

# Adversarially Robust Distillation

Micah Goldblum,\* Liam Fowl,\* Soheil Feizi, Tom Goldstein

University of Maryland  
4176 Campus Drive,  
College Park, Maryland 20742  
goldblum@umd.edu

## Abstract

Knowledge distillation is effective for producing small, high-performance neural networks for classification, but these small networks are vulnerable to adversarial attacks. This paper studies how adversarial robustness transfers from teacher to student during knowledge distillation. We find that a large amount of robustness may be inherited by the student even when distilled on only clean images. Second, we introduce Adversarially Robust Distillation (ARD) for distilling robustness onto student networks. In addition to producing small models with high test accuracy like conventional distillation, ARD also passes the superior robustness of large networks onto the student. In our experiments, we find that ARD student models decisively outperform adversarially trained networks of identical architecture in terms of robust accuracy, surpassing state-of-the-art methods on standard robustness benchmarks. Finally, we adapt recent fast adversarial training methods to ARD for accelerated robust distillation.

## 1 Introduction

State-of-the-art deep neural networks for many computer vision tasks have tens of millions of parameters, hundreds of layers, and require billions of operations per inference (He et al. 2016; Szegedy et al. 2016). However, networks are often deployed on mobile devices with limited compute and power budgets or on web servers without GPUs. Such applications require efficient networks with light-weight inference costs (Dziugaite, Ghahramani, and Roy 2016; Sandler et al. 2018; Courbariaux et al. 2016; Tai et al. 2015).

*Knowledge distillation* was introduced as a way to transfer the knowledge of a large pre-trained teacher network to a smaller light-weight student network (Hinton, Vinyals, and Dean 2015). Instead of training the student network on one-hot class labels, distillation involves training the student network to emulate the outputs of the teacher. Knowledge distillation yields compact student networks that surpass the performance achievable by training from scratch without a teacher (Hinton, Vinyals, and Dean 2015).

Classical distillation methods achieve high efficiency and accuracy but neglect security. Standard neural networks are

easily fooled by *adversarial examples*, in which small perturbations to inputs cause mis-classification (Goodfellow, Shlens, and Szegedy 2014; Szegedy et al. 2013). This phenomenon leads to major security vulnerabilities for high-stakes applications like self-driving cars, medical diagnosis, and copyright control (Santana and Hotz 2016; Lee et al. 2017; Saadatpanah, Shafahi, and Goldstein 2019). In such domains, efficiency and accuracy are not enough – networks must also be adversarially robust.

We study distillation methods that produce robust student networks. Unlike conventional adversarial training (Madry et al. 2017; Shaham, Yamada, and Negahban 2018), which encourages a network to output correct labels within an  $\epsilon$ -ball of training samples, the proposed *Adversarially Robust Distillation* (ARD) instead encourages student networks to mimic their teacher’s output within an  $\epsilon$ -ball of training samples (See Figure 1). Thus, ARD is a natural analogue of adversarial training but in the context of distillation. Formally, we solve the minimax problem

$$\min_{\theta} \mathbb{E}_{(X,y) \sim \mathcal{D}} \left[ \underbrace{\alpha t^2 \text{KL}(S_{\theta}^t(X + \delta_{\theta}), T^t(X))}_{\text{Adversarially Robust Distillation loss}} + (1 - \alpha) \underbrace{\ell(S_{\theta}^t(X), y)}_{\text{classification loss}} \right], \quad (1)$$

where  $\delta_{\theta} = \arg \max_{\|\delta\|_p < \epsilon} \ell(S_{\theta}^t(X + \delta), y)$ ,  $T$  and  $S$  are teacher and student networks, and  $\mathcal{D}$  is the data generating distribution. See Section 5 for a more thorough description. Below, we summarize our contributions in this paper:

- We show that knowledge distillation using only natural images can preserve much of the teacher’s robustness to adversarial attacks (see Table 1), enabling the production of efficient robust models without the expensive cost of adversarial training.
- We introduce Adversarially Robust Distillation (ARD) for producing small robust student networks. In our experiments, ARD students exhibit higher robust accuracy than adversarially trained models with identical architecture, and ARD often exhibits higher natural accuracy simultaneously. Interestingly, ARD students may exhibit even higher robust accuracy than their teacher (see Table 1).

\* Authors contributed equally.

- We accelerate our method for efficient training by adapting fast adversarial training methods.

Model	Robust Accuracy ( $\mathcal{A}_{adv}$ )
AT ResNet18 teacher	44.46%
AT ResNet18 $\rightarrow$ MobileNetV2	38.21%
AT ResNet18 $\xrightarrow{\text{ARD}}$ MobileNetV2	<b>50.22%</b>

Table 1: Performance of an adv. trained (AT) teacher network and its student on CIFAR-10, where robust accuracy ( $\mathcal{A}_{adv}$ ) is with respect to a 20-step PGD attack as in (Madry et al. 2017). “ $\rightarrow$ ” denotes “knowledge distillation onto”. “ $\xrightarrow{\text{ARD}}$ ” denotes “adversarially robust distillation onto”.

Table 1 shows that a student may learn robust behavior from a robust teacher, even if it only sees clean images during distillation. Our method, ARD, outperforms knowledge distillation for producing robust students. To gain initial intuition for the differences between these methods, we visualize decision boundaries for a toy problem in Figure 2. Randomly generated training data are depicted as colored dots with boxes showing the desired  $\ell_\infty$  robustness radius. Background colors represent the classification regions of the networks (10-layer teacher and 5-layer student). In each case, the network achieves perfect training accuracy. A training point is vulnerable to attack if its surrounding box contains multiple colors. Results in Figure 2 are consistent with our experiments in Section 5 on more complex datasets. We see in these decision boundary plots that knowledge distillation from a robust teacher preserves some robustness, while ARD produces a student who closely mimics the teacher.

## 2 Related Work

Early schemes for compressing neural networks involved binarized weights to reduce storage and computation costs (Courbariaux et al. 2016). Other efforts focused on speeding up calculations via low-rank regularization and pruning weights to reduce computation costs (Tai et al. 2015; Li et al. 2016). Knowledge distillation teaches a student network to mimic a more powerful teacher (Hinton, Vinyals, and Dean 2015). The student is usually a small, lightweight architecture like MobileNetV2 (MNV2) (Sandler et al. 2018).

Knowledge distillation has also been adapted for robustness in a technique called *defensive distillation* (Papernot et al. 2016). In this setting, the teacher and student have identical architectures. An initial network is trained on class labels and then distilled at temperature  $t$  onto a network of identical architecture. Defensive distillation improves robustness to a certain  $\ell_0$  attack (Papernot et al. 2016). However, defensive distillation gains robustness due to gradient masking, and this defense has been broken using  $\ell_0$ ,  $\ell_\infty$ , and  $\ell_2$  attacks (Carlini and Wagner 2016; 2017).

Various methods exist that modify networks to achieve robustness. Some model-specific variants of adversarial training utilize surrogate loss functions to minimize the difference in network output on clean and adversarial data (Zhang

et al. 2019b; Miyato et al. 2018), feature denoising blocks (Xie et al. 2018), and logit pairing to squeeze logits from clean and adversarial inputs (Kannan, Kurakin, and Goodfellow 2018). Still other methods, such as JPEG compression, pixel deflection, and image superresolution, are model-agnostic and minimize the effects of adversarial examples by transforming inputs (Dziugaite, Ghahramani, and Roy 2016; Prakash et al. 2018; Mustafa et al. 2019).

Recently, there has been work on defensive-minded compression. The authors of (Wijayanto et al. 2018) and (Zhao et al. 2018) study the preservation of robustness under quantization. Defensive Quantization involves quantizing a network while minimizing the Lipschitz constant to encourage robustness (Lin, Gan, and Han 2019). While quantization reduces space complexity, it does not reduce the number of Multiply-Add (MAdd) operations needed for inference, although operations performed at lower precision may be faster depending on hardware. Moreover, Defensive Quantization is not evaluated against strong attackers. Another compression technique involves pruning (Sehwag et al. 2019). Pruning does reduce the number of parameters in a network, but it does not decrease network depth and thus may not accelerate inference. Additionally, this work does not achieve high compression ratios and does not achieve competitive performance on CIFAR-10. Pruning maintains the same architectural framework and does not allow a user to compress a state-of-the-art large robust network into a lightweight architecture of their choice. Creating small robust models is also of interest for the few-shot setting. Adversarial querying approaches this problem from the meta-learning perspective (Goldblum, Fowl, and Goldstein 2019).

## 3 Problem Setup

Knowledge distillation employs a teacher-student paradigm in which a small student network learns to mimic the output of an often much larger teacher model (Hinton, Vinyals, and Dean 2015). Knowledge distillation in its purist form entails the minimization problem,

$$\min_{\theta} \ell_{\text{KD}}(\theta), \quad \ell_{\text{KD}}(\theta) = \mathbb{E}_{X \sim \mathcal{D}} [\text{KL}(S_{\theta}^t(X), T^t(X))], \quad (2)$$

where KL is KL divergence,  $S_{\theta}$  is a student network with parameters  $\theta$ ,  $T$  is a teacher network,  $t$  is a temperature constant, and  $X$  is an input to the networks drawn from data generating distribution  $\mathcal{D}$ . The temperature constant refers to a number by which the logits are divided before being fed into the softmax function. Intuitively, knowledge distillation involves minimizing the average distance from the student’s output to the teacher’s output over data from a distribution. The softmax outputs of the teacher network, also referred to as soft labels, may be more informative than true data labels alone. In (Hinton, Vinyals, and Dean 2015), the authors suggest using a linear combination of the loss function,  $\ell_{\text{KD}}(\theta)$ , and the cross-entropy between the softmax output of the student network and the one-hot vector representing the true label in order to improve natural accuracy of the student model, especially on difficult datasets. In this case, we

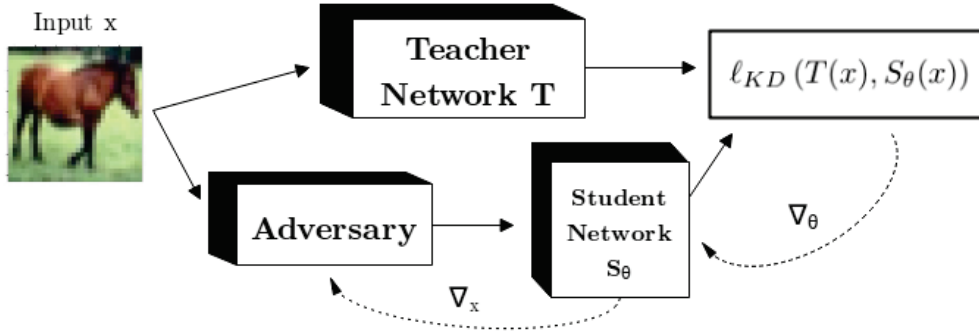


Figure 1: Adversarially Robust Distillation (ARD) works by minimizing discrepancies between the outputs of a teacher on natural images and the outputs of a student on adversarial images.

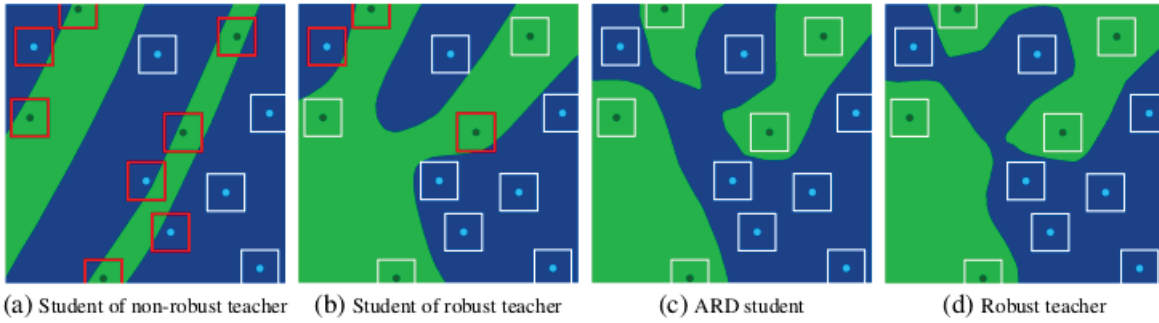


Figure 2: A student distilled from a robust teacher is more robust than a naturally trained network, but ARD produces a more robust network than either and closely mimics the teacher’s decision boundary. Adversarially vulnerable training points have  $\ell_\infty$  boxes outlined in red.

have the loss function

$$\ell_{\text{KD}}(\theta) = \mathbb{E}_{(X,d) \sim \mathcal{D}} [\alpha t^2 \text{KL}(S_\theta^t(X), T^t(X)) + (1 - \alpha) \ell(S_\theta^t(X), y)], \quad (3)$$

where  $\ell$  is the standard cross-entropy loss, and  $y$  is the label. In our experiments, we use  $\alpha = 1$  except where otherwise noted. We investigate if students trained using this method inherit their teachers’ robustness to adversarial attacks. We also combine knowledge distillation with adversarial training.

Adversarial training is another method for encouraging robustness to adversarial attacks during training (Shaham, Yamada, and Negahban 2018). Adversarial training involves the minimax optimization problem,

$$\min_{\theta} \mathbb{E}_{(X,y) \sim \mathcal{D}} \left[ \max_{\|\delta\|_p < \epsilon} L_\theta(X + \delta, y) \right], \quad (4)$$

where  $L_\theta(X + \delta, y)$  is the loss of network with parameters  $\theta$ , input  $X$  perturbed by  $\delta$ , and label  $y$ . Adversarial training encourages a student to produce the correct label in an  $\epsilon$ -ball surrounding data points. Virtual Adversarial Training (VAT) and TRADES instead use as a loss function a linear combination of cross-entropy loss and KL divergence between the network’s softmax output from clean input and from adversarial input (Miyato et al. 2018;

Zhang et al. 2019b). The KL divergence term acts as a consistency regularizer which trains the neural network to produce identical output on a natural image and adversarial images generated from the natural image. As a result, this term encourages the neural network’s output to be constant in  $\epsilon$ -balls surrounding data points.

Knowledge distillation is useful for producing accurate student networks when highly accurate teacher networks exist. However, the resulting student networks may not be robust to adversarial attacks (Carlini and Wagner 2017). We combine the central ideas of knowledge distillation and adversarial training to similarly produce robust student networks when robust teacher networks exist.

We focus on adversarial robustness to  $\ell_\infty$  attacks since these are pervasive in the robustness literature. Thus, we carry out both adversarial training and ARD with FGSM-based PGD  $\ell_\infty$  attacks similarly to (Madry et al. 2017; Zhang et al. 2019b). We re-implemented the methods from these papers to perform adversarial training and to establish performance baselines. In our experiments, we use WideResNet (34-10) and ResNet18 teacher models as well as MobileNetV2 (MNV2) students (Zagoruyko and Komodakis 2016; He et al. 2016; Sandler et al. 2018). Adversarial training and TRADES teacher models are as described in (Madry et al. 2017; Zhang et al. 2019b).

## 4 Adversarial robustness is preserved under knowledge distillation

The softmax output of a neural network classifier trained with cross-entropy loss estimates the posterior distribution over class labels on the training data distribution (Bridle 1990). Empirical study of distillation suggests that neural networks perform better when trained on the softmax output of a powerful teacher than when trained on only class labels (Hinton, Vinyals, and Dean 2015). The distribution of images generated by adversarial attacks with respect to a particular model and a data generating distribution,  $D$ , may differ from the distribution of natural images generated by  $D$ . Distillation from a naturally trained teacher is known to produce student models which are not robust to adversarial attacks (Carlini and Wagner 2017). Thus, we suspected that a non-robust teacher network trained on natural images would be poorly calibrated for estimating posterior probabilities on the distribution of images generated by adversarial attacks. On the other hand, an adversarially trained teacher network might provide a more accurate estimate of posterior probabilities on this distribution. We compare the robustness of student models distilled from both naturally trained and adversarially trained teachers. If we distill from an adversarially trained teacher, will the student inherit robustness?

If robustness transfers from teacher to student, we can harness state-of-the-art robust teacher networks to produce accurate, robust, and efficient student networks. Moreover, since adversarial training is slow due to the bottleneck of crafting adversarial samples for every batch, we could create many different student networks from one teacher, and adversarial training would only need to be performed once. This routine of training a robust teacher and then distilling onto robust students would be far more time efficient than training many robust student networks individually.

### 4.1 Non-robust teachers produce non-robust students

To establish a baseline for comparison, we distill a non-robust ResNet18 teacher and evaluate against a 20-step PGD attack as in (Madry et al. 2017). We verify known results showing that defensive distillation is ineffective for producing adversarially robust students (Carlini and Wagner 2017).

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
ResNet18 teacher	94.75%	0.0%
ResNet18 $\rightarrow$ ResNet18	94.92%	0.0%
ResNet18 $\rightarrow$ MNV2	93.53%	0.0%

Table 2: Performance of a naturally trained teacher network and its students distilled (with  $t = 30$ ) on CIFAR-10, where robust accuracy is with respect to a 20-step PGD attack as in (Madry et al. 2017).  $\mathcal{A}_{nat}$  denotes natural accuracy.

### 4.2 Robust teachers can produce robust students, even distilling on only clean data

Next, we substitute in a robust adversarially trained ResNet18 teacher network and run the same experiments.

We find that our new student networks are far more robust than students of the non-robust teacher (see Table 3). In fact, the student networks acquire most of the teacher’s robust accuracy. These results confirm that robust lightweight networks may indeed be produced cheaply through knowledge distillation without undergoing adversarial training.

Student Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
AT ResNet18 teacher	76.54%	44.46%
AT ResNet18 $\rightarrow$ ResNet18	76.13%	40.13%
AT ResNet18 $\rightarrow$ MNV2	76.86%	38.21%

Table 3: Performance of an adversarially trained ResNet18 teacher network and student networks of various sizes distilled on CIFAR-10, where robust accuracy is with respect to a 20-step PGD attack as in (Madry et al. 2017).

### 4.3 Not all robust networks are good teachers, and robustness does not transfer on some datasets

In the previous experiments, we see that a student network may inherit a significant amount of robustness from a robust teacher network during knowledge distillation. However, some robust teachers are not conducive to this robustness transfer. We use robust WideResNet (34-10) models trained using adversarial training and TRADES to show that while these models do transfer robustness during knowledge distillation, they transfer less than the weaker ResNet18 teacher network from the previous section (See Table 4). Additionally, a robust WRN teacher model transfers almost no robustness under knowledge distillation against 20-step PGD untargeted attacks on CIFAR-100, a much harder dataset for robustness to untargeted attacks than CIFAR-10 (See Table 5). Further experiments show that robustness transfer diminishes rapidly as we decrease  $\alpha$  from our default value of 1.

For these reasons, we develop ARD for distilling a variety of teachers in order to produce robust students. Using ARD, robustness is preserved on architectures and datasets that do not transfer robustness under vanilla knowledge distillation.

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
AT WRN teacher	84.41%	45.75%
TRADES WRN teacher	84.92%	56.61%
AT WRN $\rightarrow$ MNV2	92.49%	5.46%
TRADES WRN $\rightarrow$ MNV2	85.6%	21.69%

Table 4: Robust WRN teacher models and their students on CIFAR-10, where robust accuracy is with respect to a 20-step PGD attack as in (Madry et al. 2017).

## 5 Improving the robustness of student models with Adversarially Robust Distillation (ARD)

We combine the central machinery from knowledge distillation, adversarial training, and TRADES/VAT to produce

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
AT WRN teacher	59.9%	28.36%
AT WRN $\rightarrow$ MNV2	25.54%	1.30%
AT WRN $\rightarrow$ MNV2 ( $\alpha = 0.95$ )	75.94%	0.02%
AT WRN $\rightarrow$ MNV2 ( $\alpha = 0.93$ )	76.38%	0.00%

Table 5: Robust teacher network and its students on CIFAR-100, where robust accuracy is with respect to a 20-step PGD attack as in (Madry et al. 2017).

small robust student models from much larger robust teacher models using a method we call *Adversarially Robust Distillation*. ARD not only produces more robust students than knowledge distillation, but ARD also works for teachers and datasets on which knowledge distillation is ineffective for transferring robustness. Our procedure is a natural analogue of adversarial training but in a distillation setting. During standard adversarial training, we encourage a network to produce the ground truth label corresponding to a clean input when the network is exposed to an adversary. Along the same lines, our method treats the teacher network’s softmax output on clean data as the ground truth and trains a student network to reproduce this ground truth when exposed to adversarial examples. We start with a robust teacher model,  $T$ , and we train the student model  $S_\theta$  by solving the following optimization problem ((1) revisited):

$$\min_{\theta} \mathbb{E}_{(X,y) \sim \mathcal{D}} [\alpha t^2 \text{KL}(S_\theta^t(X + \delta_\theta), T^t(X)) + (1 - \alpha) \ell(S_\theta^t(X), y)],$$

where  $\delta_\theta = \arg \max_{\|\delta\|_p < \epsilon} \ell(S_\theta^t(X + \delta), y)$ ,  $\ell$  is cross-entropy loss, and we divide the logits of both student and teacher models by temperature term  $t$  during training. The  $t^2$  term is used as in (Hinton, Vinyals, and Dean 2015) since dividing the logits shrinks the gradient. The cross-entropy loss, which encourages natural accuracy, is a standard training loss. Thus,  $\alpha$  is a hyperparameter that prioritizes similarity to the teacher over natural accuracy. In our experiments, we set  $\alpha = 1$  except where otherwise noted, eliminating the cross-entropy term. We find that lower values of  $\alpha$  are useful for improving performance on harder classification tasks. Our training routine involves the following procedure:

### 5.1 ARD works with teachers that fail to transfer robustness under knowledge distillation

In Section 4, we saw that that some teacher networks do not readily transfer robustness. We see through our experiments in Table 6 that ARD is able to create robust students from teachers whose robustness failed to transfer during knowledge distillation. TRADES and adversarially trained MobileNetV2 (MNV2) models are each outperformed in both natural and robust accuracy simultaneously by an ARD variant. In these experiments, the WideResnet teacher model contains  $20\times$  as many parameters and performs  $70\times$  as many MAdd operations as the MobileNetV2 student.

### Algorithm 1: Adversarially Robust Distillation (ARD)

**Require:** Student and teacher networks  $S$  and  $T$ , learning rate  $\gamma$ , dataset  $\{(\mathbf{x}_i, y_i)\}$ , number of steps,  $K$ , per PGD attack, and  $\epsilon$  maximum attack radius. Initialize  $\theta$ , the weights  $S$ ;  
**for** Epoch =  $1, \dots, N_{epochs}$  **do**  
  **for** Batch =  $1, \dots, N_{batches}$  **do**  
    Construct adv. example  $\mathbf{x}'_i$  for each  $\mathbf{x}_i \in$  Batch by maximizing cross-entropy between  $S_\theta(\mathbf{x}'_i)$  and  $y_i$  constrained to  $\|\mathbf{x}_i - \mathbf{x}'_i\|_p < \epsilon$  using  $K$ -step PGD.  
    Compute  $\nabla_{\theta} \ell_{ARD}(\{\mathbf{x}_i\}, \theta) = \sum_i \nabla_{\theta} [\alpha t^2 \text{KL}(S_\theta^t(\mathbf{x}'_i), T^t(\mathbf{x}_i)) + (1 - \alpha) \ell(S_\theta^t(X), y)]$ , over the current batch.  
     $\theta \leftarrow \theta - \gamma \nabla_{\theta} \ell_{ARD}(\{\mathbf{x}_i\}, \theta)$

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
TRADES WRN teacher	84.92%	56.61%
AT MNV2	80.50%	46.90%
TRADES MNV2	83.59%	44.79%
TRADES WRN $\xrightarrow{\text{ARD}}$ MNV2	82.63%	<b>50.42%</b>
TRADES WRN $\xrightarrow{\text{ARD}}$ MNV2 ( $\alpha = 0.95$ )	<b>84.70%</b>	46.28%

Table 6: Performance on CIFAR-10, where robust accuracy is w.r.t. a 20-step PGD attack as in (Madry et al. 2017).

### 5.2 ARD works on datasets where knowledge distillation fails

In Section 4, we saw that a student network inherited little robustness from an adversarially trained teacher network on CIFAR-100. This dataset is very difficult to protect from untargeted attacks because it contains many classes that are similar in appearance. In Table 7, we see that a MobileNetV2 student model trained on CIFAR-100 using ARD from an adversarially trained WideResNet is significantly more robust than an adversarially trained MobileNetV2. In fact, the ARD model is nearly as robust as its teacher.

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
AT WRN teacher	59.90%	28.36%
AT MNV2	55.62%	22.80%
AT WRN $\xrightarrow{\text{ARD}}$ MNV2 ( $\alpha = 0.93$ )	55.47%	<b>27.64%</b>

Table 7: Performance on CIFAR-100, where robust accuracy is w.r.t. a 20-step PGD attack as in (Madry et al. 2017).

### 5.3 ARD can produce networks more robust than their teacher

In some experiments, ARD student networks are more robust than their teacher. Interestingly, this behavior does not depend on distilling from a high-capacity network to a low capacity network; distilling Resnet18 onto itself and Mo-

MobileNetV2 onto itself using ARD results in far better robustness than adversarial training alone.

We seek to understand if this increased robustness is caused by differences between the student and teacher architectures, and so we use ARD to distill a teacher network onto a student network with an identical architecture. In our experiments, ARD boosted the robustness of both the ResNet18 and MobileNetV2 models (See Table 8).

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
AT ResNet18	76.54%	44.46%
AT MNV2	80.50%	46.90%
AT ResNet18 $\xrightarrow{\text{ARD}}$ ResNet18	79.49%	<b>51.21%</b>
AT MNV2 $\xrightarrow{\text{ARD}}$ MNV2	81.22%	47.95%
AT ResNet18 $\xrightarrow{\text{ARD}}$ MNV2	79.47%	50.22%

Table 8: Performance on CIFAR-10, where robust accuracy is w.r.t. a 20-step PGD attack as in (Madry et al. 2017).

#### 5.4 Accelerating ARD using fast adversarial training methods

The ARD procedure described above takes approximately the same amount of time as adversarial training. Adversarial training is slow since it requires far more gradient calculations than natural training. Several methods have been proposed recently for accelerating adversarial training (Shafahi et al. 2019; Zhang et al. 2019a). We similarly accelerate performance for ARD by adapting “free” adversarial training to distillation. This version, *Fast-ARD*, described in Algorithm 2, is equally fast to knowledge distillation (see Table 10 for a list of training times). During training, we replay each mini-batch several times in a row. On each replay, we simultaneously compute the gradient of the loss w.r.t. the image and parameters using the same backward pass. Then, we update the adversarial attack and the network’s parameters simultaneously. Empirically, Fast-ARD produces less robust students than the full ARD above, but it produces higher robust accuracy compared to models with identical architecture trained using existing accelerated free adversarial training methods as seen in Table 9. Furthermore, Fast-ARD from a TRADES WideResNet onto MobileNetV2 produces a more robust student than our most robust MobileNetV2 produced during vanilla knowledge distillation and in the same amount of training time. Our accelerated algorithm is detailed in Algorithm 2.

#### 5.5 ARD and Fast-ARD models are more robust than their adversarially trained counterparts

While 20-step PGD is a powerful attack, we also test ARD against other  $\ell_\infty$  attackers including Momentum Iterative Fast Gradient Sign Method (Dong et al. 2018), DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016), 1000-step PGD, and PGD with random restarts. We find that ARD and Fast-ARD outperform adversarial training and free training respectively across all attacks we tried (see Table 11).

---

#### Algorithm 2: Fast-ARD with free adversarial training

---

**Requires:** Student and teacher networks  $S$  and  $T$ , learning rate  $\gamma$ , norm  $p$ , dataset  $\{(\mathbf{x}_i, y_i)\}$ , and attack step-size  $r$  and radius  $\epsilon$   
Initialize  $\theta$ , the weights of network  $S$ , and set  $\delta = 0$ .  
**for** Epoch = 1, ...,  $\frac{N_{epochs}}{m}$  **do**  
  **for** Batch = 1, ...,  $N_{batches}$  **do**  
    **for** j = 1, ..., m **do**  
      For  $\mathbf{x}_i \in$  Batch, find new perturbation  $\delta'_i$  by maximizing cross-entropy between  $S_\theta(\mathbf{x}_i + \delta_i + \delta'_i)$  and  $y_i$  over  $\delta'_i$ , constrained to  $\|\delta_i + \delta'_i\|_p < \epsilon$ , using a 1-step PGD attack.  
      Compute the gradient of the loss function  $\nabla_\theta \ell_{ARD}(\{\mathbf{x}_i\}, \theta) = \sum_i \nabla_\theta [\alpha^2 \text{KL}(S_\theta^t(\mathbf{x}'_i), T^t(\mathbf{x}_i)) + (1 - \alpha)\ell(S_\theta^t(X), \mathbf{y})]$ , over current batch.  
       $\delta_i \leftarrow \delta_i + \delta'_i$   
       $\theta \leftarrow \theta - \gamma \nabla_\theta \ell_{ARD}(\{\mathbf{x}_i\}, \theta)$

---

Model	$\mathcal{A}_{nat}$	$\mathcal{A}_{adv}$
Free trained MNV2 (m=4)	82.63%	23.13%
TRADES WRN $\xrightarrow{\text{F-ARD}}$ MNV2 (m=4)	<b>83.51%</b>	<b>37.07%</b>
Free trained MNV2 (m=8)	72.30%	27.96%
TRADES WRN $\xrightarrow{\text{F-ARD}}$ MNV2 (m=8)	76.38%	36.85%

Table 9: Performance of MobileNetV2 classifiers, free trained and Fast-ARD, on CIFAR-10, where robust accuracy is with respect to a 20-step PGD attack as in (Madry et al. 2017). “ $\xrightarrow{\text{F-ARD}}$ ” denotes “Fast-ARD onto”.

Model	Time (hrs)
AT MNV2	41.09
TRADES WRN $\rightarrow$ MNV2	5.13
TRADES WRN $\xrightarrow{\text{ARD}}$ MNV2	41.06
TRADES WRN $\xrightarrow{\text{F-ARD}}$ MNV2 (m=4)	5.15
TRADES WRN $\xrightarrow{\text{F-ARD}}$ MNV2 (m=8)	<b>5.10</b>

Table 10: Training times for adversarial training, clean distillation, ARD, and Fast-ARD. Each model was trained with 200 parameter updates for each training image (equivalent of 200 epochs). All models were trained on CIFAR-10 with a single RTX 2080 Ti GPU and identical batch sizes. The AT model and the ARD model were trained with a 10-step PGD attack. “ $\xrightarrow{\text{F-ARD}}$ ” denotes “Fast-ARD onto”.

## 6 Space and time efficiency of student and teacher models

We perform our experiments for ARD with WideResNet (34-10) and ResNet18 teacher models as well as a MobileNetV2 student model. We consider two network quali-

Model	MI-FGSM <sub>20</sub>	DeepFool	1000-PGD	20-PGD 100-restarts
AT MNV2	50.82%	57.74%	46.51%	46.79%
$\xrightarrow{\text{ARD}}$ MNV2	<b>55.16%</b>	<b>64.61%</b>	<b>49.98%</b>	<b>50.30%</b>
Free trained MNV2 (m=4)	30.60%	41.09%	22.23%	22.94%
$\xrightarrow{\text{F-ARD}}$ MNV2(m=4)	<b>44.78%</b>	<b>60.03%</b>	<b>36.01%</b>	<b>36.88%</b>

Table 11: Robust validation accuracy of adversarially trained and free trained MobileNetV2 and TRADES WRN ARD (and Fast-ARD) onto MobileNetV2 on CIFAR-10 under various attacks. All attacks use  $\epsilon = \frac{8}{255}$ .

ties for quantifying compression. First, we study space efficiency by counting the number of parameters in a network. Second, we study time complexity. To this end, we compute the multiply-add (MAdd) operations performed during a single inference. The real time elapsed during this inference will vary as a result of implementation and deep learning framework, so we use MAdd, which is invariant under implementation and framework, to study time complexity.

The WideResNet and ResNet18 teachers we employ contain  $\sim 46.2\text{M}$  and  $\sim 11.2\text{M}$  parameters respectively, while the MobileNetV2 student contains  $\sim 2.3\text{M}$  parameters. A forward pass through the WRN and ResNet18 teacher models takes  $\sim 13.3\text{B}$  MAdd operations and  $\sim 1.1\text{B}$  MAdd operations respectively, while a forward pass through the student model takes  $\sim 187\text{M}$  MAdd operations. To summarize these network traits, compared to a WideResnet (34-10) teacher, the MobileNetV2 student model:

- Contains  $\sim 5\%$  as many parameters
- Performs  $\sim 1.4\%$  as many MAdd operations during a forward pass

## 7 Discussion

We find that knowledge distillation allows a student network to absorb a large amount of a teacher network’s robustness to adversarial attacks, even when the student is only trained on clean data. However, in some cases, a distilled student model is still far less robust than the teacher. To improve student robustness, we introduce Adversarially Robust Distillation (ARD). In our experiments, student models trained using our method outperform similar networks trained using adversarial training in robust and often natural accuracy. Our models exceed state-of-the-art performance on CIFAR-10 and CIFAR-100 benchmarks. Furthermore, we develop a free adversarial training variant of ARD and demonstrate appreciably accelerated performance.

Recent work on distillation has produced significant improvements over vanilla knowledge distillation (Chen et al. 2018). We believe that Knowledge Distillation with Feature Maps could improve both natural and robust accuracy of student networks. Adaptive data augmentation like AutoAugment (Cubuk et al. 2018) may also improve performance of both normal knowledge distillation for robust teachers and ARD. Finally, with the recent publication of fast adversarial training methods (Shafahi et al. 2019; Zhang et al. 2019a), we hope to further accelerate ARD.

## 8 Experimental details

We train our models for 200 epochs with SGD and a momentum term of  $2(10^{-4})$ . Fast-ARD models are trained for  $\frac{200}{m}$  epochs so that they take the same amount of time as natural distillation. We use an initial learning rate of 0.1, and we decrease the learning rate by a factor of 10 on epochs 100 and 150 (epochs  $\frac{100}{m}$  and  $\frac{150}{m}$  for Fast-ARD). We use a temperature term of 30 for CIFAR-10 and 5 for CIFAR-100. To craft adversarial examples during training, we use FGSM-based PGD with 10 steps,  $\ell_\infty$  attack radius of  $\epsilon = \frac{8}{255}$ , a step size of  $\frac{2}{255}$ , and a random start.

A PyTorch implementation of ARD can be found at: <https://github.com/goldblum/AdversariallyRobustDistillation>

## Acknowledgments

This research was generously supported by DARPA, including the GARD program, QED for RML, and the Young Faculty Award program. Further funding was provided by the AFOSR MURI program, and the National Science Foundation (DMS).

## References

- Bridle, J. S. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*. Springer. 227–236.
- Carlini, N., and Wagner, D. 2016. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*.
- Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.
- Chen, W.-C.; Chang, C.-C.; Lu, C.-Y.; and Lee, C.-R. 2018. Knowledge distillation with feature maps for image classification. *arXiv preprint arXiv:1812.00660*.
- Courbariaux, M.; Hubara, I.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.
- Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.
- Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9185–9193.

- Dziugaite, G. K.; Ghahramani, Z.; and Roy, D. M. 2016. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*.
- Goldblum, M.; Fowl, L.; and Goldstein, T. 2019. Adversarially robust few-shot learning: A meta-learning approach. *arXiv preprint arXiv:1910.00982*.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Kannan, H.; Kurakin, A.; and Goodfellow, I. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.
- Lee, J.-G.; Jun, S.; Cho, Y.-W.; Lee, H.; Kim, G. B.; Seo, J. B.; and Kim, N. 2017. Deep learning in medical imaging: general overview. *Korean journal of radiology* 18(4):570–584.
- Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. P. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.
- Lin, J.; Gan, C.; and Han, S. 2019. Defensive quantization: When efficiency meets robustness. *arXiv preprint arXiv:1904.08444*.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- Miyato, T.; Maeda, S.-i.; Ishii, S.; and Koyama, M. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*.
- Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.
- Mustafa, A.; Khan, S. H.; Hayat, M.; Shen, J.; and Shao, L. 2019. Image super-resolution as a defense against adversarial attacks. *arXiv preprint arXiv:1901.01677*.
- Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; and Swami, A. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, 582–597. IEEE.
- Prakash, A.; Moran, N.; Garber, S.; DiLillo, A.; and Storer, J. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8571–8580.
- Saadatpanah, P.; Shafahi, A.; and Goldstein, T. 2019. Adversarial attacks on copyright detection systems. *arXiv preprint arXiv:1906.07153*.
- Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.
- Santana, E., and Hotz, G. 2016. Learning a driving simulator. *arXiv preprint arXiv:1608.01230*.
- Sehwag, V.; Wange, S.; Mittal, P.; and Jana, S. 2019. Towards compact and robust deep neural networks. *arXiv preprint arXiv:1906.06110*.
- Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *arXiv preprint arXiv:1904.12843*.
- Shaham, U.; Yamada, Y.; and Negahban, S. 2018. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing* 307:195–204.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Tai, C.; Xiao, T.; Zhang, Y.; Wang, X.; et al. 2015. Convolutional neural networks with low-rank regularization. *arXiv preprint arXiv:1511.06067*.
- Wijayanto, A. W.; Jin, C. J.; Madhawa, K.; and Murata, T. 2018. Robustness of compressed convolutional neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, 4829–4836. IEEE.
- Xie, C.; Wu, Y.; van der Maaten, L.; Yuille, A.; and He, K. 2018. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*.
- Zagoruyko, S., and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.
- Zhang, D.; Zhang, T.; Lu, Y.; Zhu, Z.; and Dong, B. 2019a. You only propagate once: Painless adversarial training using maximal principle. *arXiv preprint arXiv:1905.00877*.
- Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; El Ghaoui, L.; and Jordan, M. I. 2019b. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*.
- Zhao, Y.; Shumailov, I.; Mullins, R.; and Anderson, R. 2018. To compress or not to compress: Understanding the interactions between adversarial attacks and neural network compression. *arXiv preprint arXiv:1810.00208*.