Adversarially Perturbed Batch Normalization: A Simple Way to Improve Image Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recently, it has been shown that adversarial training (AT) by injecting adversarial samples can improve the quality of recognition. However, the existing AT methods suffer from the performance degradation on the benign samples, leading to a gap between robustness and generalization. We argue that this gap is caused by the inaccurate estimation of the Batch Normalization (BN) layer, due to the distributional discrepancy between the training and test set. To bridge this gap, this paper identifies the adversarial robustness against the indispensable noise in BN statistics. In particular, we proposed a novel strategy that adversarially perturbs the BN layer, termed ARAPT. The ARAPT leverages the gradients to shift BN statistics and helps models resist the shifted statistics to enhance robustness to noise. Then, we introduce ARAPT into a new paradigm of AT called model-based AT, which strengthens models' tolerance to noise in BN. Experiments indicate that the APART can improve model generalization, leading to significant improvements in accuracy on benchmarks like CIFAR-10, CIFAR-100, Tiny-ImageNet, and ImageNet.

1 Introduction

Recent works [1, 2, 3, 4] show that deep neural networks are sensitive to adversarial perturbations, which gives rise to the rapid development of adversarial training (AT) methods [5, 6, 7, 8]. These AT methods enhance models' robustness against adversarial samples by solving a min-max optimization problem [5]. However, many efforts [6, 9, 10] have corroborated that there is a trade-off between standard and robust accuracy, in which AT usually degrades models' performance on benign samples, though they enjoy the accuracy gains on adversarial samples.

Xie *et al*. [11] challenges the widely accepted idea that AT hurts models' generalization. They proposed adversarial propagation (AdvProp) to exploit the adversarial features via auxiliary Batch Normalization (BN) [12] layers. However, the huge computational overhead discourages more efforts in its applications. Thus, the further work proposed fast AdvProp [13] to reduce the computations by leveraging the acceleration of AT [14]. Nevertheless, leveraging adversarial samples to perform AT in *non-safety* tasks leads to the following question:

What does the model defend against?

Indeed, adversarially trained models appear closely tied to the robustness against adversarial attacks for safety concerns. However, in non-safety situations, there is an open question of what these models defend against. This question is essentially related to the gap between models' generalization and adversarial robustness. Prior efforts [15, 16, 17] reposition the adversarial robustness as the robustness against the worst-case unseen domains, in attempts to bridge the generalization-robustness gap. They usually enhance models' robustness to adversarially generated domains to improve the generalization. Nonetheless, there is inevitably a mismatch between such generated domains and the actual domains. The mismatch hinders their further applications.

In this paper, we answer this question by identifying the robustness against the noise of BN statistics that are the estimated mean and variance. The statistics noise is indispensable due to the distributional discrepancy between training and test domains [18, 19, 20, 21, 22, 23]. Moreover, insufficient batch size will cause the severer noise in some computation-demanding tasks [18, 19, 20]. In this study, we cast the statistics noise as a numerical problem to avoid the issue of how to match the adversarially generated domains with the actual ones. Since the noise degrades BN's performance, numerically strengthening models' tolerance to such noise will boost the generalization of BN-based models.

In this work, we train models by **A**dversarially **P**erturbed b**A**tch no**R**maliza**T**ion (APART) that perturbs BN statistics and updates the model parameters to resist the perturbation on the fly. More concretely, APART performs backward passes twice over each batch of benign samples. The first backward pass produces two gradient computations: one is normal gradient that helps update parameters of model *w.r.t.* samples' patterns, and the other one is statistics gradient that is used to perturb the statistics parameters in BN. Then, the second pass is performed to generate the defensive gradient that helps the model resist the adversarial statistics perturbation. The normal and defensive gradients are combined to improve both generalization and robustness of the model. All gradients are computed by the regular gradient descent algorithm. Note that APART combines the normal gradient with the defensive one without changing the update strategy and without crafting the adversarial samples. This process follows a paradigm of AT performing attacks and defense within models instead of on samples, hence the name model-based AT. Besides, as suggested by AdvProp [11], the BN statistics computed over the adversarial passes are dropped to avoid the corruption.

Experimentally, APART makes models less brittle to noisy BN statistics. As a consequence, the models enjoy significant accuracy gains on CIFAR [24], Tiny-ImageNet [25] and ImageNet [26] datasets. Moreover, the improvement brought by APART only depends on BN, allowing the combination with other training methods, *e.g.* data augmentation [27] and sharpness-aware minimization [28].

Summary of contributions:

- We identify the adversarial robustness against the noise in BN statistics to bridge the gap between models' generalization and robustness. Enhancing such robustness by AT improves models' generalization on benign samples.
- We proposed APART to achieve the robustness against the statistics noise. APART follows a new paradigm of AT utilizing the gradients efficiently. By strengthening BN-based models' tolerance to BN statistics noise, APART significantly improves the models' performance.
- With its plug-and-play nature, APART allows the combination with other training methods and enjoys the further accuracy gains.

2 Related Work

2.1 Adversarial Training

Adversarial training (AT) [1, 2, 5, 29] is empirically demonstrated to be one of the most effective defense methods for models' safety concerns. Instead, many non-AT methods [30, 31, 32, 33] fail to defend against adaptive attacks [4]. However, AT sacrifices the standard accuracy on benign samples to increase models' robustness [9]. Thus, there is a trade-off between the robustness and generalization [6]. Furthermore, many efforts [34, 9, 35, 36, 37] theoretically and experimentally corroborate the difficulty of achieving adversarial robustness over limited data. Besides adversarial robustness, other works [38, 14] focus on the efficiency of AT due to the high computational overhead of vanilla AT methods [5, 6]. The proposed fast AT method [14] accelerates the training in a simple way, but suffers from catastrophic overfitting [39, 40]. This problem gives rise to more efforts [39, 40, 41, 42].

Besides performing AT over samples, Adversarial Weight Perturbation [8] additionally perturbs parameters to enhance the generalization from a perspective of loss landscape. In non-safety tasks, some efforts [28, 43, 44, 45] have been devoted to such parameter-based AT and show promise in improving models' generalization. In this study, the proposed method follows a more generic paradigm of AT that allows the attacks on each component of models even including the non-parameter BN statistics.

2.2 Adversarial Robustness Beyond Safety

Though the disadvantage of AT in models' generalization discourages the efforts of its non-safety applications, Xie *et al.* [11] proposed AdvProp to challenge this issue. AdvProp utilizes auxiliary BN layers to avoid corrupting the BN statistics estimated over benign samples. In this manner, AdvProp improves models' generalization and inspires the further studies [46, 16, 47, 15, 48, 49] of the adversarial robustness beyond safety. Indeed, AT provides the framework of crafting and countering the worse-case unseen domains [15, 16, 17], and enhances adversarial robustness varying in different contexts. Besides, Mei *et al.* [13] utilize the acceleration of fast AT [14] to significantly reduce the computational overhead of AdvProp [11].

In this work, the proposed AT method, termed APART, increases models' robustness against the noise of BN statistics. Though the perturbation formula of the statistics is somewhat similar to that of Adversarial Batch Normalization (AdvBN) [15], there are three major differences between them in implementations: 1) APART perturbs the entire network by slightly changing each BN layer, instead of perturbing the features generated from a specific non-BN layer [15]; 2) in each iteration, APART performs backward passes only twice to carry out the attack and defense efficiently, instead of performing multiple backward passes inefficiently [15]; 3) APART trains each model from the scratch, instead of fine-tuning a pre-trained model [15], which leads to incomparability between APART and AdvBN.

2.3 Normalization

Batch Normalization (BN) [12] has successfully boosted a broad range of deep neural networks by accelerating the training. However, the noisy statistics of BN degrade its performance experimentally [20] and theoretically [50]. Many efforts have been devoted to more accurate estimators of the statistics [18, 19, 20, 21, 22, 23]. Some estimators perform the normalization along different axes, *e.g.* Layer Normalization [18], Instance Normalization [19] and Group Normalization [20]. They reduce the noise in the case of tiny batch but suffer from performance degradation under large batch as the alternative to BN. More efforts [21, 51, 52, 16] exploit the combination of these normalization methods. They selectively use the axis-specific statistics to perform normalization in response to different domains. Additionally, the on-the-fly estimation of BN statistics over adversarial samples is experimentally found to have negative impacts on the standard accuracy [11, 53]. This finding leads to more exploration in BN under AT [11, 13, 15, 46, 49].

The noisy statistics result from a mismatch between the seen and unseen domains and are therefore indispensable without the domain-specific knowledge. Furthermore, tiny batch size caused by the computation-demanding tasks results in the severer noise. From an opposite perspective of these methods denoising the statistics, our method hardens BN-based models' robustness against the noise.

3 Method

In this section, we firstly introduce a new paradigm of AT that allows us to perform attacks and defense within models rather than on samples. Then, we propose APART to implement this paradigm in a simple way. Finally, we discuss the enhancement of APART, which is derived from the potential link between APART and the other training method.

3.1 Model-Based Adversarial Training

The vanilla AT formulates a min-max game [5] by adversarially crafting and defensively countering the imperceptible perturbations to samples. Specifically, given the ground truth $\boldsymbol{y}$ and sample $\boldsymbol{x}$'s allowed neighborhood $\mathcal{S}(\boldsymbol{x})$, we minimize the expectation of a $\boldsymbol{\theta}$-parameterized loss $\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}; \boldsymbol{\theta})$ with $\boldsymbol{x}^* \in \mathcal{S}(\boldsymbol{x})$ maximizing $\mathcal{L}(\cdot, \boldsymbol{y}; \boldsymbol{\theta})$, *i.e.*,

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}; \boldsymbol{\theta}), \quad \text{where} \quad \boldsymbol{x}^* := \underset{\boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x})}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{x}', \boldsymbol{y}; \boldsymbol{\theta}). \tag{1}$$

Empirically, the maximization of Eq. (1) is achieved by a gradient ascent method for each sample. The gradient $\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ is iteratively computed by full forward and backward passes on the model. This process merely requires the inputs' gradients $\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ and drops all the internal gradients $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ without their further utilization after finishing the backward pass. Thus, such vanilla AT suffers from low efficiency of utilizing the gradients. Meanwhile, AT's potency is limited by such sample-based attacks and defense. Therefore, we propose a paradigm of *model-based* AT to leverage internal gradients efficiently and allow the attacks and defense within models.

To perform such AT, each component of a model is categorized into two types: one for the attacker, and one for the defender. Denoting by $\boldsymbol{\theta}, \boldsymbol{\phi}$ the parameters of the adversarial and defensive components respectively, we formulate the model-based AT as follows

$$\min_{\boldsymbol{\phi}} \mathbb{E}_{(\boldsymbol{x},\mathbf{y})} \left[\mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi}) + \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi}) \right], \tag{2}$$

where $\boldsymbol{\Theta}$ is a parameter space that can be bounded to avoid trivial results; $\mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi})$ is a task-specific loss allowing models to learn the normal patterns in samples, and $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi})$ can share a similar formulation of $\mathcal{R}$ to enable more pattern exploration in an adversarial manner. Overall, Eq. (2) provides a generic formulation of model-based AT. For example, Generative Adversarial Networks (GANs) [54] can be repositioned as a special case of such AT, in which the generator and discriminator are regarded as the attacker and defender respectively, and the discriminative losses are cast as proper $\mathcal{R}$ and $\mathcal{L}$ in Eq. (2). Next, we introduce APART to implement this model-based AT in a simple way.

3.2 Adversarially Perturbed Batch Normalization

Model-based AT helps models harden their robustness against a specific problem. We shift our attention to the noisy BN statistics [20], and apply the proposed AT to address this problem.

Firstly, we embed two temporary parameters $\boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}}$ into each BN layer as the adversarial parameters $\boldsymbol{\theta}$ in Eq. (2), which will be dropped after the training. Inspired by AdvBN [15], with $\boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}} \leftarrow \mathbf{0}$, we reformulate the BN mapping as

$$\mathbf{BN}(\boldsymbol{x}; \boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}}) = \boldsymbol{\gamma}(\mathbf{1} + \boldsymbol{\delta}_{\boldsymbol{\sigma}}) \cdot \frac{\boldsymbol{x} - (\mathbf{1} + \boldsymbol{\delta}_{\boldsymbol{\mu}})\hat{\boldsymbol{\mu}}}{\hat{\boldsymbol{\sigma}}} + \boldsymbol{\beta}, \tag{3}$$

where each operator is element-wise; $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}$ are the mean and standard deviation estimated over a batch of $\boldsymbol{x}$ respectively; $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are the parameters of BN's affine mapping. We bound the d-dimensional $\boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}}$ such that $\boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}} \in [-\epsilon, \epsilon]^d$ for a small sufficient perturbation radius $\epsilon > 0$. The bound avoids trivial results, *e.g.* $\mathbf{BN}(\boldsymbol{x}; \boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}})|_{\boldsymbol{\delta}_{\boldsymbol{\sigma}}=-\mathbf{1}} \equiv \mathbf{0}$. Now all the other trainable parameters within the entire model including $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are naturally the defensive parameters $\boldsymbol{\phi}$. The losses $\mathcal{R}, \mathcal{L}$ in Eq. (2) are both the same task-specific loss, *i.e.*, the cross entropy in recognition. Therefore, Eq. (3) implies ($\boldsymbol{\theta}$ indicates all BN layers' $\boldsymbol{\delta}_{\boldsymbol{\mu}}, \boldsymbol{\delta}_{\boldsymbol{\sigma}}$)

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\mathbf{0}} = \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi}) \quad \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\mathbf{0}} = \nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi}), \tag{4}$$

by which we can get both $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\mathbf{0}}$ and $\nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi})$ in a single backward pass with $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\mathbf{0}}$ as the loss.

Secondly, we propose APART that follows a gradient accumulation strategy, instead of the alternative update strategy used by GANs [54]. Specifically, in each iteration of a normal gradient descent algorithm over a batch of samples $\mathcal{D} := \{(\boldsymbol{x}_i, \boldsymbol{y}_i), 1 \leq i \leq M\}$,

Algorithm 1: Pseudo code of APART getting the gradient for a batch of samples, given some perturbation radius ϵ, number of samples in the second pass N, and group number n

Data: A batch of samples $\mathcal{D} := \{(\boldsymbol{x}_i, \boldsymbol{y}_i), 1 \leq i \leq M\}$
Result: The gradient $\boldsymbol{g}$ for this batch of samples

$\boldsymbol{\theta} \leftarrow \mathbf{0}$
Perform forward and backward passes once over $\mathcal{D}$, generating $\boldsymbol{g}_{\boldsymbol{\theta}}$ and $\boldsymbol{g}_{\boldsymbol{\phi}}$ simultaneously
    $\boldsymbol{g}_{\boldsymbol{\theta}} \leftarrow \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\mathbf{0}}$
    $\boldsymbol{g}_{\boldsymbol{\phi}} \leftarrow \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}} \nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\phi})$
$\boldsymbol{\theta} \leftarrow \epsilon \text{sign}(\boldsymbol{g}_{\boldsymbol{\theta}})$
Randomly draw N samples $\mathcal{S} \subseteq \mathcal{D}$
Group $\mathcal{S}$ into n equally sized subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$
$\boldsymbol{h}_{\boldsymbol{\phi}} \leftarrow \mathbf{0}$
for $j \leftarrow 1$ **to** n **do**
    $\boldsymbol{h}_{\boldsymbol{\phi}} \leftarrow \boldsymbol{h}_{\boldsymbol{\phi}} + \frac{1}{n} \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{S}_j} \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\epsilon \text{sign}(\boldsymbol{g}_{\boldsymbol{\theta}})}$
end
$\boldsymbol{g} \leftarrow \frac{M}{M+N} \boldsymbol{g}_{\boldsymbol{\phi}} + \frac{N}{M+N} \boldsymbol{h}_{\boldsymbol{\phi}}$

- **Step 1:** With $\boldsymbol{\theta} \leftarrow \mathbf{0}$, APART performs the forward and backward passes once over this batch of samples to generate the gradients *w.r.t.* the adversarial and defensive parameters, *i.e.*, $\boldsymbol{g}_{\boldsymbol{\theta}} := \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\mathbf{0}}$ and $\boldsymbol{g}_{\boldsymbol{\phi}} := \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}} \nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\phi})$ according to Eq. (4). Like Fast Gradient Sign Method [2], APART uses $\epsilon \text{sign}(\boldsymbol{g}_{\boldsymbol{\theta}})$ to assign $\boldsymbol{\theta}$, which empirically performs the inner maximization of Eq. (2) and generates the adversarially perturbed statistics in each BN layer.

- **Step 2:** With the adversarial BN statistics, APART performs the forward and backward passes again, over a full/incomplete batch of the same samples $\mathcal{S} \subseteq \mathcal{D}$. This backward pass yields the gradient resisting the attack, *i.e.*, $\boldsymbol{h}_{\boldsymbol{\phi}} := \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{S}} \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}, \boldsymbol{\phi})|_{\boldsymbol{\theta}=\epsilon \text{sign}(\boldsymbol{g}_{\boldsymbol{\theta}})}$. The weighted gradient $\boldsymbol{g} := (1-r)\boldsymbol{g}_{\boldsymbol{\phi}} + r\boldsymbol{h}_{\boldsymbol{\phi}}$ is finally used in the outer minimization of Eq. (2) for this batch of samples, where $r \in [0, 0.5]$ re-balances the gradients.

Apparently, using a full batch of the samples in the second pass leads to the best performance, but results in more computational overhead. Instead, using the incomplete batch of these samples allows the less computation but suffers from insufficient defense against the attack. Thus, the ratio $r = N/(M+N)$ is introduced to re-balance the gradients with $N = |\mathcal{S}|$ the number of the samples used in the second pass. Additionally, the on-the-fly BN statistics estimated in the second pass are completely dropped to avoid corrupting the statistics at inference, like the auxiliary BN layers [11].

Note that stronger attacks in AT indirectly enhance the adversarial robustness [5]. Thus, we slightly modify the process of the second pass to strengthen the attack. The modification increases the noise in the adversarial BN statistics without additional computation. In details, we group the samples into equally sized sets and stop their group-to-group communications in BN layers during the second forward pass. This is inspired by the fact that smaller batch size results in larger noise in the statistics. In this manner, the BN statistics are estimated over less samples without reducing the entire batch size, giving rise to the less adversarial accuracy.

Overall, APART only changes the way of getting gradients in each iteration, without involving in data augmentation or network modification. Therefore, APART has plug-and-play nature that allows the combination with a broad range of training methods. We summarize APART in Algorithm 1, and then introduce the enhancement of APART.

3.3 Enhancement by Combination with Sharpness-Aware Minimization

Besides APART, Sharpness-Aware Minimization (SAM) [28] also belongs to and use the proposed model-based AT paradigm. SAM improves network training from a perspective of loss landscape relating to models' generalization. Given the empirical loss function $\mathcal{L}_{\mathcal{D}}$ estimated over a dataset $\mathcal{D}$, SAM minimizes $\mathcal{L}_\mathcal{D}$ with a sharpness measure:

$$\min_{\boldsymbol{w}} \left[\max_{||\boldsymbol{\delta}||_2 \leq \rho} \mathcal{L}_\mathcal{D}(\boldsymbol{w}+\boldsymbol{\delta}) - \mathcal{L}_\mathcal{D}(\boldsymbol{w}) \right] + \mathcal{L}_\mathcal{D}(\boldsymbol{w}) + \frac{\lambda}{2}||\boldsymbol{w}||_2^2, \tag{5}$$

where $\boldsymbol{w}$ is the model's trainable parameters; $\rho > 0$ is small sufficient to restrict the perturbation $\boldsymbol{\delta}$; $\lambda > 0$ is used to control the regularizer $||\boldsymbol{w}||_2^2$; the term in the square bracket measures $\mathcal{L}_\mathcal{D}$'s sharpness. Obviously, Eq. (5) is equivalent to

$$\min_{\boldsymbol{w}} \max_{||\boldsymbol{\delta}||_2 \leq \rho} \mathcal{L}_\mathcal{D}(\boldsymbol{w}+\boldsymbol{\delta}) + \frac{\lambda}{2}||\boldsymbol{w}||_2^2. \tag{6}$$

Actually, Eq.(6) performs the model-based AT formulated by Eq. (2), in which we treat $\boldsymbol{\delta}, \boldsymbol{w}$ as the adversarial and defensive parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$ respectively, and let $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}_\mathcal{D}(\boldsymbol{w} + \boldsymbol{\delta}), \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi}) \equiv 0$ with the regularizer $\frac{\lambda}{2}||\boldsymbol{w}||_2^2$ included in the empirical optimization. Therefore, SAM and APART essentially share the same training paradigm. Surprisingly, these two methods implement this paradigm in a complementary way: SAM focuses on the trainable parameters optimized by gradient descent, while APART concentrates on the non-trainable BN statistics requiring estimation instead of optimization. The training paradigm enables them to enhance models' robustness in different contexts, which inspires a combination of them.

Note that the inner maximization in Eq. (6) is approximately achieved by one-step gradient ascent. Then, the outer minimization is performed by estimating the gradient *w.r.t.* the adversarially shifted parameters $\boldsymbol{w} \leftarrow \boldsymbol{w}+\boldsymbol{\delta}$ [28]. Thus, SAM shares a similar two-step strategy of APART. Such similarity allows us to perform APART and SAM simultaneously by a slight modification of Algorithm 1, termed APART-SAM. Specifically, we adopt the weights' perturbations, *i.e.*, Eq. (2) therein [28], reformulated as

$$\hat{\boldsymbol{\delta}}(\boldsymbol{\phi}) = \rho \text{sign}\left(\nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi})\right) |\nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi})|^{q-1} / \left(||\nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi})||_q^q\right)^{1/p}, \tag{7}$$

where $1/p + 1/q = 1$ and experimentally let $p = 2$ as suggested by [28]. Then, we modify APART's first step by additionally perturbing the defensive parameters $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \hat{\boldsymbol{\delta}}(\boldsymbol{\phi})$ to enhance the attacks with the second step unchanged. The additional perturbation $\hat{\boldsymbol{\delta}}(\boldsymbol{\phi})$ just employs the gradient $\nabla_{\boldsymbol{\phi}} \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\phi})$ previously computed by APART's first step, and normalizes them with ignorable extra computations. Therefore, APART-SAM is computation-friendly enhancement of APART.

4 Experiments

4.1 Experimental Setup

Datasets and Models. We evaluate APART on CIFAR-10, CIFAR-100 [24], Tiny-ImageNet [25] and ImageNet [26]. On CIFAR datasets, we employ a WideResNet-40-2 [55] (as implemented in [56]), PreAct-ResNet-18 [57] (as implemented in [58]). We use a PreAct-ResNet-18 on Tiny-ImageNet, and ResNet-18 [59] (as implemented in *torchvision* library [60]) on ImageNet.

Implementation Details. On CIFAR-10 and CIFAR-100, we run all experiments by five different random seeds and report the mean and standard derivation of test accuracy. We employ SGD with initial learning rate 0.1, momentum 0.9 and weight decay 0.0005. We train models for 200 epochs and reduce the learning rate by 0.1 at the 100-th and 150-th epoch with batch size of 128. We use only the standard augmentations (*i.e.*, random flipping and translation) in the basic experiments, and additionally leverage mixup [27] for further comparison. The hyperparameter α of mixup is set to 1 in the baseline as suggested by [27] and is properly chosen for APART. For comparison, we report the empirical results of SAM-trained WideResNet-40-2 and PreAct-ResNet-18 under standard augmentation, where we set $\rho = 0.05$ and $\rho = 0.1$ on CIFAR-10 and CIFAR-100 respectively, as suggested by [28]. On Tiny-ImageNet, we use batch size of 256 and set other hyperparameters in the same way of the CIFAR experiments; under mixup, we set $\alpha = 0.2$ for both the standard method and APART. On ImageNet, We employ SGD with initial learning rate 0.1, momentum 0.9 and weight decay 0.0001. We train models with batch size of 256 for 105 epochs, where the learning rate is reduced by 0.1 at the 30-th, 60-th, 90-th and 100-th epoch. We randomly resize and crop images to 224×224 resolution with random flipping to perform the standard augmentation. For the hyperparameters of APART and APART-SAM, we evaluate a few combinations and show the best performance in the main results. More results of these combinations are reported in the ablation studies and appendix A.1. Considering the $2\times$ training budget of APART, we also conduct the experiments of the standard training with $2\times$ total and decay epochs to show APART's non-trivial performance. Our implementations use Pytorch [61], and all models are trained on a server with three NVIDIA RTX 3090 GPUs. Please see appendix B and the code at `https://github.com/unknown9567/apart.git` for more details.

4.2 Main Results

Table 1: Results on CIFAR-10 and CIFAR-100.

Method (Augmentation)	Budget	CIFAR-10	CIFAR-100
WideResNet-40-2			
Standard (Standard)	1×	$94.67_{\pm0.10}(+0.00)$	$76.10_{\pm0.24}(+0.00)$
Standard (Standard)	2×	$94.99_{\pm0.11}(+0.32)$	$76.73_{\pm0.27}(+0.63)$
SAM (Standard)	2×	$95.39_{\pm0.14}(+0.72)$	$77.47_{\pm0.09}(+1.37)$
APART (Standard)	2×	$95.69_{\pm0.13}(+1.02)$	$79.05_{\pm0.25}(+2.95)$
APART-SAM (Standard)	2×	$\mathbf{95.81_{\pm0.27}(+1.14)}$	$\mathbf{79.21_{\pm0.23}(+3.11)}$
Standard (Mixup)	1×	$95.43_{\pm0.11}(+0.76)$	$76.63_{\pm0.34}(+0.53)$
Standard (Mixup)	2×	$\mathbf{96.03_{\pm0.11}(+1.36)}$	$77.96_{\pm0.43}(+1.86)$
APART (Mixup)	2×	$95.86_{\pm0.05}(+1.19)$	$\mathbf{79.22_{\pm0.22}(+3.12)}$
APART-SAM (Mixup)	2×	$95.78_{\pm0.08}(+1.11)$	$79.00_{\pm0.09}(+2.90)$
PreAct-ResNet-18			
Standard (Standard)	1×	$94.60_{\pm0.17}(+0.00)$	$76.30_{\pm0.11}(+0.00)$
Standard (Standard)	2×	$94.76_{\pm0.12}(+0.16)$	$75.34_{\pm0.21}(-0.96)$
SAM (Standard)	2×	$95.56_{\pm0.16}(+0.96)$	$78.57_{\pm0.17}(+2.27)$
APART (Standard)	2×	$95.84_{\pm0.16}(+1.24)$	$79.48_{\pm0.15}(+3.18)$
APART-SAM (Standard)	2×	$\mathbf{96.12_{\pm0.06}(+1.52)}$	$\mathbf{80.07_{\pm0.18}(+3.77)}$
Standard (Mixup)	1×	$95.76_{\pm0.11}(+1.16)$	$77.30_{\pm0.50}(+1.00)$
Standard (Mixup)	2×	$96.19_{\pm0.12}(+1.59)$	$78.81_{\pm0.45}(+2.51)$
APART (Mixup)	2×	$\mathbf{96.28_{\pm0.09}(+1.68)}$	$80.07_{\pm0.17}(+3.77)$
APART-SAM (Mixup)	2×	$96.08_{\pm0.18}(+1.48)$	$\mathbf{80.19_{\pm0.15}(+3.89)}$

Table 2: Results on Tiny-ImageNet and ImageNet.

Method (Augmentation)	Budget	Accuracy (%)
Tiny-ImageNet		
Standard (Standard)	1×	63.52 (+0.00)
Standard (Mixup)	1×	64.34 (+0.82)
Standard (Standard)	2×	63.94 (+0.42)
Standard (Mixup)	2×	64.54 (+1.02)
APART (Standard)	2×	67.00 (+3.48)
APART (Mixup)	2×	67.26 (+3.74)
APART-SAM (Standard)	2×	67.53 (+4.01)
APART-SAM (Mixup)	2×	**68.66 (+5.14)**
ImageNet		
Standard (Standard)	1×	70.24 (+0.00)
Standard (Standard)	2×	71.25 (+1.01)
Standard (Standard)	4×	71.45 (+1.21)
APART (Standard)	2×	70.86 (+0.62)
APART (Standard)	4×	**72.14 (+1.90)**
APART-SAM (Standard)	2×	70.82 (+0.58)

Evaluation on CIFAR-10 and CIFAR-100. As is shown in Table 1, APART helps models significantly outperform their counterparts trained by the standard method. Under standard augmentation, without considering the training budget, APART improves the accuracy by over 1.02% on CIFAR-10 and 2.95% on CIFAR-100 for each model; considering the training budget leads to the accuracy gains of over 0.70% on CIFAR-10 and 2.32% on CIFAR-100; enhanced by SAM, APART-SAM further improves the accuracy of APART-trained models by over 0.12% on CIFAR-10 and 0.16% on CIFAR-100. In addition, APART and APART-SAM outperform SAM under this experimental setting. Under mixup [27], the improvements achieved by APART are generally consistent, though the APART-trained WideResNet-40-2 is somewhat inferior to the standard counterpart with $2\times$ budget on CIFAR-10; on the other hand, APART-SAM slightly degenerates due to the potential conflicts between SAM and mixup on CIFAR datasets. Besides, mixup with sufficient training budgets boosts the standard models more significantly, reducing the accuracy gap between them and the APART-trained counterparts.

Evaluation on Tiny-ImageNet and ImageNet. As is shown in Table 2, APART and APART-SAM consistently improve the accuracy on ImageNet and its variant. On Tiny-ImageNet, the accuracy gains are significant, *e.g.* the models trained by APART-SAM outperforms the standard counterparts by over 4% and 3.5% for $1\times$ and $2\times$ training budgets respectively under standard augmentation. Furthermore, APART-SAM enjoys the combination with mixup and improves the accuracy by over 5%. On ImageNet, APART with $2\times$ budget outperforms the baseline with $1\times$ budget, but is inferior to the standard training with $2\times$ budget. However, scaling the training budgets leads to a different result: APART with $4\times$ budget outperforms the standard method with both $2\times$ and $4\times$ training budgets. It seems that APART employed on the large-scale dataset requires more steps to show its promise. Besides, APART-SAM slightly degenerates due to the insufficient tuning of its more hyperparameters.

4.3 Ablation Study

Table 3 shows the performance of the WideResNet-40-2 trained by APART with different hyperparameters in Algorithm 1 on CIFAR-100. Overall, APART-trained models outperform all standard models despite training budgets and hyperparameters. For example, even the model trained by APART with $1.19\times$ budget performs better than the standard model with $2.00\times$ budget. On the other hand, APART's hyperparameters have impacts at different levels on its performance.

Table 3: Ablation studies of APART's hyperparameters.

Budget	N	ϵ	n	**Accuracy (%)**
Standard Training				
$1.00\times$	NA	NA	NA	$76.10_{\pm 0.24}(+0.00)$
$1.20\times$				$76.24_{\pm 0.30}(+0.14)$
$2.00\times$				$76.73_{\pm 0.27}(+0.63)$
APART				
$1.19\times$	24	0.1	1	$77.58_{\pm 0.17}(+1.48)$
		0.1	2	$77.50_{\pm 0.17}(+1.40)$
		0.1	8	$77.35_{\pm 0.36}(+1.25)$
$2.00\times$	128	0.05	8	$78.45_{\pm 0.12}(+2.35)$
		0.1	8	$78.80_{\pm 0.23}(+2.70)$
		0.2	8	$77.95_{\pm 0.30}(+1.85)$
		0.4	8	$71.86_{\pm 0.12}(-4.24)$
		0.1	1	$78.36_{\pm 0.22}(+2.26)$
		0.1	2	$78.54_{\pm 0.39}(+2.44)$
		0.1	16	$79.05_{\pm 0.25}(+2.95)$
		0.1	32	$78.69_{\pm 0.19}(+2.59)$

Impact of N. The number of samples used in APART's second step has a significant impact on its performance. Indeed, models with APART's adversarial BN statistics implicitly generate adversarial features within the models in the second pass. Therefore, more samples in this pass lead to more diversity required by models' robustness against the noisy BN statistics and improve the performance more significantly.

Impact of ϵ. Large perturbation radii (*e.g.* $\epsilon = 0.4$) degenerate models' performance, since strong attacks caused by such radii force the models to sacrifice their generalization for more robustness. In contrast, smaller radii reduce both the robustness and accuracy, which illustrates the link between the generalization and robustness of APART-trained models.

Impact of n. The group number has a relatively slight impact on the accuracy, since it implicitly enhances the attack. A properly chosen n can help APART achieve the best performance.

4.4 Evaluation on APART's Attacks

Experimental Setup. We evaluate APART's attacks to provide a basic insight of its effectiveness. We use a WideResNet-40-2 [55] pretrained on CIFAR-100. We perform APART's first step to adversarially shift its BN statistics without changing the other parameters. We use only a batch of training samples for the attack, but evaluate the accuracy over the entire training dataset. For comparison, we provides the accuracy in the cases of random perturbations, *i.e.*, $\boldsymbol{\delta_\mu}, \boldsymbol{\delta_\sigma} \sim \text{Uniform}[-\epsilon, \epsilon]^d$ or

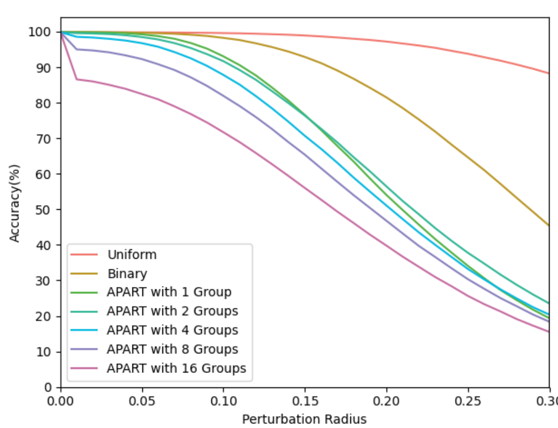

Figure 1: Evaluation on APART's Attacks.

randomly drawing $\boldsymbol{\delta_\mu}, \boldsymbol{\delta_\sigma}$ from $\{-\epsilon, \epsilon\}^d$ formed by the binary values. Besides, we test different group numbers of APART to substantiate our insight of this trick.

Results. As is shown in Figure 1, the uniform random perturbations result in almost no reduction in the accuracy despite the radii, while the binary random perturbations require sufficient large radii for the attack. In contrast, APART uses only a batch of samples to generate the effective perturbations that reduce the accuracy even under a small radius. Additionally, the larger group numbers of APART provide more significant accuracy reduction especially when the radii are more limited, demonstrating our insight.

4.5 Robustness against Perturbed BN Statistics

Experimental Setup. We evaluate the robustness of the APART-trained models against perturbed BN statistics to provide the insight of APART's effectiveness. We employ the WideResNet-40-2 trained by the standard method with $1\times$ and $2\times$ training budgets and APART with different perturbation radii and group numbers on CIFAR-100. First, we randomly draw a direction $\boldsymbol{v}$ from $\{-1, 1\}^d$ for each BN statistics with the same initial random seed shared across each experiment. Second, we scale $\boldsymbol{v}$ by different perturbation radii ϵ to perturb the estimated BN statistics, *i.e.*, $\hat{\boldsymbol{\mu}} \leftarrow (1 + \epsilon \boldsymbol{v})\hat{\boldsymbol{\mu}}$ or $\hat{\boldsymbol{\sigma}} \leftarrow (1 + \epsilon \boldsymbol{v})\hat{\boldsymbol{\sigma}}$. Then, each model with the perturbed statistics is evaluated over the test samples.

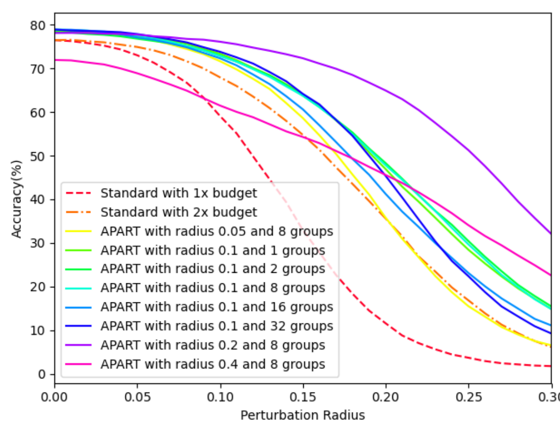

Figure 2: Robustness of WideResNet-40-2 against perturbed BN statistics on CIFAR-100.

Results. As is shown in Figure 2, models' generalization is measured by the non-perturbed accuracy, and their robustness is illustrated by the accuracy reduction resulting from the perturbations. APART-trained models generally outperform the standard models for both the generalization and robustness. Specifically, the standard models (dashed lines) yield lower non-perturbed accuracy and suffers from more accuracy reduction as the perturbations increase. Meanwhile, more training epochs (dashed orange line) slightly improve the performance of the standard methods. On the other hand, APART performs better but requires a trade-off between the generalization and robustness. Increasing APART's radii improves both the robustness and generalization to some extent. However, a large radius ($\epsilon = 0.4$) results in the severe degeneration (solid pink line) in the generalization. Additionally, different group numbers of APART lead to improvement of the generalization and robustness to varying degrees but have no clear trend. In summary, APART consolidates models' robustness against noisy BN statistics to boost models' performance but requires a further generalization-robustness trade-off achieved by tuning the hyperparameters.

5 Conclusion and Discussion

In this paper, we identify the robustness against the noise in BN statistics to bridge the generalization-robustness gap. Then, we proposed APART that implements a new AT paradigm, termed model-based AT, to achieve such robustness. APART performs attacks and defense within models by two backward passes over each batch of benign samples, utilizing gradients efficiently. The empirical results demonstrate APART's effectiveness in improving the robustness, which further boosts model generalization on benign samples.

Limitations. Though APART improves models by solving a BN-specific problem, it and its variant suffer from the potential degeneration in case of the combination with other training methods implicitly involving BN, which results in more demand for fine-tuning the hyperparameters.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582, 2016.

[4] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On Adaptive Attacks to Adversarial Example Defenses. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual.*, 2020.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[6] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7472–7482. PMLR, May 2019.

[7] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[8] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial Weight Perturbation Helps Robust Generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 2958–2969. Curran Associates, Inc., 2020.

[9] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[10] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Adversarial Training Can Hurt Generalization. *arXiv:1906.06032*, August 2019.

[11] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial Examples Improve Image Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 816–825, 2020.

[12] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015.

[13] Jieru Mei, Yucheng Han, Yutong Bai, Yixiao Zhang, Yingwei Li, Xianhang Li, Alan Yuille, and Cihang Xie. Fast AdvProp. In *International Conference on Learning Representations*, 2021.

[14] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[15] Manli Shu, Zuxuan Wu, Micah Goldblum, and Tom Goldstein. Encoding robustness to image style via adversarial feature perturbations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[16] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[17] Tejas Gokhale, Rushil Anirudh, Bhavya Kailkhura, Jayaraman J. Thiagarajan, Chitta Baral, and Yezhou Yang. Attribute-Guided Adversarial Training for Robustness to Natural Perturbations. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7574–7582, 2021.

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

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

[20] Yuxin Wu and Kaiming He. Group Normalization. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, pages 3–19, 2018.

[21] Ping Luo, Jiamin Ren, Zhanglin Peng, Ruimao Zhang, and Jingyu Li. Differentiable Learning-to-Normalize via Switchable Normalization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[22] Saurabh Singh and Abhinav Shrivastava. EvalNorm: Estimating Batch Normalization Statistics for Evaluation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3632–3640, 2019.

[23] Guangrun Wang, Jiefeng Peng, Ping Luo, Xinjiang Wang, and Liang Lin. Kalman Normalization: Normalizing Internal Representations Across Network Layers. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*, pages 21–31, 2018.

[24] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[25] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A Downsampled Variant of ImageNet as an Alternative to the CIFAR datasets. *arXiv:1707.08819*, August 2017.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[27] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[28] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[29] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent Advances in Adversarial Training for Adversarial Robustness. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021.*, pages 4312–4321, 2021.

[30] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The Odds are Odd: A Statistical Test for Detecting Adversarial Examples. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5498–5507, 2019.

[31] Yingzhen Li, John Bradshaw, and Yash Sharma. Are Generative Classifiers More Robust to Adversarial Attacks? In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3804–3814, 2019.

[32] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup Inference: Better Exploiting Mixup to Defend Adversarial Attacks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[33] Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 7025–7034, 2019.

[34] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

[35] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially Robust Generalization Requires More Data. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*, pages 5019–5031, 2018.

[36] Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, pages 129–139, 2021.

[37] Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are Labels Required for Improving Adversarial Robustness? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*, pages 12192–12202, 2019.

[38] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*, pages 3353–3364, 2019.

[39] Maksym Andriushchenko and Nicolas Flammarion. Understanding and Improving Fast Adversarial Training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual.*, 2020.

[40] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding Catastrophic Overfitting in Single-step Adversarial Training. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8119–8127, 2021.

[41] B. S. Vivek and R. Venkatesh Babu. Single-Step Adversarial Training With Dropout Scheduling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 947–956, June 2020.

[42] BS Vivek and R Venkatesh Babu. Regularizers for single-step adversarial training. In *arXiv Preprint arXiv:2002.00614*, 2020.

[43] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event.*, pages 5905–5914, 2021.

[44] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, James s Duncan, Ting Liu, et al. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2021.

[45] Xu Sun, Zhiyuan Zhang, Xuancheng Ren, Ruixuan Luo, and Liangyou Li. Exploring the Vulnerability of Deep Neural Networks: A Study of Parameter Corruption. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11648–11656, 2021.

[46] Tianlong Chen, Yu Cheng, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zhangyang Wang, and Jingjing Liu. Adversarial feature augmentation and normalization for visual recognition. *arXiv preprint arXiv:2103.12171*, 2021.

[47] Amil Merchant, Barret Zoph, and Ekin Dogus Cubuk. Does data augmentation benefit from split batchnorms. *arXiv preprint arXiv:2010.07810*, 2020.

[48] Charles Herrmann, Kyle Sargent, Lu Jiang, Ramin Zabih, Huiwen Chang, Ce Liu, Dilip Krishnan, and Deqing Sun. Pyramid Adversarial Training Improves ViT Performance. *arXiv:2111.15121 [cs]*, November 2021.

[49] Xinyu Gong, Wuyang Chen, Tianlong Chen, and Zhangyang Wang. Sandwich Batch Normalization: A Drop-In Replacement for Feature Distribution Heterogeneity. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2957–2967, Waikoloa, HI, USA, January 2022. IEEE.

[50] You Huang and Yuanlong Yu. An Internal Covariate Shift Bounding Algorithm for Deep Neural Networks by Unitizing Layers' Outputs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8462–8470, 2020.

[51] Wenqi Shao, Tianjian Meng, Jingyu Li, Ruimao Zhang, Yudian Li, Xiaogang Wang, and Ping Luo. SSN: Learning Sparse Switchable Normalization via SparsestMax. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 443–451, 2019.

[52] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, pages 484–500, 2018.

[53] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[55] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *British Machine Vision Conference 2016*, York, France, 2016. British Machine Vision Association.

[56] Jason Kuen. URL https://github.com/xternalz/WideResNet-pytorch.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 630–645, 2016.

[58] Kuang Liu. URL https://github.com/kuangliu/pytorch-cifar.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[60] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1485–1488, 2010.

[61] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (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)? [Yes] See the supplemental materials
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the experimental setup
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(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]

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]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Only existing datasets are used.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]