Adversarial Examples Improve Image Recognition

Cihang Xie[1,2]* Mingxing Tan[1] Boqing Gong[1] Jiang Wang[1] Alan Yuille[2] Quoc V. Le[1]
[1]Google [2]Johns Hopkins University

Abstract

Adversarial examples are commonly viewed as a threat to ConvNets. Here we present an opposite perspective: adversarial examples can be used to ***improve image recognition models*** *if harnessed in the right manner. We propose AdvProp, an enhanced adversarial training scheme which treats adversarial examples as additional examples, to prevent overfitting. Key to our method is the usage of a separate auxiliary batch norm for adversarial examples, as they have different underlying distributions to normal examples.*

We show that AdvProp improves a wide range of models on various image recognition tasks and performs better when the models are bigger. For instance, by applying AdvProp to the latest EfficientNet-B7 [41] on ImageNet, we achieve significant improvements on ImageNet (+0.7%), ImageNet-C (+6.5%), ImageNet-A (+7.0%) and Stylized-ImageNet (+4.8%). With an enhanced EfficientNet-B8, our method achieves the state-of-the-art ***85.5%*** *ImageNet top-1 accuracy* ***without*** *extra data. This result even surpasses the best model in [24] which is trained with 3.5B Instagram images (∼3000× more than ImageNet) and ∼9.4× more parameters. Models are available at* https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet.

1. Introduction

Adversarial examples crafted by adding imperceptible perturbations to images, can lead Convolutional Neural Networks (ConvNets) to make wrong predictions. The existence of adversarial examples not only reveals the limited generalization ability of ConvNets, but also poses security threats on the real-world deployment of these models. Since the first discovery of the vulnerability of ConvNets to adversarial attacks [40], many efforts [2, 7, 15, 16, 18, 23, 29, 36, 42, 45, 50] have been made to improve network robustness.

In this paper, rather than focusing on defending against adversarial examples, we shift our attention to leveraging adversarial examples to improve accuracy. Previous works show that training with adversarial examples can enhance model generalization but are restricted to certain situations—the improvement is only observed either on small datasets (*e.g.*, MNIST) in the fully-supervised setting [7, 20], or on larger datasets but in the semi-supervised setting [26, 30]. Meanwhile, recent works [18, 16, 45] also suggest that training with adversarial examples on large datasets, *e.g.*, ImageNet [33], with supervised learning results in performance degradation on clean images. To summarize, it remains an open question of how adversarial examples can be used effectively to help vision models.

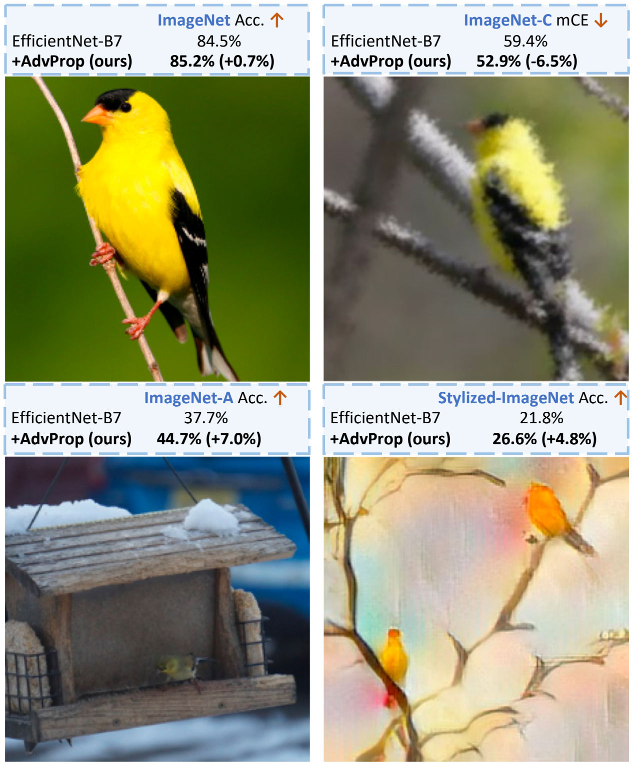


Figure 1. **AdvProp improves image recognition**. By training models on ImageNet, AdvProp helps EfficientNet-B7 [41] to achieve 85.2% accuracy on ImageNet [33], 52.9% mCE (mean corruption error, lower is better) on ImageNet-C [9], 44.7% accuracy on ImageNet-A [10] and 26.6% accuracy on Stylized-ImageNet [6], beating its vanilla counterpart by 0.7%, 6.5%, 7.0% and 4.8%, respectively. Theses sample images are randomly selected from the category "goldfinch".

*Work done during an internship at Google.

We observe all previous methods jointly train over clean images and adversarial examples without distinction even though they should be drawn from different underlying distributions. We hypothesize this distribution mismatch between clean examples and adversarial examples is a key factor that causes the performance degradation in previous works [16, 18, 45].

In this paper, we propose AdvProp, short for Adversarial Propagation, a new training scheme that bridges the distribution mismatch with a simple yet highly effective two-batchnorm approach. Specifically, we propose to use two batch norm statistics, one for clean images and one auxiliary for adversarial examples. The two batchnorms properly disentangle the two distributions at normalization layers for accurate statistics estimation. We show this distribution disentangling is crucial, enabling us to successfully improve, rather than degrade, model performance with adversarial examples.

To our best knowledge, our work is the first to show adversarial examples can improve model performance in the fully-supervised setting on the large-scale ImageNet dataset. For example, an EfficientNet-B7 [41] trained with AdvProp achieves 85.2% top-1 accuracy, beating its vanilla counterpart by 0.8%. The improvement by AdvProp is more notable when testing models on distorted images. As shown in Fig. 1, AdvProp helps EfficientNet-B7 to gain an absolute improvement of 9.0%, 7.0% and 5.0% on ImageNet-C [9], ImageNet-A [10] and Stylized-ImageNet [6], respectively.

As AdvProp effectively prevents overfitting and performs better with larger networks, we develop a larger network, named EfficientNet-B8, by following similar compound scaling rules in [41]. With our proposed AdvProp, EfficientNet-B8 achieves the state-of-the-art **85.5%** top-1 accuracy on ImageNet **without** any extra data. This result even surpasses the best model reported in [24], which is pretrained on 3.5B extra Instagram images (~3000× more than ImageNet) and requires ~9.4× more parameters than our EfficientNet-B8.

2. Related Work

Adversarial Training. Adversarial training, which trains networks with adversarial examples, constitutes the current foundation of state-of-the-arts for defending against adversarial attacks [7, 18, 23, 45]. Although adversarial training significantly improves model robustness, how to improve clean image accuracy with adversarial training is still under-explored. VAT [26] and deep co-training [30] attempt to utilize adversarial examples in semi-supervised settings, but they require enormous extra unlabeled images. Under supervised learning settings, adversarial training is typically considered hurting accuracy on clean images [32], *e.g.*, ~10% drop on CIFAR-10 [23] and ~15% drop on ImageNet [45]. Tsipras *et al.* [43] argue that the performance tradeoff between adversarial robustness and standard accuracy is provably inevitable, and attribute this phenomenon as a consequence of robust classifiers learning fundamentally different feature representations than standard classifiers. Other works try to explain this tradeoff phenomenon from the perspective of the increased sample complexity of adversary [37, 25, 28], the limited amount of training data [1, 27, 34, 44, 48], or network overparameterization [31].

This paper focuses on standard supervised learning without extra data. Although using similar adversarial training techniques, we stand on an opposite perspective to previous works—we aim at using adversarial examples to improve clean image recognition accuracy.

Benefits of Learning Adversarial Features. Many works corroborate that training with adversarial examples brings additional features to ConvNets. For example, compared with clean images, adversarial examples make network representations align better with salient data characteristics and human perception [43]. Moreover, such trained models are much more robust to high frequency noise [47]. Zhang *et al.* [51] further suggest these adversarially learned feature representations are less sensitive to texture distortions and focus more on shape information.

Our proposed AdvProp can be characterized as a training paradigm which fully exploits the complementarity between clean images and their corresponding adversarial examples. The results further suggest that adversarial features are indeed beneficial for recognition models, which agree with the conclusions drawn from these aforementioned studies.

Data augmentation. Data augmentation, which applies a set of label-preserving transformations to images, serves as an important and effective role to prevent networks from overfitting [17, 35, 8]. Besides traditional methods like horizontal flipping and random cropping, different augmentation techniques have been proposed, *e.g.*, applying masking out [5] or adding Gaussian noise [22] to regions in images, or mixing up pairs of images and their labels in a convex manner [49]. Recent works also demonstrate that it is possible to learn data augmentation policies automatically for achieving better performance on image classification [3, 4, 19, 21, 52] and object detection [53, 4].

Our work can be regarded as one type of data augmentation: creating additional training samples by injecting noise. However, all previous attempts, by augmenting either with random noise (*e.g.*, Tab. 5 in [18] shows the result of training with random normal perturbations) or adversarial noise [16, 18, 42], fail to improve accuracy on clean images.

3. A Preliminary Way to Boost Performance

Madry *et al.* [23] formulate adversarial training as a min-max game and train models exclusively on adversar-

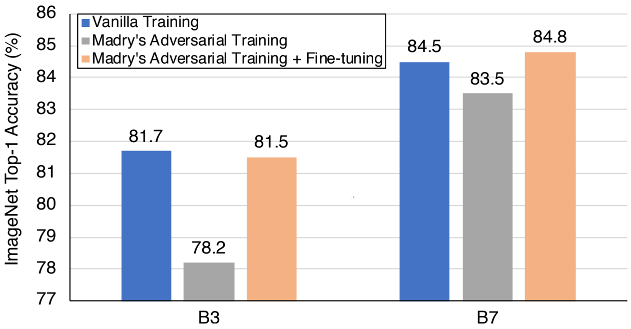


Figure 2. Two take-home messages from the experiments on ImageNet: (1) training exclusively on adversarial examples results in performance degradation; and (2) simply training with adversarial examples and clean images in turn can improve network performance on clean images. Fine-tuning details: we train networks with adversarial examples in the first 175 epochs, and then fine-tune with clean images in the rest epochs.

ial examples to effectively boost model robustness. However, such trained models usually cannot generalize well to clean images as shown in [23, 45]. We validate this result by training a medium-scale model (EfficientNet-B3) and a large-scale model (EfficientNet-B7) on ImageNet using PGD attacker[1] [23]—both adversarially trained models obtain much lower accuracy on clean images compared to their vanilla counterparts. For instance, such adversarially trained EfficientNet-B3 only obtains an accuracy of 78.2% on the clean images, whereas vanilla trained EfficientNet-B3 achieves 81.7% (see Fig. 2).

We hypothesize such performance degradation is mainly caused by *distribution mismatch*—adversarial examples and clean images are drawn from two different domains therefore training exclusively on one domain cannot well transfer to the other. If this distribution mismatch can be properly bridged, then performance degradation on clean images should be mitigated even if adversarial examples are used for training. To validate our hypothesis, we hereby examine a simple strategy—pre-train networks with adversarial examples first, and then fine-tune with clean images.

The results are summarized in Fig. 2. As expected, this simple fine-tuning strategy (marked in light orange) always yields much higher accuracy than Madry's adversarial training baseline (marked in grey), *e.g.*, it increases accuracy by 3.3% for EfficientNet-B3. Interestingly, while compared to the standard vanilla training setting where only clean images are used (marked in blue), this fine-tuning strategy sometimes even help networks to achieve superior performance, *e.g.*, it increases EfficientNet-B7 accuracy by 0.3%, achieving 84.8% top-1 accuracy on ImageNet.

The observation above delivers a promising signal—adversarial examples can be beneficial for model performance if harnessed properly. Nonetheless, we note that this approach fails to improve performance in general, *e.g.*, though such trained EfficientNet-B3 significantly outperforms the Madry's adversarial training baseline, it is still slightly below (-0.2%) the vanilla training setting. Therefore, a natural question arises: is it possible to distill valuable features from adversarial examples in a more effective manner and boost model performance further generally?

4. Methodology

The results in Sec. 3 suggest that properly integrating information from both adversarial examples and clean images even in a simple manner improves model performance. However, such fine-tuning strategy may partially override features learned from adversarial examples, leading to a sub-optimal solution. To address this issue, we propose a more elegant approach, named AdvProp, to jointly learn from clean images and adversarial examples. Our method handles the issue of distribution mismatch via explicitly decoupling batch statistics on normalization layers, and thus enabling a better absorption from both adversarial and clean features. In this section, we first revisit the adversarial training regime in Sec. 4.1, and then introduce how to enable disentangled learning for a mixture of distributions via auxiliary BNs in Sec. 4.2. Finally, we summarize the training and testing pipeline in Sec. 4.3.

4.1. Adversarial Training

We first recall the vanilla training setting, and the objective function is

$$\arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \Big[L(\theta, x, y) \Big], \tag{1}$$

where $\mathbb{D}$ is the underlying data distribution, $L(\cdot,\cdot,\cdot)$ is the loss function, θ is the network parameter, and x is training sample with ground-truth label y.

Consider Madry's adversarial training framework [23], instead of training with original samples, it trains networks with maliciously perturbed samples,

$$\arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \Big[\max_{\epsilon\in\mathbb{S}} L(\theta, x+\epsilon, y) \Big], \tag{2}$$

where ϵ is a adversarial perturbation, $\mathbb{S}$ is the allowed perturbation range. Though such trained models have several nice properties as described in [51, 47, 43], they cannot generalize well to clean images [23, 45].

Unlike Madry's adversarial training, our main goal is to improve network performance on clean images by leveraging the regularization power of adversarial examples. Therefore we treat adversarial images as additional training samples and train networks with a mixture of adversarial examples and clean images, as suggested in [7, 18],

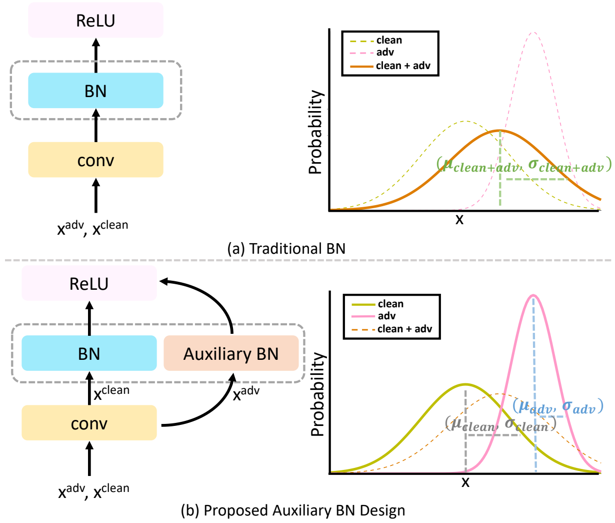
$$\arg\min_{\theta} \Bigg[\mathbb{E}_{(x,y)\sim\mathbb{D}} \Big(L(\theta, x, y) + \max_{\epsilon\in\mathbb{S}} L(\theta, x+\epsilon, y) \Big) \Bigg]. \tag{3}$$

[1] For PGD attacker, we set the maximum perturbation per pixel ϵ=4, the step size α=1 and the number of attack iteration $n = 5$.

Ideally, such trained models should enjoy the benefits from both adversarial and clean domains. However, as observed in former studies [7, 18], directly optimizing Eq. (3) generally yields lower performance than the vanilla training setting on clean images. We hypothesize that the distribution mismatch between adversarial examples and clean images prevents networks from accurately and effectively distilling valuable features from both domains. Next, we will introduce how to properly disentangle different distributions via our auxiliary batch norm design.

4.2. Disentangled Learning via An Auxiliary BN

Batch normalization (BN) [14] serves as an essential component for many state-of-the-art computer vision models [8, 12, 39]. Specifically, BN normalizes input features by the mean and variance computed *within each mini-batch*. One intrinsic assumption of utilizing BN is that the input features should come from a single or similar distributions. This normalization behavior could be problematic if the mini-batch contains data from different distributions, therefore resulting in inaccurate statistics estimation.



(a) Traditional BN

(b) Proposed Auxiliary BN Design

Figure 3. Comparison between (a) traditional BN usage and (b) the utilization of auxiliary BN. The left and right panels illustrate the information flow in the corresponding network architectures and the estimated normalization statistics when facing a mixture of adversarial and clean images, respectively.

We argue that adversarial examples and clean images have different underlying distributions, and the adversarial training framework in Eq. (3) essentially involves a two-component mixture distribution. To disentangle this mixture distribution into two simpler ones respectively for the clean and adversarial images, we hereby propose an auxiliary BN to guarantee its normalization statistics are exclusively preformed on the adversarial examples. Specifically, as illustrated in Fig. 3(b), our proposed auxiliary BN helps to disentangle the mixed distributions by keeping separate BNs to features that belong to different domains. Otherwise, as illustrated in Fig. 3(a), simply maintaining one set of BN statistics results in incorrect statistics estimation, which could possibly lead to performance degradation.

Note that we can generalize this concept to multiple auxiliary BNs, where the number of auxiliary BNs is determined by the number of training sample sources. For example, if training data contains clean images, distorted images and adversarial images, then two auxiliary BNs should be maintained. Ablation studies in Sec. 5.4 demonstrates that such fine-grained disentangled learning with multiple BNs can improve performance further. A more general usage of multiple BNs will be further explored in future works.

4.3. AdvProp

We formally propose AdvProp in Algorithm 1 to accurately acquire clean and adversarial features during training. For each clean mini-batch, we first attack the network using the auxiliary BNs to generate its adversarial counterpart; next we feed the clean mini-batch and the adversarial mini-batch to the same network but applied with different BNs for loss calculation, *i.e.*, use the main BNs for the clean mini-batch and use the auxiliary BNs for the adversarial mini-batch; finally we minimize the total loss w.r.t. the network parameter for gradient updates. In other words, except BNs, convolutional and other layers are jointly optimized for both adversarial examples and clean images.

Note the introduction of auxiliary BN in AdvProp only increases a negligible amount of extra parameters for network training, *e.g.*, 0.5% more parameters than the baseline on EfficientNet-B7. *At test time, these extra auxiliary BNs are all dropped, and we only use the main BNs for inference.*

Algorithm 1: Pseudo code of AdvProp

```
Data: A set of clean images with labels;
Result: Network parameter $\theta$;
for each training step do
    Sample a clean image mini-batch $x^c$ with label $y$;
    Generate the corresponding adversarial mini-batch $x^a$
      using the auxiliary BNs;
    Compute loss $L^c(\theta, x^c, y)$ on clean mini-batch $x^c$ using
      the main BNs;
    Compute loss $L^a(\theta, x^a, y)$ on adversarial mini-batch $x^a$
      using the auxiliary BNs;
    Minimize the total loss w.r.t. network parameter
      $\arg\min_{\theta} L^a(\theta, x^a, y) + L^c(\theta, x^c, y)$.
end
return $\theta$
```

Experiments show that such disentangled learning framework enables networks to get much stronger performance than the adversarial training baseline [7, 18]. Besides, compared to the fine-tuning strategy in Sec. 3, AdvProp also demonstrates superior performance as it enables networks to jointly learn useful feature from adversarial examples and clean examples at the same time.

5. Experiments

5.1. Experiments Setup

Architectures. We choose EfficientNets [41] at different computation regimes as our default architectures, ranging from the light-weight EfficientNet-B0 to the large EfficientNet-B7. Compared to other ConvNets, EfficientNet achieves much better accuracy and efficiency. We follow the settings in [41] to train these networks: RMSProp optimizer with decay 0.9 and momentum 0.9; batch norm momentum 0.99; weight decay 1e-5; initial learning rate 0.256 that decays by 0.97 every 2.4 epochs; a fixed AutoAugment policy [3] is applied to augment training images.

Adversarial Attackers. We train networks with a mixture of adversarial examples and clean images as in Eq. (3). We choose Projected Gradient Descent (PGD) [23] under L_∞ norm as the default attacker for generating adversarial examples on-the-fly. We try PGD attackers with different perturbation size ϵ, ranging from 1 to 4. We set the number iteration for the attackers n=ϵ+1, except for the case ϵ=1 where n is set to 1. The attack step size is fixed to α=1.

Datasets. We use the standard ImageNet dataset [33] to train all models. In addition to reporting performance on the original ImageNet validation set, we go beyond by testing the models on the following test sets:

- **ImageNet-C** [9]. The ImageNet-C dataset is designed for measuring the network robustness to common image corruptions. It consists of 15 diverse corruption types and each type of corruption has five levels of severity, resulting in 75 distinct corruptions.
- **ImageNet-A** [10]. The ImageNet-A dataset adversarially collects 7,500 natural, unmodified but "hard" real-world images. These images are drawn from some challenging scenarios (*e.g.*, occlusion and fog scene) which are difficult for recognition.
- **Stylized-ImageNet** [6]. The Stylized-ImageNet dataset is created by removing local texture cues while retaining global shape information on natural images via AdaIN style transfer [13]. As suggested in [6], networks are required to learn more shape-based representations to improve accuracy on Stylized-ImageNet.

Compared to ImageNet, images from ImageNet-C, ImageNet-A and Stylized-ImageNet are much more challenging, even for human observers.

5.2. ImageNet Results and Beyond

ImageNet Results. Fig. 4 shows the results on the ImageNet validation set. We compare our method with the vanilla training setting. The family of EfficientNets provides a strong baseline, *e.g.*, EfficientNet-B7's 84.5% top-1 accuracy is the prior art on ImageNet [41].

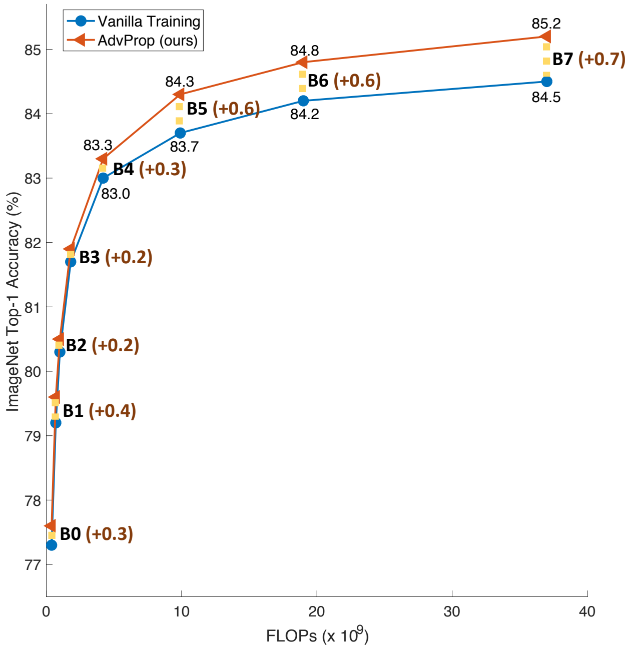


Figure 4. AdvProp boosts model performance over the vanilla training baseline on ImageNet. This improvement becomes more significant if trained with larger networks. Our strongest result is reported by the EfficientNet-B7 trained with AdvProp, *i.e.*, 85.2% top-1 accuracy on ImageNet.

As different networks favor different attacker strengths when trained with AdvProp (which we ablate next), we first report the best result in Fig. 4. Our proposed AdvProp substantially outperforms the vanilla training baseline on all networks. This performance improvement is proportional to the network capacity and larger networks tend to perform better if they are trained with AdvProp. For example, the performance gain is *at most* 0.4% for networks smaller than EfficientNet-B4, but is *at least* 0.6% for networks larger than EfficientNet-B4.

Compared to the prior art, *i.e.*, 84.5% top-1 accuracy, an EfficientNet-B6 trained with AdvProp (with ∼2× less FLOPs than EfficientNet-B7) already surpasses it by 0.3%. Our strongest result is obtained by the EfficientNet-B7 trained with AdvProp which achieves 85.2% top-1 accuracy on ImageNet, beating the prior art by 0.7%.

Generalization on Distorted ImageNet Datasets. Next, we evaluate models on distorted ImageNet datasets, which are much more difficult than the original ImageNet. For instance, though ResNet-50 demonstrates reasonable performance on ImageNet (76.7% accuracy), it only achieves 74.8% mCE (mean corruption error, lower is better) on ImageNet-C, 3.1% top-1 accuracy on ImageNet-A and 8.0% top-1 accuracy on Stylized-ImageNet.

The results are summarized in Tab. 1. Again, our proposed AdvProp consistently outperforms the vanilla training baseline for all models on all distorted datasets. The improvement here is much more significant than that on

Model	ImageNet-C* [9]	ImageNet-A [10]	Stylized-ImageNet* [6]
	mCE ↓	Top-1 Acc. ↑	Top-1 Acc. ↑
ResNet-50	74.8	3.1	8.0
EfficientNet-B0	70.7	6.7	13.1
+ AdvProp **(ours)**	**66.2 (-4.5)**	**7.1 (+0.4)**	**14.6 (+1.5)**
EfficientNet-B1	65.1	9.0	15.0
+ AdvProp **(ours)**	**60.2 (-4.9)**	**10.1 (+1.1)**	**16.7 (+1.7)**
EfficientNet-B2	64.1	10.8	16.8
+ AdvProp **(ours)**	**61.4 (-2.7)**	**11.8 (+1.0)**	**17.8 (+1.0)**
EfficientNet-B3	62.9	17.9	17.8
+ AdvProp **(ours)**	**57.8 (-5.1)**	**18.0 (+0.1)**	**21.4 (+3.6)**
EfficientNet-B4	60.7	26.4	20.2
+ AdvProp **(ours)**	**58.6 (-2.1)**	**27.9 (+1.5)**	**22.5 (+1.7)**
EfficientNet-B5	62.3	29.4	20.8
+ AdvProp **(ours)**	**56.2 (-6.1)**	**34.4 (+5.0)**	**24.4 (+3.6)**
EfficientNet-B6	60.6	34.5	20.9
+ AdvProp **(ours)**	**53.6 (-7.0)**	**40.6 (+6.1)**	**25.9 (+4.0)**
EfficientNet-B7	59.4	37.7	21.8
+ AdvProp **(ours)**	**52.9 (-6.5)**	**44.7 (+7.0)**	**26.6 (+4.8)**

Table 1. AdvProp significantly boost models' generalization ability on ImageNet-C, ImageNet-A and Stylized-ImageNet. The highest result on each dataset is 52.9%, 44.7% and 26.6% respectively, all achieved by the EfficientNet-B7 trained with AdvProp. *For ImageNet-C and Stylized-ImageNet, as distortions are specifically designed for images of the size 224×224×3, so we follow the previous setup [6, 9] to *always fix the testing image size at the scale of 224×224×3* for a fair comparison.

the original ImageNet. For example, AdvProp improves EfficientNet-B3 by 0.2% on ImageNet, and substantially boosts the performance by 5.1% on ImageNet-C and 3.6% on Stylized-ImageNet.

The EfficientNet-B7 trained with AdvProp reports the strongest results on these datasets—it obtains 52.9% mCE on ImageNet-C, 44.7% top-1 accuracy on ImageNet-A and 26.6% top-1 accuracy on Stylized-ImageNet. These are the best results so far *if models are not allowed to train with corresponding distortions [6] or extra data [24, 46].*

To summarize, the results suggest that AdvProp significantly boosts the generalization ability by allowing models to learn much richer internal representations than the vanilla training. The richer representations not only provide models with global shape information for better classifying Stylized-ImageNet dataset, but also increase model robustness against common image corruptions.

Ablation on Adversarial Attacker Strength. We now ablate the effects of attacker strength used in AdvProp on network performance. Specifically, the attacker strength here is determined by perturbation size ϵ, where larger perturbation size indicates stronger attacker. We try with different ϵ ranging from 1 to 4, and report the corresponding accuracy on the ImageNet validation set in Tab. 2.

	B0	B1	B2	B3	B4	B5	B6	B7
PGD5 (ϵ=4)	77.1	79.2	80.3	81.8	**83.3**	**84.3**	**84.8**	**85.2**
PGD4 (ϵ=3)	77.3	79.4	80.4	**81.9**	**83.3**	**84.3**	84.7	85.1
PGD3 (ϵ=2)	77.4	79.4	80.4	**81.9**	83.1	**84.3**	84.7	85.0
PGD1 (ϵ=1)	**77.6**	**79.6**	**80.5**	81.8	83.1	**84.3**	84.6	85.0

Table 2. ImageNet performance of models trained with AdvProp and different attack strength. In general, smaller networks favor weaker attackers, while larger networks favor stronger attackers.

With AdvProp, we observe that smaller networks generally favor weaker attackers. For example, the lightweight EfficientNet-B0 achieves the best performance by using 1-step PGD attacker with perturbation size 1 (denoted as PGD1 (ϵ=1)), significantly outperforms the counterpart which trained with 5-step PGD attacker with perturbation size 4 (denoted as PGD5 (ϵ=4)), *i.e.*, 77.6% v.s. 77.1%. This phenomenon is possibly due to that small networks are limited by their capacity to effectively distill information from strong adversarial examples, even the mixture distributions are well disentangled via auxiliary BNs.

Meanwhile, networks with enough capacity tend to favor stronger attackers. By increasing attacker strength from PGD1 (ϵ=1) to PGD5 (ϵ=4), AdvProp boosts EfficientNet-B7's accuracy by 0.2%. This observation motivate our later ablation on keeping increasing attackers strength to fully exploit the potential of large networks.

5.3. Comparisons to Adversarial Training

As shown in Fig. 4 and Tab. 1, AdvProp improves models for better recognition than the vanilla training baseline. These results contradict previous conclusions [18, 42, 16] that the performance degradation is always observed if adversarial examples are used for training. We hereby provide a set of ablations for explaining this inconsistency. We choose the PGD5 (ϵ=4) as the default attacker to generate adversarial examples during training.

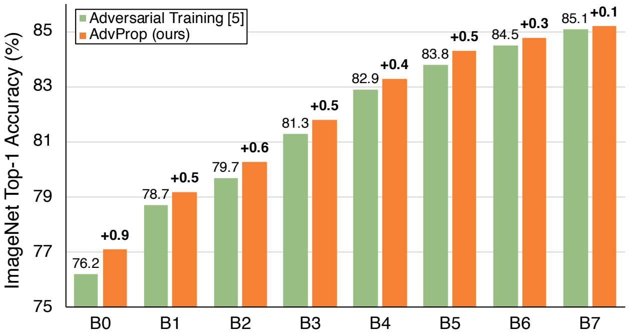


Figure 5. AdvProp substantially outperforms adversarial training [7] on ImageNet, especially for small models.

Comparison Results. We compare AdvProp to traditional adversarial training [7], and report evaluation results on ImageNet validation set in Fig. 5. Compared to the traditional adversarial training, our method consistently achieves better accuracy on all models. This result suggests that carefully handling BN statistics estimation is important for training better models with adversarial examples.

The biggest improvement is observed when using EfficientNet-B0 where our method beats the traditional adversarial training by 0.9%. While by using larger models, this improvement becomes smaller—it stays at ~0.5% until scaling to EfficientNet-B5, but then drops to 0.3% for EfficientNet-B6 and 0.1% for EfficientNet-B7, respectively.

Quantifying Domain Differences. One possible hypothesis for the observation above is that more powerful networks have stronger ability to learn a unified internal representations on the mixed distributions, therefore mitigate the issue of distribution mismatch at normalization layers even without the help of auxiliary BNs. To support this hypothesis, we take models trained with AdvProp, and compare the performance difference between the settings that use either the main BNs or the auxiliary BNs. As such resulted networks share all other layers except BNs, the corresponding performance gap empirically captures the degree of distribution mismatch between adversarial examples and clean images. We use ImageNet validation set for evaluation, and summarize the results in Tab. 3.

	B0	B1	B2	B3	B4	B5	B6	B7
BN	77.1	79.2	80.3	81.8	83.3	84.3	84.8	85.2
Auxiliary BN	73.7	75.9	77.0	78.6	80.5	82.1	82.7	83.3
△	+3.4	+3.3	+3.3	+3.2	+2.8	+2.2	+2.1	+1.9

Table 3. Performance comparison between settings that use either the main BNs and auxiliary BNs on ImageNet. This performance difference captures the degree of distribution mismatch between adversarial examples and clean images.

By training with larger networks, we observe this performance difference gets smaller. Such gap for EfficientNet-B0 is 3.4%, but then is reduced to 1.9% for EfficientNet-B7. It suggests that the internal representations of adversarial examples and clean images learned on large networks are much more similar than that learned on small networks. Therefore, with a strong enough network, it is possible to accurately and effectively learn a mixture of distributions even without a careful handling at normalization layers.

Why AdvProp? For small networks, our comparison shows that AdvProp substantially outperforms the adversarial training baseline. We attribute this performance improvement mainly to the successful disentangled learning via auxiliary BNs.

For larger networks, though the improvement is relatively small on ImageNet, AdvProp consistently outperforms the adversarial training baseline by a large margin on distorted ImageNet datasets. As shown in Tab. 4, AdvProp improves EfficientNet-B7 by 3.1% on ImageNet-C, 4.3% on ImageNet-A and 1.5% on Stylized-ImageNet over the adversarial training baseline.

Model	ImageNet-C [9]	ImageNet-A [10]	Stylized-ImageNet [6]
	mCE ↓	Top-1 Acc. ↑	Top-1 Acc. ↑
B6 + Adv. Training	55.8	37.0	24.7
B6 + AdvProp (**ours**)	**53.6**	**40.6**	**25.9**
B7 + Adv. Training	56.0	40.4	25.1
B7 + AdvProp (**ours**)	**52.9**	**44.7**	**26.6**

Table 4. AdvProp demonstrates much stronger generalization ability on distorted ImageNet datasets (*e.g.*, ImageNet-C) than the adversarial training baseline for larger models.

Moreover, *AdvProp enables large networks to perform better if trained with stronger attackers*. For example, by slightly increasing attacker strength from PGD5 (ϵ=4) to PGD7 (ϵ=6), AdvProp further helps EfficientNet-B7 to achieve **85.3%** top-1 accuracy on ImageNet. Conversely, applying such attacker to traditional adversarial training decreases EfficientNet-B7's accuracy to 85.0%, possibly due to a more severe distribution mismatch between adversarial examples and clean images.

In summary, AdvProp enables networks to enjoy the benefits of adversarial examples even with limited capacity. For networks with enough capacity, compared to adversarial training, AdvProp demonstrates much stronger generalization ability and better at exploiting model capacity for improving performance further.

Missing Pieces in Traditional Adversarial Training. In our reproduced adversarial training, we note it is already better than the vanilla training setting on large networks. For example, our adversarially trained EfficientNet-B7 has 85.1% top-1 accuracy on ImageNet, which beats the vanilla training baseline by 0.6%. However, previous works [18, 16] show adversarial training *always* degrades performance.

Compared to [18, 16], we make two changes in our re-implementation: (1) using stronger networks; and (2) training with weaker attackers. For examples, previous works use networks like Inception or ResNet for training, and set the perturbation size ϵ=16; while we use much stronger EfficientNet for training, and limit the perturbation size to a much smaller value ϵ=4. Intuitively, weaker attackers push the distribution of adversarial examples less away from the distribution of clean images, and larger networks are better at bridging domain differences. Both factors mitigate the issue of distribution mismatch, thus making networks much easier to learn valuable feature from both domains.

5.4. Ablations

Fine-grained Disentangled Learning via Multiple Auxiliary BNs. Following [41], our networks are trained with AutoAugment [3] by default, which include operations like rotation and shearing. We hypothesize these operations (slightly) shift the original data distribution and propose to add an extra auxiliary BN to disentangle these augmented data further for fine-grained learning. In total, we keep one main BN for clean images *without* AutoAugment, and two auxiliary BNs for clean images *with* AutoAugment and adversarial examples, respectively.

We try PGD attackers with perturbation size ranging from 1 to 4, and report the best result on ImageNet in Tab. 5. Compared to the default AdvProp, this fine grained strategy further improves performance. It helps EfficientNet-B0 to achieve **77.9%** accuracy with just 5.3M parameters, which is the state-of-the-art performance for mobile networks. As a comparison, MobileNetv3 has 5.4M parameters with 75.2% accuracy [11]. These results encourage the future investigation on more fine-grained disentangled learning with mixture distributions in general, not just for adversarial training.

	B0	B1	B2	B3	B4	B5	B6	B7
AdvProp	77.6	79.6	80.5	81.9	83.3	84.3	**84.8**	**85.2**
Fine-Grained AdvProp	**77.9**	**79.8**	**80.7**	**82.0**	**83.5**	**84.4**	**84.8**	**85.2**

Table 5. Fine-grained AdvProp substantially boosts model accuracy on ImageNet, especially for small models. We perform fine-grained disentangled learning by keeping an additional auxiliary BN for AutoAugment images.

Comparison to AutoAugment. Training with adversarial examples is a form of data augmentation. We choose the standard Inception-style pre-processing [38] as baseline, and compare the benefits of additionally applying AutoAugment or AdvProp. We train networks with PGD5 (ϵ=4) and evaluate performance on ImageNet.

Results are summarized in Tab. 6. For small models, AutoAugment is slightly better than AdvProp although we argue this gap can be addressed by adjusting the attacker strength. For large models, AdvProp significantly outperforms AutoAugment. Training with AutoAugment and AdvProp in combination is better than using AdvProp alone.

	B0	B1	B2	B3	B4	B5	B6	B7
Inception Pre-process [38]	76.8	78.8	79.8	81.0	82.6	83.2	83.7	84.0
+ AutoAugment [3]	+0.5	+0.4	+0.5	+0.7	+0.4	+0.5	+0.5	+0.5
+ AdvProp (**ours**)	+0.3	+0.3	+0.2	+0.4	+0.3	+0.8	+0.9	+0.9
+ **Both** (**ours**)	+0.3	+0.4	+0.5	+0.8	+0.7	+1.1	+1.1	+1.2

Table 6. Both AutoAugment and AdvProp improves model performance over the Inception-style pre-processing baseline on ImageNet. Large Models generally perform better with AdvProp than AutoAugment. Training with a combination of both is better than using AdvProp alone on all networks.

Attackers Other Than PGD. We hereby study the effects of applying different attackers in AdvProp on model performance. Specifically, we try two different modifications on PGD: (1) we no longer limit the perturbation size to be within the ϵ-ball, and name this attacker to Gradient Descent (GD) as it removes the projection step in PGD; or (2) we skip the random noise initialization step in PGD, turn it to I-FGSM [18]. Other attack hyper-parameters are unchanged: the maximum perturbation size ϵ=4 (if applicable), number of attack iteration n=5 and attack step size α=1.0.

For simplicity, we only experiment with EfficientNet-B3, EfficientNet-B5 and EfficientNet-B7, and report the ImageNet performance in Tab. 7. We observe that all attackers substantially improve model performance over the vanilla training baseline. This result suggests that our AdvProp is not designed for a specific attacker (*e.g.*, PGD), but a general mechanism for improving image recognition models with different adversarial attacker.

	B3	B5	B7
Vanilla Training	81.7	83.7	84.5
PGD [23]	81.8	84.3	85.2
I-FGSM [18]	81.9	84.3	85.2
GD	81.7	84.3	85.3

Table 7. ImageNet performance when trained with different attackers. With AdvProp, all attackers successfully improve model performance over the vanilla training baseline.

ResNet Results. Besides EfficientNets, we also experiment with ResNet [8]. We compare AdvProp against two baselines: vanilla training and adversarial training. We apply PGD5 (ϵ=4) to generate adversarial examples, and follow the settings in [8] to train all networks.

We report model performance on ImageNet in Tab. 8. Compared to vanilla training, adversarial training always degrades model performance while AdvProp consistently leads to better accuracy on all ResNet models. Take ResNet-152 for example, adversarial training *decreases* the baseline performance by 2.0%, but our AdvProp further *boosts* the baseline performance by 0.8%.

	ResNet-50	ResNet-101	ResNet-152	ResNet-200
Vanilla Training	76.7	78.3	79.0	79.3
Adversarial Training	-3.2	-1.8	-2.0	-1.4
AdvProp (**ours**)	**+0.4**	**+0.6**	**+0.8**	**+0.8**

Table 8. Performance comparison among vanilla training, adversarial training and AdvProp on ImageNet. AdvProp reports the best result on all ResNet models.

In Sec. 5.3, we show that adversarial training can improve performance if large EfficientNets are used for training. However, this phenomenon is not observed on ResNet, *e.g.*, adversarial training still leads to inferior accuracy even trained with the large ResNet-200. It may suggest that architecture design also plays an important role when training with adversarial example, and we leave it as a future work.

Pushing The Envelope with a Larger Model. Previous results suggest AdvProp performs better with larger networks. To push the envelope, we train a larger network, EfficientNet-B8, by scaling up EfficientNet-B7 further according to the compound scaling rule in [41].

Our AdvProp improves the accuracy of EfficientNet-B8 from 84.8% to 85.5%, achieving a new state-of-the-art accuracy on ImageNet without using extra data. This result even surpasses the best model reported in [24], which is pretrained on 3.5B extra Instagram images (~3000× more than ImageNet) and requires ~9.4× more parameters (829M vs. 88M) than our EfficientNet-B8.

6. Conclusion

Previous works commonly view adversarial examples as a threat to ConvNets, and suggest training with adversarial examples lead to accuracy drop on clean images. Here we offer a different perspective: to use adversarial examples for improving accuracy of ConvNets. As adversarial examples have different underlying distributions to normal examples, we propose to use an auxiliary batch norm for disentangled learning by processing adversarial examples and clean images separately at normalization layers. Our method, AdvProp, significantly improves accuracy of all ConvNets in our experiments. Our best model reports the state-of-the-art 85.5% top-1 accuracy on ImageNet without any extra data.

Acknowledgement: This work was partially supported by ONR N00014-15-1-2356 to Cihang Xie and Alan Yuille.

References

[1] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019. 2

[2] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020. 1

[3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019. 2, 5, 7, 8

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019. 2

[5] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[6] 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. In *ICLR*, 2018. 1, 2, 5, 6, 7

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 3, 4, 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4, 8

[9] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 1, 2, 5, 6, 7

[10] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 1, 2, 5, 6, 7

[11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 7

[12] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 4

[13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 5

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[15] Charles Jin and Martin Rinard. Manifold regularization for adversarial robustness. *arXiv preprint arXiv:2003.04286*, 2020. 1

[16] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018. 1, 2, 6, 7

[17] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017. 1, 2, 3, 4, 6, 7, 8

[19] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, 2017. 2

[20] Yan Li, Ethan X Fang, Huan Xu, and Tuo Zhao. Inductive bias of gradient descent based adversarial training on separable data. *arXiv preprint arXiv:1906.02931*, 2019. 1

[21] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *arXiv preprint arXiv:1905.00397*, 2019. 2

[22] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. 2

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 3, 5, 8

[24] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 1, 2, 6, 8

[25] Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. *arXiv preprint arXiv:2002.11080*, 2020. 2

[26] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 2018. 1, 2

[27] Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *NeurIPS*, 2019. 2

[28] Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019. 2

[29] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Hang Su, and Jun Zhu. Boosting adversarial training with hypersphere embedding. *arXiv preprint arXiv:2002.08619*, 2020. 1

[30] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *ECCV*, 2018. 1, 2

[31] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020. 2

[32] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019. 2

[33] 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. *IJCV*, 2015. 1, 5

[34] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*. 2

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[36] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training and detection: More robust models generalizing beyond the attack used during training. *arXiv preprint arXiv:1910.06259*, 2019. 1

[37] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *CVPR*, 2019. 2

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 8

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 4

[40] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 1, 2, 5, 7, 8

[42] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018. 1, 2, 6

[43] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv:1805.12152*, 2018. 2, 3

[44] Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *NeurIPS*, 2019. 2

[45] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 1, 2, 3

[46] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019. 6

[47] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019. 2, 3

[48] 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. 2

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2

[50] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1

[51] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *ICML*, 2019. 2, 3

[52] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. 2020. 2

[53] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. *arXiv preprint arXiv:1906.11172*, 2019. 2