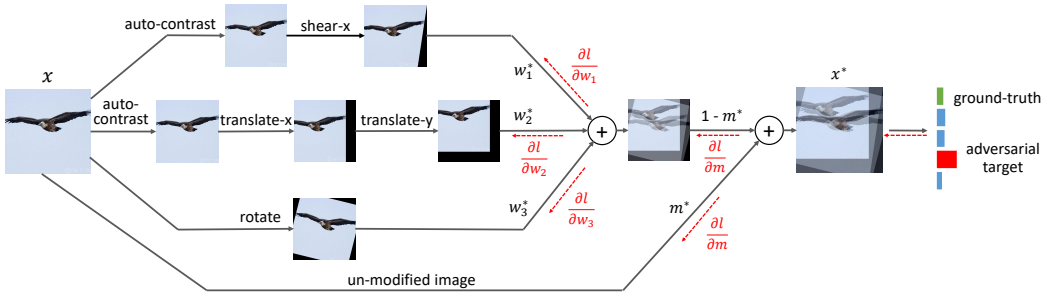


Figure 2: AugMax overview. Black and red arrows represent forward paths and back-propagation paths to generate AugMax images, respectively. In contrast to AugMax, where the mixing parameters m and $\boldsymbol{w}$ are adversarially learned, AugMix randomly samples m and $\boldsymbol{w}$ from predefined distributions (and thus no backpropergation on m and $\boldsymbol{w}$).

training methods, while improving robustness significantly. For example, AugMax training time is only ~ 1.5 times that of AugMix on ImageNet (see Table 7). Moreover, Augmax is significantly more efficient than traditional adversarial training which has ~ 10 times training time overhead compared with AugMix.

Being a stronger form of data augmentation with adversarial sample generation, AugMax leads to a significantly augmented input distribution which makes model training more challenging. This naturally motivates us to propose a novel and finer-grained normalization scheme termed Dual-Batch-and-Instance Normalization (DuBIN). As illustrated in Figure 3, DuBIN adds an additional instance normalization (IN) in parallel to the traditional Dual Batch Normalization (DuBN) [7, 27] used in adversarial training, in order to better model and disentangle the instance-wise feature heterogeneity arising from AugMax. We show that adding the instance normalization is an important knob to promote the capability of AugMax in boosting model robustness.

Our framework, AugMax-DuBIN, is illustrated in Figure 2. AugMax-DuBIN trains on clean images and achieves state-of-the-art robustness on natural corruption benchmarks [1], and also improves model robustness against other common distribution shifts [28, 2]. In particular, our method surpasses state-of-the-art method on CIFAR10-C, CIFAR100-C, Tiny ImageNet-C and ImageNet-C by 2.01%, 2.52%, 1.84% and 2.13% respectively.

2 Related Work

2.1 Robustness to Unseen Natural Corruptions

Hendrycks *et al.* [1] pioneer the study of prediction errors exhibited by machine learning models on unseen natural corruptions and perturbations. They introduce two variants of the original ImageNet validation set to evaluate the model's robustness: the ImageNet-C for input corruption robustness, and the ImageNet-P for input perturbation robustness. The former applies 15 diverse corruptions drawn from four main categories (noise, blur, weather, and digital) on ImageNet validation images. The latter contains perturbation sequences generated from each image. Recht *et al.* [28, 2] collect CIFAR10.1 and ImageNet-V2 as a reproduction of the original CIFAR10 [29] and ImageNet [30] test sets, and find that a minute natural distribution shift caused by the minutiae in dataset collection process leads to a large drop in accuracy for a broad range of image classifiers. Hendrycks *et al.* [31] collect ImageNet-A with the new concept of natural adversarial examples: unmodified real-world images that falsify common machine learning models. Other representative benchmark efforts include [32–35, 3], all demonstrating the brittleness of deep models under various distribution shifts.

2.2 Data Augmentation: Random and Adversarial

Data augmentations have been widely used to increase training set diversity and improve model generalization ability [36, 20, 37, 21]. Among those methods, AugMix [5] achieves the outstanding robustness against natural corruptions by randomly mixing multiple diverse augmentations. It establishes high performance bars on many natural corruption benchmarks including CIFAR10-C,

CIFAR100-C and ImageNet-C [1]. DeepAugment [3] feeds clean images to image-to-image models with randomly perturbed model weights to generate visually diverse augmented images. Combined with AugMix, it achieves state-of-the-art robustness on ImageNet-C [1].

Besides, Adversarial training (AT) [6] utilizes adversarial samples as a special data augmentation method, and has also shown promise in improving model robustness against natural corruptions [22] and domain gaps [23], usually at the cost of increasing training time [38] and degrading the standard accuracy on clean images [39]. Volpi *et al.* [23] and Zhao *et al.* [24] augment the training set with samples from a fictitious adversarial domain to improve domain adaptation performance. AdvProp [7] shows adversarial training can improve both standard accuracy and robustness against distribution shifts, with the help of auxiliary batch normalization. Adversarial Noise Training (ANT) [8] uses random Gaussian noise with learned hyperparameters (mean and variance) as the augmentation and achieves comparable robustness with AugMix on unseen corruptions.

2.3 Normalization

Batch Normalization (BN) [40] has been a cornerstone for training deep networks. Inspired by BN, more task-specific modifications are proposed by exploiting different normalization axes, such as Instance Normalization (IN) [41], Layer Normalization (LN) [42] and Group Normalization (GN) [43]. A few latest works investigate to use multiple normalization layers instead of one BN. Zajkac *et al.* [44] use two separate BNs to handle the domain shift between labeled and unlabeled data in semi-supervised learning. Xie *et al.* [45, 7] observe the difference between standard and adversarial feature statistics during AT, and craft *dual batch normalization* (DuBN) to disentangle the standard and adversarial feature statistic to improve both the standard accuracy and robustness. While most deep models just use one normalization type throughout the network, IBN-Net [27] unifies IN and BN in one model. It finds that when applied together, IN tends to learn features invariant to appearance (colors, styles, *etc.*) changes, while BN is essential for preserving high-level semantic contents. IBN-Net shows that combining IN and BN in an appropriate manner can improve both standard generalization and domain adaptation. A similar combination was also explored in the style transfer field [46], learning to selectively normalize only disturbing styles while preserving useful styles.

3 Method

3.1 Preliminary: The AugMix Framework

At a high level, AugMix creates a new composite image from a clean sample by first applying several simple augmentation operations to it before combining the results through random linear combinations. This augmentation scheme is coupled with a Jensen-Shannon consistency loss that enforces feature similarity among clean and augmented images. Specifically, AugMix consists of multiple augmentation chains (3 by default), each applied in parallel to the same input image. Each chain composes one to three randomly selected augmentation operations. These augmentations are the same as AutoAugment [20], but excluding those that overlap with ImageNet-C corruptions. The augmented images from each augmentation chains are then combined together with the original, clean sample using a linear combinations with randomly samples coefficients. The final image thus incorporates several sources of randomness, coming respectively from the choice of operations, the severity level of these operations, the lengths of the augmentation chains and the mixing weights. Despite its simplicity, AugMix achieves state-of-the-art corruption robustness. In the remaining of this section, we build on the previous AugMix work and propose AugMax, an augmentation strategy that further improves robustness.

3.2 AugMax: Augmented Training with Unified Diversity and Hardness

In this section, we rigorously introduce our new data augmentation method, AugMax, illustrated in Figure 2. At a high level, the main difference between AugMax and AugMix is a new optimization procedure used to select the mixing weights m and $\boldsymbol{w}$.

Given a data distribution $\mathbb{D}$ over images $\boldsymbol{x} \in \mathbb{R}^d$ and labels $\boldsymbol{y} \in \mathbb{R}^c$, our goal is to train a mapping (classifier) $f : \mathbb{R}^d \to \mathbb{R}^c$ from images to output softmax probabilities, parameterized by $\boldsymbol{\theta}$, that is robust to (unknown) distribution shifts. We minimize the empirical risk

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\, \mathcal{L}(f(\boldsymbol{x};\boldsymbol{\theta}),\, \boldsymbol{y}), \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ is the loss function (*e.g.*, cross-entropy). As illustrated in Figure 2, an AugMax image $\boldsymbol{x}^*$ is generated by learning a set of adversarial mixing parameters $m^*, \boldsymbol{w}^*$ which maximizes the loss:

$$\boldsymbol{x}^* = g(\boldsymbol{x}_{orig}; m^*, \boldsymbol{w}^*), \tag{2}$$

where $g(\cdot)$ denotes the AugMax augmentation function, $\boldsymbol{x}_{orig}$ is the original image, and

$$m^*, \boldsymbol{w}^* = \arg\max_{m,\boldsymbol{w}} \mathcal{L}(f(g(\boldsymbol{x}_{orig}; m, \boldsymbol{w}); \theta), \boldsymbol{y}), \quad \text{s.t. } m \in [0,1], \boldsymbol{w} \in [0,1]^b, \boldsymbol{w}^T\mathbf{1} = 1. \tag{3}$$

To simplify the optimization problem in Eq. (3), we use a re-parameterization trick by setting $\boldsymbol{w} = \sigma(\boldsymbol{p})$, where $\sigma(\cdot)$ is the softmax function, and convert the optimization into a differentiable one:

$$m^*, \boldsymbol{p}^* = \arg\max_{m\in[0,1], \boldsymbol{p}\in\mathbb{R}^b} \mathcal{L}(f(g(\boldsymbol{x}_{orig}; m, \sigma(\boldsymbol{p})); \theta), \boldsymbol{y}); \quad \boldsymbol{w}^* = \sigma(\boldsymbol{p}^*) \tag{4}$$

Training with AugMax augmentation can then be written altogether as minimax optimization:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\, \frac{1}{2}[\mathcal{L}(f(\boldsymbol{x}^*);\boldsymbol{\theta}), \boldsymbol{y}) + \mathcal{L}(f(\boldsymbol{x});\boldsymbol{\theta}), \boldsymbol{y})] + \lambda \mathcal{L}_c(\boldsymbol{x}, \boldsymbol{x}^*) \\ &\text{s.t. } \boldsymbol{x}^* = g(\boldsymbol{x}; m^*, \boldsymbol{w}^*); \quad m^*, \boldsymbol{w}^* = \arg\max_{m\in[0,1], \boldsymbol{p}\in\mathbb{R}^b} \mathcal{L}(f(g(\boldsymbol{x}; m, \sigma(\boldsymbol{p})); \boldsymbol{\theta}), \boldsymbol{y}) \end{aligned} \tag{5}$$

where $\mathcal{L}_c$ is a consistency loss regularizing augmented images to have similar model outputs with the original images, following [5], and λ is the trade-off parameter. $\mathcal{L}_c$ can have many different implementations. Our default one is adapted from [5]:

$$\mathcal{L}_c(\boldsymbol{x}, \boldsymbol{x}^*) = \text{JS}(f(\boldsymbol{x};\theta), f(\boldsymbol{x}^*;\theta)) \tag{6}$$

with $\text{JS}(\cdot)$ being the Jensen-Shannon divergence. In order to reduce the training complexity overhead caused by adversarial training, we employ an accelerated adversarial attack method [26] to solve Eq. (4), which adds only a reasonable amount of extra complexity compared to AugMix baseline. The algorithm to solve Eq. (5) is summarized in Appendix B.

Visualization of the effects of diversity and hardness To illustrate the motivation behind AugMax, we visualize the feature representations induced by different augmentation methods to showcase their diversity and hardness in Figure 1. Specifically, we visualize the impact of various data augmentation methods on the feature representations obtained with a ResNeXt29 that is normally trained on CIFAR-10. Note that the model is trained using only default standard augmentations (*i.e.*, random flipping and translation), and therefore has never seen augmentation from AugMix, AugMax, or PGD attack. Following [4], we visualize the penultimate layer feature distributions of training samples from three different classes. 300 training images are selected from each class for visualization. For AugMix and ℓ_∞ PGD attack, we use the same hyperparameters in the original papers [5, 6].

We observe in Figure 1 that the clean images form well-separated clusters in the feature space, leaving large regions between those clusters empty. This potentially causes uncertain and unreliable generalization if new test samples (from a shifted distribution) fall into those regions. AugMix enlarges each cluster to cover a broader region, effectively increasing sample diversity. However, after augmentation, very few samples are close to the decision boundaries, revealing an inability to generate hard samples. Samples from the PGD attack, on the other hand, collapse into a small region around the classification borders, losing feature diversity. We further investigate the method through ablation studies in Section 4.3.

3.3 DuBIN: Disentangled Normalization for Heterogeneous Features

AugMax leads to better coverage of the input space by unifying sample diversity and hardness, as demonstrated in Figure 1. While such more versatile distribution in data augmentation has the potential to increase robustness, we do observe that *naive* incorporation of AugMax leads to relatively marginal improvement over AugMix (see Section 4.3). This is due to the comprehensiveness of AugMax can also lead to a higher feature heterogeneity, which may require larger model capacity to encode. To address this problem, we design a novel normalization layer, termed *Dual Batch-and-Instance Normalization* (**DuBIN**), to disentangle the instance-level heterogeneity.

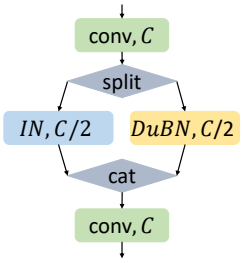


Figure 3: Illustration of DuBIN. "Split" and "cat" represent splitting and concatenating on features along the channel dimension. Channel number (*i.e.*, C or $C/2$) is shown alongside the layer name.

As illustrated in Figure 3, DuBIN consists of two parallel parts: (1) a dual batch normalization (DuBN) [7], which is the **by-default** normalization layer to use for adversarial training [7, 45], to disentangle the group-level statistics of the clean and corner-case augmented samples[1]; and (2) an additional instance normalization, to account for instance-level feature statistics due to augmentation diversity. At each layer, the input feature is split into halves along the channel dimension: one half is then fed into IN, and the other fed into DuBN. Finally, the outputs of IN and DuBN are concatenated back along the channel dimension for the next layer.

To provide additional insights of DuBIN, we compare the BN statistics of AugMax-DuBN and AugMax-DuBIN in Table 1. $\overline{\sigma}_c^2$ and $\overline{\sigma}_a^2$ represent the average batch normalization variance over all channels of BN_c and BN_a[1], respectively. We observe that both $\overline{\sigma}_c^2$ and $\overline{\sigma}_a^2$ are smaller in the DuBIN network than those in the DuBN counterpart. This shows that the IN branch in DuBIN can reduce the feature diversity that BN otherwise needs to model, by encoding instance-level diversities. With lower feature variation, BNs can converge better with improved performance [47].

Table 1: BN statistics of different layers in WRN40-2 with DuBN or DuBIN trained on CIFAR100 using AugMax training.

Layer		Block 1 layer 2	Block 1 layer 3	Block 2 layer 2
AugMax-DuBN	$\overline{\sigma}_c^2$	0.0369	0.0450	0.0301
	$\overline{\sigma}_a^2$	0.0469	0.0585	0.0382
AugMax-DuBIN	$\overline{\sigma}_c^2$	0.0306	0.0403	0.0264
	$\overline{\sigma}_a^2$	0.0348	0.0466	0.0292

4 Experiments

Here, we introduce in detail the experimental setting, and report quantitative results for robustness against natural corruptions and distribution shifts. We show that our method induces better robustness than existing ones. We study the properties of our approach in thorough ablation studies.

4.1 Experimental Setup

Datasets and Models We evaluate our proposed method on CIFAR10, CIFAR100 [29], ImageNet [30] and Tiny ImageNet (TIN)[2]. For neural architectures, we use a ResNet18 [48], WRN40-2 [49] and ResNeXt29 [50] on the CIFAR datasets, and ResNet18 on ImageNet and Tiny ImageNet[3]. To evaluate the model's robustness against common natural corruptions, we use CIFAR10-C, CIFAR100-C, ImageNet-C and Tiny ImageNet-C (TIN-C) [1], which are generated by adding natural corruptions to the original test set images.

Table 2: Evaluation results on CIFAR10 and CIFAR10-C. The best metric is shown in bold.

Model	Metric	Normal	AugMix	AugMax-DuBIN
ResNet18	SA (%)	95.56	**95.79**	95.63 (-0.16)
	RA (%)	74.75	89.49	**90.62** (+1.13)
WRN40-2	SA (%)	94.78	95.67	**95.68** (+0.01)
	RA (%)	73.71	89.01	**90.67** (+1.66)
ResNeXt29	SA (%)	95.60	96.25	**96.40** (+0.15)
	RA (%)	71.70	89.08	**91.09** (+2.01)

[1]DuBN consists of two separate BNs: one BN_c for clean images and the other BN_a for adversarially augmented (e.g., AugMax) images.

[2]https://www.kaggle.com/c/tiny-imagenet

[3]Our WRN40-2 and ResNeXt29 models are identical to those used in AugMix [5].

Evaluation Metrics Following established conventions [1, 5], we define robustness accuracy (RA) as the average classification accuracy over all 15 corruptions, to measure the models' robustness on CIFAR10-C and CIFAR100-C. On ImageNet-C and Tiny ImageNet-C, we use both RA and mean corruption error (mCE) to evaluate robustness. As defined by [1], mCE is the weighted average of target model corruption errors normalized by the corruption errors of a baseline model across different types of corruptions. We use a conventionally trained ResNet18 on ImageNet or Tiny ImageNet as the baseline model. We use standard accuracy (SA) to denote the classification accuracy on the original clean testing images.

Baseline Methods On CIFAR10-C and CIFAR100-C, we compare our method with the state-of-the-art method, AugMix [5]. In addition, AugMix has previously been combined with other orthogonal methods, such as DeepAug [3], to further boost performance on ImageNet-C. Specifically, [3] combine DeepAug as an orthogonal component to AugMix, and achieve state-of-the-art performance on ImageNet-C with DeepAug + AugMix [3]. To allow comparison with this scenario, we also run the following comparison experiment on ImageNet-C and Tiny ImageNet-C: (1) AugMix v.s. AugMax and (2) DeepAug + AugMix v.s. DeepAug + AugMax. We show that the advantages of AugMix still exist when combined with other orthogonal methods.

Table 3: Evaluation results on CIFAR100 and CIFAR100-C. The best metric is shown in bold.

Model	Metric	Normal	AugMix	AugMax-DuBIN
ResNet18	SA (%)	77.99	78.23	**78.52** (+0.29)
	RA (%)	48.46	62.67	**64.02** (+1.35)
WRN40-2	SA (%)	76.19	**77.03**	76.95 (-0.08)
	RA (%)	46.80	64.56	**66.32** (+1.76)
ResNeXt29	SA (%)	79.95	78.58	**80.16** (+1.58)
	RA (%)	47.76	65.37	**67.89** (+2.52)

Finally, we also compare AugMax with ANT [8], which utilizes Gaussian noises with learned distribution parameters for data augmentation. Since models tend to generalize better on test images similar to those seen during training [51], ANT mainly improves robustness against high frequency corruptions (*e.g.*, additive noises), while achieving comparable robustness on other unseen corruptions (*e.g.*, bluring) with AugMix. Following [8],we perform the comparison on the subset of Tiny ImageNet-C with 12 remaining corruptions after removing all additive noises.

Table 4: Evaluation results on ImageNet and ImageNet-C. The best metric is shown in bold.

Method	SA (%, ↑)	RA (%, ↑)	mCE (%, ↓)
Normal	68.79	13.90	100.00
AugMix	**68.92**	20.93	91.66
AugMax-DuBIN	68.08 (-0.84)	**22.76** (+1.83)	**89.53** (-2.13)
DeepAug + AugMix	**66.63**	25.60	86.32
DeepAug + AugMax-DuBIN	66.07 (-0.56)	**27.20** (+1.60)	**84.44** (-1.88)

Implementation Details In Eq. (4), m and $\boldsymbol{p}$ are uniformly randomly initialized between 0 and 1. We use PGD to update m: first do gradient ascend on m and then project it back into the $[0, 1]$ interval. $\boldsymbol{p}$ is updated using gradient ascend. AugMax follows the same rules as AugMix [5] to randomly select augmentation operation type, severity level and chain length. Following [6], we iteratively solve the inner maximization and outer minimization problems in Eq. (5): the inner maximization is updated for n steps, for every 1 step update on the outer minimization. Both m and $\boldsymbol{p}$ are updated with step size α. We set n as 5 and α as 0.1 as default. During training, we set λ in Eq. (5) to be 10 for all settings. More implementation details (*e.g.*, learning rate, training epoch number, batch size, etc.) are provided in Appendix A.

Table 5: Evaluation results on TIN and TIN-C. The best metric is shown in bold.

Method	SA (%, ↑)	RA (%, ↑)	mCE (%, ↓)
Normal	61.64	23.91	100.00
AugMix	**61.79**	36.85	83.04
AugMax-DuBIN	61.72 (-0.07)	**38.41** (+1.56)	**81.20** (-1.84)
DeepAug + AugMix	55.80	40.21	78.94
DeepAug + AugMax-DuBIN	**55.96** (+0.16)	**40.79** (+0.58)	**78.23** (-0.71)

4.2 Robustness against Natural Corruptions

Results on CIFAR10-C and CIFAR100-C are shown in Table 2 and 3 respectively, where "normal" means training with default standard augmentations (*e.g.*, random flipping and translation). From the results, we see that AugMax-DuBIN achieves new state-of-the-art performance on both benchmark datasets with different model structures. For example, compared with AugMix, our method improves accuracy by as high as 2.01% and 2.52% on CIFAR10-C and CIFAR100-C, respectively. Our method is more beneficial for larger models. Specifically, in both CIFAR10 and CIFAR100 experiments, the

largest robustness improvement is achieved on ResNeXt29, which has the largest capacity among the three models.

Results on ImageNet are shown in Table 4. AugMax-DuBIN outperforms AugMix by 2.13% in terms of mCE on ImageNet-C. Further combining AugMax-DuBIN with DeepAug outperforms AugMix + DeepAug by 1.88% in terms of mCE, achieving a new state-of-the-art performance on ImageNet-C. This shows that AugMax can be used as a more advanced basic building block for model robustness and inspires future researches to build other defense methods on top of it.

Table 6: Evaluation results on TIN and TIN-C (w/o noise). The best metric is shown in bold.

Method	SA (%, ↑)	RA (%, ↑)	mCE (%, ↓)
Normal	61.64	24.38	100.00
AugMix	**61.79**	37.63	82.54
$\text{ANT}^{1\times1}$	60.95	35.70	85.14
$\text{ANT}^{3\times3}$	61.26	35.30	85.70
AugMax-DuBIN	61.72 (-0.07)	**38.92** (+1.29)	**81.08** (-1.46)

Results on Tiny ImageNet-C are shown in Table 5 and Table 6. From Table 5, we see that our method improves the accuracy by 1.56% compared with AugMix and 0.58% when combined with DeepAug. From Table 6, we can see our method outperforms ANT in both the uncorrelated ($\text{ANT}^{1\times1}$) and correlated ($\text{ANT}^{3\times3}$) settings by at least 3.22% RA.

Training Time Since we use an accelerated adversarial training method [26] to solve Eq. (4), the worst-case mixing strategy can be searched efficiently. As a result, AugMax adds only a reasonable amount of extra training complexity compared with AugMix. Moreover, Augmax is significantly more efficient than traditional adversarial training (AT). Detailed training time is shown in Table 7.

Table 7: Training time on ImageNet with ResNet18. All numbers are reported on a single NVIDIA A6000.

Method	Time (sec/epoch)
Normal	2669
AugMix	3622
AT [6]	37162
AugMax	5264

4.3 Ablation Study

Analysis of Different Normalization Layers In this paragraph, we compare AugMax/AugMix when combined with different normalization schemes. When using DuBN or DuBIN, we route original images to the BN_c, and AugMax/AugMix images to BN_a. Since it is the by-default setting to use disentangled normalization layers (e.g., DuBN) for original and augmentation images in adversarial training [7, 45], we follow this traditional routing and combine AugMax with DuBN or DuBIN instead of BN or IBN. It is fair to compare AugMax-DuBN/DuBIN with either AugMix-BN/IBN or AugMix-DuBN/DuBIN.

Table 8: Ablation results on DuBIN. RA (%) on CIFAR10-C and CIFAR100-C with WRN40-2 backbone are reported.

Method	CIFAR10-C	CIFAR100-C
AugMix-BN	89.01 (± 0.03)	64.56 (± 0.04)
AugMix-IBN	89.17 (± 0.24)	63.94 (± 0.28)
AugMix-DuBN	89.11 (± 0.21)	64.07 (± 0.38)
AugMix-DuBIN	89.74 (± 0.35)	65.02 (± 0.58)
AugMax-DuBN	89.60 (± 0.62)	65.06 (± 0.28)
AugMax-DuBIN	**90.67** (± 0.16)	**66.31** (± 0.24)

Results are shown in Table 8. As we can see, AugMax-DuBN outperforms both AugMix-BN and AugMix-DuBN, and AugMax-DuBIN outperforms both AugMix-IBN and AugMix-DuBIN. Noticeably, when combined with AugMix, DuBIN can also help improve robustness. This shows the potential of applying DuBIN to other diversity augmentation methods for general performance improvement.

Table 9: Ablation results on AugMax-DuBIN update step number n and consistency loss tradeoff parameter λ. Results on reported on CIFAR10/CIFAR10-C with WRN40-2.

	n			λ		
	2	5	10	1	10	15
SA (%)	95.78	95.68	95.58	90.34	95.68	95.88
RA (%)	90.49	90.67	90.67	75.29	90.67	90.30

AugMax Hyperparameters In this paragraph, we check the sensitivity of AugMax with respect to its hyperparameters: maximization steps n, step size α, and consistency loss tradeoff parameter λ. We keep $n \times \alpha = 1$ to guarantee m can explore the entire $[0, 1]$ interval in Eq. (4). Results are shown in Table 9. We first fix $\lambda = 10$, and find that smaller n tends to result in better standard accuracy, while larger n benefits more robustness. This is consistent with our intuition since the augmented images tend to be more adversarial if Eq. (4) is solved with more iterations. The changes stabilize at $n = 10$, which we set as the default value. Fixing n at 10 (and thus $\alpha = 0.1$), we find robustness peaks at around $\lambda = 10$, which we set as the default value.

Comparison with Different Diversity and Hardness Strategies In this paragraph, we study different strategies to combine diversity and hardness. The results confirm that diversity and hardness are two complementary dimensions and a proper unification strategy is the key to boost model robustness.

Specifically, we compare AugMax with baseline methods from each strategy group (*i.e.*, diversity or hardness), and other possible strategies to combine both. Besides AugMix and Adversarial Training (AT) [6], which are the two representative examples for diversity and hardness as shown in Figure 1, we also design the following baseline methods. (1) AugMix+AT: using both AugMix and adversarial images for augmentation. This is a naive baseline to combine diversity with hardness. (2) AdvMix: applying adversarial attacks on the augmentation hyperparameters such as the rotation angles and translation pixel numbers [52, 53], while randomly selecting the mixing parameters; (3) AdvMax: applying adversarial attacks on both the augmentation hyperparameters and the mixing weights.[4] Please see Appendix C for their implementation details.

Table 10: Results of different augmentation strategies on CIFAR100(-C) with ResNet18.

Method	Strategy	SA (%)	RA (%)
Normal	-	77.99	48.46
AugMix	(diversity)	78.23	62.67
AT	(hardness)	67.64	52.81
AdvMax		56.61	38.65
AugMix+AT	(diversity & hardness)	69.61	57.66
AdvMix		72.36	56.89
AugMax-DuBIN		**78.52**	**64.02**

The results are shown in Table 10. AugMax-DuBIN achieves the best performance, outperforming all methods from either diversity or hardness group. This shows diversity and hardness are indeed two complementary dimensions and a proper combination of them can boost model robustness. On the other hand, the other two naive baselines jointly considering diversity and hardness achieve poor performance, showing it nontrivial to design a method achieving good balance between diversity and hardness.

4.4 Robustness against Other Distribution Shifts

Although AugMax mainly aims at improving model robustness against common corruptions, we show that it can also generally improve robustness against other types of distribution shifts. Specifically, we evaluate robustness against distribution shifts caused by minor differences in data collection process on CIFAR10.1 [28] and ImageNet-V2 (IN-V2) Threshold0.7 set [2]. Results are reported in Table 11 (using the same ResNeXt29 and ResNet18 models as in Section 4.2). As we can see, AugMax-DuBIN outperforms AugMix on both benchmark datasets.

Table 11: Robustness against other distribution shifts. Accuracy (%) on CIFAR10.1 and IN-V2 are evaluated on ResNeXt29 and ResNet18, respectively.

Method	CIFAR10.1	IN-V2
Normal	88.90	55.79
AugMix	88.90	55.85
AugMax-DuBIN	**89.78**	**55.97**

5 Conclusion

In this paper, we propose AugMax, a novel data augmentation, that significantly improves robustness by achieving a unification between diversity and hardness. To enable efficient training in the face of the resulting heterogeneous feature statistics, we design a novel normalization scheme termed DuBIN. Using the combination of AugMax and DuBIN, we demonstrate state-of-the-art robustness on several natural corruption benchmarks.

AugMax unifies diversity and hardness in a heuristic manner. While we showed in ablation studies that AugMax outperforms other heuristic combinations, there may be other strategies to combine diversity and hardness. We leave such study for future work and focus in this paper on showing the benefit of unifying diversity and hardness, which were separately considered in previous research.

[4] Note that AdvMax implies full hardness and little diversity, while AdvMix switches the order of randomness and adversarial augmentation in the AugMax pipeline.

References

[1] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. 1, 3, 4, 6, 7

[2] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? *arXiv preprint arXiv:1902.10811*, 2019. 3, 9

[3] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 1, 3, 4, 7

[4] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019. 2, 5

[5] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 2, 3, 5, 6, 7

[6] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 4, 5, 7, 8, 9

[7] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4, 6, 8

[8] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision (ECCV)*, pages 53–69, 2020. 1, 2, 4, 7

[9] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, volume 30, pages 2266–2276, 2017. 1

[10] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane S. Boning, Inderjit S. Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pages 5273–5282, 2018.

[11] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018. 1

[12] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4480–4488, 2016. 1

[13] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019. 1

[14] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020.

[15] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 2020. 1

[16] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning (ICML)*, pages 7324–7334, 2019. 1

[17] Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Nicolas Le Roux, and Ross Goroshin. An effective anti-aliasing approach for residual networks. *arXiv preprint arXiv:2011.10675*, 2020.

[18] Jungkyu Lee, Taeryun Won, and Kiho Hong. Compounding the performance improvements of assembled techniques in a convolutional neural network. *arXiv preprint arXiv:2001.06268*, 2020. 1

[19] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. MixUp: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[20] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 3, 4

[21] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019. 1, 3

[22] Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In *International Conference on Machine Learning (ICML)*, pages 2280–2289, 2019. 2, 4

[23] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C. Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, volume 31, pages 5334–5344, 2018. 2, 4

[24] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Advances in Neural Information Processing Systems*, volume 33, 2020. 2, 4

[25] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. *arXiv preprint arXiv:1905.00877*, 2019. 2

[26] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning (ICML)*, pages 11278–11287, 2020. 2, 5, 8

[27] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 3, 4

[28] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv:1806.00451*, 2018. 3, 9

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009. 3, 6

[30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 3, 6

[31] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 3

[32] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 3

[33] Keren Gu, Brandon Yang, Jiquan Ngiam, Quoc Le, and Jonathon Shlens. Using videos to evaluate image model robustness. *arXiv preprint arXiv:1904.10076*, 2019.

[34] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? *arXiv preprint arXiv:1906.02168*, 2019.

[35] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[36] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 3

[37] 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. *arXiv preprint arXiv:1811.12231*, 2018. 3

[38] Alexey Dosovitskiy and Josip Djolonga. You only train once: Loss-conditional training of deep networks. In *International Conference on Learning Representations (ICLR)*, 2020. 4

[39] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. 4

[40] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015. 4

[41] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[42] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[43] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 4

[44] Michał Zając, Konrad Żołna, and Stanisław Jastrzębski. Split batch normalization: Improving semi-supervised learning under domain shift. *arXiv preprint arXiv:1904.03515*, 2019. 4

[45] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training. *arXiv preprint arXiv:1906.03787*, 2019. 4, 6, 8

[46] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. *Advances in Neural Information Processing Systems*, 31:2558–2567, 2018. 4

[47] Xiangru Lian and Ji Liu. Revisit batch normalization: New understanding and refinement via composition optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3254–3263. PMLR, 2019. 6

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[49] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, pages 87.1–87.12, 2016. 6

[50] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017. 6

[51] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018. 7

[52] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019. 9

[53] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018. 9

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not see any potential negative impacts of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Codes and pretrained models will be released upon acceptance.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 8.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.1.

(b) Did you mention the license of the assets? [No] All datasets used in this paper are public with appropriate licenses.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] All datasets used in this paper are public available.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] To the best of our knowledge, the datasets used in this paper do not contain any personally identifiable information.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]