RLNet: Robust Linearized Networks for Efficient Private Inference

Sreetama Sarkar[1,*] Souvik Kundu[2,*] Peter A. Beerel[1]
[1]Universiy of Southern California, Los Angeles, USA [2]Intel Labs, San Diego, USA
{sreetama,pabeerel}@usc.edu souvikk.kundu@intel.com

Abstract

The growing concern about data privacy has led to the development of private inference (PI) frameworks in client-server applications which protects both data privacy and model IP. However, the cryptographic primitives required yield significant latency overheads which limits their widespread application. At the same time, changing environments demand PI services to be robust against various naturally occurring and gradient-based perturbations. Despite several works focused on the development of latency-efficient models suitable for PI, the impact of these models on robustness has remained unexplored. Towards this goal, this paper presents RLNet, a class of models that can yield latency improvement via the reduction of high-latency ReLU operations while improving the model performance on both clean and corrupted images. In particular, RLNet models provide a "triple win ticket" of improved classification accuracy on clean, naturally perturbed, and gradient-based perturbed images using a shared-mask shared-weight architecture with over an order of magnitude fewer ReLUs than baseline models. To demonstrate the efficacy of RLNet, we perform extensive experiments with ResNet and WRN model variants on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets. Our experimental evaluations show that RLNet can yield models with up to 11.14× *fewer ReLUs, with accuracy close to the all-ReLU models, on clean, naturally perturbed, and gradient-based perturbed images. Compared with the SoTA non-robust linearized models at similar ReLU budgets, RLNet achieves an improvement in adversarial accuracy of up to* ∼47%*, in naturally perturbed accuracy of up to* ∼16.4%*, while improving clean image accuracy up to* ∼1.5%*. Code is available at:* https://github.com/sreetamasarkar/rlnet.

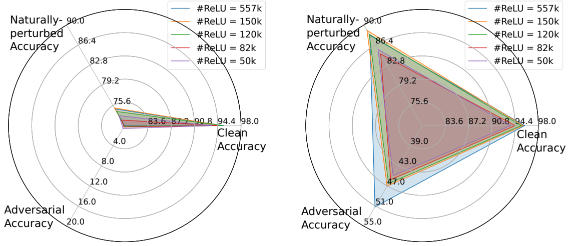


Figure 1. Clean, naturally-perturbed, and adversarial accuracy of ResNet18 for non-robust linearized models (left) and RLNet models (right) on CIFAR-10 for different ReLU budget. SoTA linearized models lack robustness against natural and adversarial perturbations, while RLNet performs well on all three fronts outperforming its non-robust counterpart even in clean accuracy. (Note: all axes do not have same scale)

1. Introduction

In recent years, there has been a growing concern about data privacy, particularly in applications that rely on Machine Learning as a Service (MLaaS) [20, 26] in which user data is sent to the cloud to perform inference. This has led to the development of private inference (PI) frameworks, where the server performs computations on a client's encrypted data, preserving both model and data privacy. Nevertheless, the implementation of the required cryptographic protocols including homomorphic encryption [31] and secure multi-party communication [28] dramatically increase the computation and communication latency, making it impractical to perform private inference on large scale models. A deeper look into the inference latency of these cryptographic primitives applied to CNNs suggest that the overhead associated with the ReLU layers is ∼340× more than that of the convolutional layers [21]. This has sparked an interest in the reduction of ReLU non-linearity for latency-efficient private inference of CNNs.

Techniques for ReLU reduction without compromising accuracy has been extensively studied in the literature [6, 16, 21]. However, these models lack robustness against naturally or adversarially perturbed images, necessary to ensure trustworthy deployment, as demonstrated in Figure 1 (left). The natural perturbations may stem from seasonal changes (summer versus winter), environmental conditions (rain, fog, and snow), and/or camera noise and blurring effects [12]. Adversarial perturbations [9, 27] are carefully

[*] Equally contributing authors.

crafted using gradient-based attacks such that they are not noticeable to the human eye but cause misclassifications. These issues are particularly important for applications such as household robots [29], which operates in dynamic environments such as changing lighting conditions, and collects enormous amount of user's sensitive personal data, necessitating PI. To the best of our knowledge, robustness against natural and adversarial perturbations of the partially linearized models is yet to be explored. With this motivation, the objective of this paper is to understand, analyze, and improve the relevant trade-offs as well as to build a training framework that guarantees both privacy and robustness.

The failure of models on naturally perturbed images is caused by a shift in the distribution between the inference and training data. To improve generalization to these distribution shifts, data augmentation techniques [13, 39] are commonly used, but they cannot defend against strong gradient-based perturbations. Adversarial training [27, 34] is the most popular and effective defense against gradient-based adversarial attacks. However, the improved adversarial accuracy does not guarantee improved performance on natural perturbations and is often achieved at the cost of a significant reduction in clean accuracy [47, 48]. One means of achieving high clean, naturally perturbed, and adversarial accuracy is to have multiple models in the server and switch models in real time during inference. However, this triples off-line processing and storage requirements, induces energy and latency overheads when switching between models, and increases the logistics of maintaining and delivering multiple models to the customer. To solve this problem, this paper proposes RLNet, a class of shared-mask shared-weight conditional models, that provides a configurable trade-off between accuracy and robustness while improving latency via reduced ReLU operations. As shown in Figure 1 (right), RLNet maintains close to baseline accuracies on clean, naturally perturbed, and adversarial images with up to an $11.14\times$ reduction in ReLUs.

Our Contributions Our contributions in this work are three-fold. *Firstly,* we propose a training framework that achieves a *"triple win ticket"*, that is, improved accuracy on clean, naturally perturbed, and adversarial images. Specifically, we implement a conditional learning [22, 38] strategy using dual Batch Normalization (BN) to build a multi-path model, to retain performance on all three fronts. Unlike [22, 38], our model requires no additional layers or parameters and, hence, incurs no increase in computational overhead. *Secondly,* we develop a fine-tuning framework for partial ReLU (PR) model distillation from an all ReLU (AR) model such that it provides both accuracy and robustness with a reduced PI latency budget. We present RLNet, a class of conditionally trained PR models, efficiently fine-tuned to yield improved performance. *Finally,* we conduct extensive experimental evaluations to demonstrate the efficacy of the proposed training framework across various models on multiple datasets. Compared with the SoTA non-robust linearized model [21], at similar ReLU budgets, RLNet achieves an improved adversarial accuracy of up to $\sim$47%, naturally perturbed accuracy up to $\sim$16.4%, while improving clean image accuracy up to $\sim 1.5\%$.

2. Related Work

2.1. Model Robustness

Robustness against Natural Perturbations Data augmentation methods [13, 39] are commonly used to improve generalization to distribution shifts. They are easy to implement, have low computational overhead, and often improve clean accuracy. A number of simple augmentation techniques, including Cutout [5], occluding random portions in an image, CutMix [44], replacing sections of an image with another image, and MixUp [46], generating an image using a linear combination of two different images, have yielded promising results. Augmix [13], which randomly samples a set of augmentations and linearly combines the augmented images, has shown to be one of the most effective against the common image corruptions benchmarks like CIFAR-10-C and ImageNet-C [12]. DeepAugment [14] generates augmented images by distorting the weights of image-to-image models. Augmax [39] tries to further improve Augmix by learning the mixing coefficients of augmented images to generate harder samples. In this work, we generate augmented images to train our conditional model inspired from the strategy proposed in Augmix [13].

Adversarial Robustness Adversarial images may be viewed as augmented images, with l_p-norm bounded perturbations, generated using strong gradient-based attacks. Data augmentation techniques like AdversarialAugment,[1], an enhancement of DeepAugment [14], are designed to achieve better robustness against natural as well as adversarial perturbations. Training with adversarial images or adversarial training (AT) [9, 19, 27, 37] is the most commonly used defense against adversarial attacks, although it causes a degradation in clean accuracy. The increased training overhead of standard AT techniques like PGDAT [27] has led to the development of compute-efficient alternatives like FreeAT [34] and FastAT [40]. Recently, robust distillation methods [8, 47, 48] have proved to achieve better adversarial robustness compared to standard AT. While TRADES [47] uses a self-distillation technique, which utilizes clean predictions of the same model to learn its adversarial predictions, ARD [8] and RSLAD [48] leverage knowledge distillation to learn from an adversarially robust teacher. We perform robust distillation inspired from RSLAD, to generate robust and latency-efficient PR models from robust AR models.

2.2. Efficient Private Inference

PI frameworks use cryptograhic protocols such as Homomorphic Encryption (HE) [17, 35] and Additive Secret Sharing (ASS) [6, 25]. Fully HE based protocols like CryptoNets [7], CHET [4], TAPAS [33] incur huge computation and communication latency, limiting their applications to networks that are only a few layers deep. Gazelle [17], Delphi [35], Cheetah [32] uses HE for linear operations such as convolution and fully-connected layers, while Garbled Circuits (GC) [42] are used for non-linear ReLU operations on the client's encrypted data. Delphi [35] introduces an online-offline topology, where the client-data-independent components are pre-computed during an offline phase, enabling online plaintext computation for linear operations. Although this reduces online latency due to linear operations, the input dependent ReLU computation using GC still causes high inference latency. This necessitates either removing ReLU operations [2, 21] or replacing them with some other compute-efficient alternatives like polynomial or quadratic functions [7, 25, 35]. Various approaches for ReLU reduction have been proposed in literature. They range from manually dropping ReLU layers from existing models [16], l_1-regularization based approaches [2] to evolutionary neural architecture search (NAS) techniques [6] for ReLU reduction. Kundu et al. [21, 24] proposed a 3-stage training approach that meets a target ReLU budget for negligible accuracy reduction. In this paper, we enhance the training pipeline proposed in [21] for robust generalization.

2.3. Conditional Learning

Conditional learning involves training a single model with multiple paths that can be selectively enabled during inference. Conditional models have been used to provide an in-situ trade-off between efficiency and accuracy [43] or accuracy and adversarial robustness [22, 38]. OAT [38] uses a parameter lambda to control the trade-off between clean and adversarial accuracy through feature-wise linear modulation (FiLM [30]) layers, conditioned on lambda, and dual BN [41]. To remove the FiLM latency overhead, recently, a few works have proposed [22, 23] weight-conditioned learning for accuracy robustness trade-off. We propose a conditional model using dual BN for accuracy robustness trade-off in PR models with no extra parameters or computational overhead.

3. Proposed Approach

We propose a three-stage pipeline for training RLNet models that consists of training a robust AR teacher, generating a ReLU mask for achieving a target number of ReLU operations in the PR model, and finally, fine-tuning the PR model with the ReLU mask frozen to reduce the performance gap with the AR model. RLNets have two different modes of operation: *normal mode* which targets clean and naturally-perturbed images, and *adversarial mode* which targets adversarial images. The model may be equipped to automatically switch modes based on prediction confidence, as suggested in [36]. We leverage data augmentation, adversarial distillation, and dual BN to achieve our three-fold objective: clean accuracy (CA), naturally perturbed accuracy (NPA), and adversarial accuracy (AdvA).

3.1. Data Augmentation

In our framework, we generate augmented images (x_{aug}) using Augmix [13] to improve generalization against natural perturbations. Augmix (augment+mix) involves applying a chain of simple augmentation operations including translation, rotation, shear, auto-contrast, and linearly combining or mixing the augmented images. Augmentations based on brightness, contrast, colour, used in CIFAR-10-C or CIFAR-100-C, which are used for robustness evaluation, are excluded from the set of augmentations during training. Images from multiple augmentation chains (by default, 3) are then mixed together using a set of convex coefficients randomly sampled from a Dirichlet distribution.

3.2. Adversarial Training

We generate adversarial images (x_{adv}) using the PGD attack [27] to use for robust teacher training. Attacked images are generated by finding perturbations within the maximum perturbation strength ϵ that maximizes the cross-entropy (CE) loss, as shown in Equation 1.

$$x_{adv} = \underset{||x_{adv}-x||_p \leq \epsilon}{\arg\max} \; CE(\phi_T^{\lambda=1}(x_{adv}), y) \tag{1}$$

The AR teacher model is represented as ϕ_T and the PR student model is denoted as ϕ_S. $\lambda = 0$ denotes the path trained using clean and augmented images and $\lambda = 1$ denotes the path trained using adversarial images. During training of the PR model from the AR model through distillation, x_{adv} is generated by maximizing the KL divergence loss, inspired from RSLAD [48], as shown in Equation 2.

$$x_{adv} = \underset{||x_{adv}-x||_p \leq \epsilon}{\arg\max} \; KL(\phi_S^{\lambda=1}(x_{adv}), \phi_T^{\lambda=0}(x)) \tag{2}$$

3.3. Dual Batch Normalization

Previous work [38, 39, 41] has pointed out that separation of the BN statistics is critical for a single model to perform well on both clean and adversarial images. BN relies on the fact that input images have the same underlying distribution. Because the distribution of adversarial images is significantly different from that of clean images, adversarial training fails to perform well on clean images. Xie et al. [41] demonstrated that adversarial images can in fact improve clean accuracy only if the BN statistics of adversarial images do not interfere with clean training.

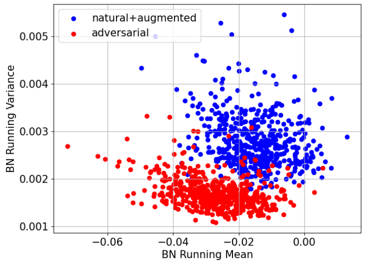


Figure 2. Running mean and variance of the last BN layer of ResNet18 trained on Tiny-ImageNet using dual BN

Inspired by [22], we build a model with two distinct paths for $\lambda = 0$ and $\lambda = 1$, for two different modes of operation, which differ only in BN layers. BN_c denotes the BN layers for clean, augmented, and BN_a for adversarial images (Figure 3). [41] uses a triple BN formulation when training with clean, augmented and adversarial images. We do not use separate BN for clean and augmented images, because we observe that using simple augmentations actually makes the dataset more diverse and improves clean accuracy rather than adversely affecting it. The BN statistics of a ResNet18 model trained on Tiny-ImageNet, illustrated in Figure 2, clearly shows two distinct clusters, justifying this choice. We further demonstrate that a triple BN formulation, separating clean and augmented images, is redundant (see Section 4.3).

3.4. Training Framework

3.4.1 Training a Robust Teacher

The PR model is distilled from a teacher model, which has the same architecture as the PR model but with all ReLUs present. Therefore, the first step is to train a teacher model that is robust against natural as well as adversarial perturbations and still retains its clean accuracy.

The teacher model is trained on a classification task with input x and corresponding labels y. x_{aug} and x_{adv} are generated from x as described in Sections 3.1 and 3.2 respectively. ϕ_T is trained by minimizing the loss function $\mathcal{L}_T$ with respect to x, x_{aug}, and x_{adv}, as shown in Equation 3, where CE denotes cross-entropy loss.

$$\begin{aligned}\mathcal{L}_T = (CE(\phi_T^{\lambda=0}(x), y) + CE(\phi_T^{\lambda=0}(x_{aug}), y) + \\ CE(\phi_T^{\lambda=1}(x_{adv}), y))/3\end{aligned} \tag{3}$$

3.4.2 ReLU Mask Identification

Given a global ReLU budget, the number of ReLU units per layer has to be determined. [21] found an inverse relation between pruning sensitivity and ReLU sensitivity and formulated an approach to determine the layerwise ReLU budget. Following [21], we allocate the layerwise ReLU count, given by r_l for ReLU layer l. A binary ReLU mask is initialized in layer l, with r_l 1's in random locations, where 1's and 0's indicate the presence or absence of ReLU units. ReLU returns 0 if the activation value is negative but retains the original value if the activation value is positive. Hence, the absence of a ReLU unit only makes a difference for negative activation values. At the end of each epoch, the ReLU mask is updated with 1's in the top r_l locations, where the absolute difference between the post ReLU activation map of the AR and the PR model is the highest. During mask search, the masks and weights of the PR model are updated in parallel. The model weights are updated for each minibatch using the loss function $\mathcal{L}_{stage2}$ described in Eq. 8. Note, the activation maps for absolute difference calculation and mask updating are obtained using clean images only.

3.4.3 Three-way Robust Distillation

We formulate three different loss functions for our dual BN framework $\mathcal{L}_{ce}$, $\mathcal{L}_{kl}$ and $\mathcal{L}_{PRAM}$ based on CE loss, KL divergence based distillation loss [15], and post-ReLU activation mismatch (PRAM) loss [21].

$$\begin{aligned}\mathcal{L}_{ce} =& (CE(\phi_S^{\lambda=0}(x), y) + CE(\phi_S^{\lambda=0}(x_{aug}), y) + \\ & CE(\phi_S^{\lambda=1}(x_{adv}), y))/3\end{aligned} \tag{4}$$

$$\begin{aligned}\mathcal{L}_{kl} =& (KL(\phi_S^{\lambda=0}(x), \phi_T^{\lambda=0}(x)) + KL(\phi_S^{\lambda=0}(x_{aug}), \\ & \phi_T^{\lambda=0}(x)) + KL(\phi_S^{\lambda=1}(x_{adv}), \phi_T^{\lambda=0}(x)))/3\end{aligned} \tag{5}$$

Each of these loss functions can have three different components corresponding to three different versions of the input, x, x_{aug} and x_{adv}, and their corresponding paths through the model. While $\mathcal{L}_{ce}$ ensures learning from the original hard labels y, $\mathcal{L}_{kl}$ enforces that the student model learns the output for clean, augmented, and adversarial images through distillation from the clean predictions of the teacher.

PRAM loss [21] is given by

$$PRAM(x) = \left\| \frac{\Psi_{PR}^m(x)}{\|\Psi_{PR}^m(x)\|_2} - \frac{\Psi_{AR}^m(x)}{\|\Psi_{AR}^m(x)\|_2} \right\|_2 \tag{6}$$

where $\Psi_{PR}^m(x)$ and $\Psi_{AR}^m(x)$ denote the m^{th} pair of post ReLU activation maps for input x for the PR and the AR model. $\mathcal{L}_{PRAM}$ ensures feature similarity between the AR and PR model and is used only during the final fine-tuning to reduce the performance gap. We consider PRAM loss for x and x_{aug}, as we find PRAM loss for x_{adv} do not provide significant benefits (see Section 4.3).

$$\mathcal{L}_{PRAM} = 0.5 \times (PRAM(x) + PRAM(x_{aug})) \tag{7}$$

The necessity of each of these loss functions has been justified through ablation studies in Section 4.3. Weight updates

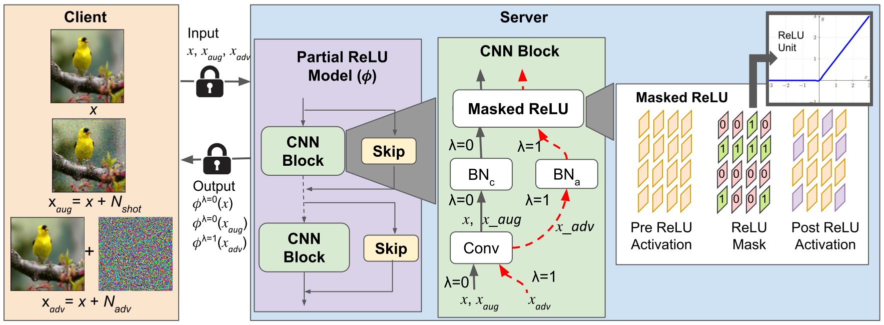


Figure 3. RLNet framework: model architecture and inference path. Here, $\lambda = 0$ and 1 correspond to clean and adversarial paths respectively. We use the same path as clean for classifying against natural perturbations. This means that unless the perturbation is attacker-driven, we use the $\lambda = 0$ path for inference.

for the student model ϕ_S during mask identification and final fine-tuning are performed through optimizing $\mathcal{L}_{stage2}$ and $\mathcal{L}_{stage3}$ respectively.

$$\mathcal{L}_{stage2} = \mathcal{L}_{kl} + \mathcal{L}_{ce} \tag{8}$$

$$\mathcal{L}_{stage3} = \mathcal{L}_{kl} + \mathcal{L}_{ce} + \beta * \mathcal{L}_{PRAM} \tag{9}$$

4. Experimental Results

4.1. Experimental Setup

Models and Dataset We evaluate the efficacy of RLNet on three datasets, namely, CIFAR-10, CIFAR-100 [18], and Tiny-ImageNet [10] with ResNet-18, ResNet-34 [11], and WRN-22-8 [45]. Evaluation against natural perturbations is performed using the common image corruption benchmarks: CIFAR-10-C, CIFAR-100-C, and Tiny-ImageNet-C [12], which are generated by adding fifteen different corruptions (for example, Gaussian noise, shot noise, contrast) at five severity levels to the original dataset.

Evaluation Metrics Performance on clean, naturally-perturbed, and adversarial images is evaluated using CA, NPA, and AdvA respectively. NPA is the average accuracy over all 15 corruptions, where accuracy on each corruption is averaged over five severity levels. In addition, we also report mean corruption error or mCE on naturally perturbed samples. mCE on CIFAR-10-C and CIFAR-100-C is evaluated as $1 - NPA$, whereas for Tiny-ImageNet the error for each type of corruption is first normalized by the corruption error of a baseline model and then averaged [12, 13, 39]. CA and AdvA are given by the Top-1 accuracies evaluated on the original dataset and using PGD-7 attack [27], if not mentioned otherwise. Communication savings is the ratio of communication cost of an AR to that of a PR model.

Training Hyperparameters The baseline AR model training and final fine-tuning are performed for 240 and 120 epochs on CIFAR and Tiny-ImageNet datasets, using SGD optimizer, with a starting learning rate (lr) of 0.05 for baseline training and 0.01 for fine-tuning. A step lr decay policy is followed during both, where lr drops by a factor of 0.1 at 62.5%, 75%, and 87.5% of the total training epochs. Mask search is performed for 150 and 100 epochs on CIFAR and Tiny-ImageNet, without any drop in learning rate. Distillation temperature is maintained at 4.0, unless otherwise stated. $\beta = 1000$ is used in Equation 9. Data augmentations are generated using the default parameters as in [13]. Adversarial augmentations during training are generated using PGD-7, where the attack is performed for 7 iterations with maximum perturbation strength $\epsilon = 8/255$ and step size $\alpha = 2/255$. Attack evaluations are also performed using FGSM with $\epsilon = 8/255$, PGD-20, with identical parameters as PGD-7 with 20 steps, and Auto-PGD [3], a variant of AutoAttack, with $\epsilon = 8/255$ and 100 iterations.

4.2. Results and Analysis

AR Model In Figure 4, we compare our conditional teacher training approach with individual SoTA training methods using natural, augmented, and adversarial images. Our approach achieves improvement in CA by 0.65% over a standard trained model, NPA by 0.89% over a model trained using Augmix [13], and AdvA by 2.68% over a model trained using PGDAT [27] for ResNet18 on CIFAR-10. While TRADES [47] achieves the best AdvA among these training methods, however costs a degradation of 10.72% in CA.

PR Model The performance of RLNet on CIFAR-10, CIFAR-100, and Tiny-Imagenet for different ReLU budget is presented in Table 1. On CIFAR-10, our method achieves 3.7× ReLU reduction with almost no degradation in CA and NPA and AdvA degradation of 4.14%. We achieve up to 5.8× ReLU reduction on CIFAR-100 dataset with a nominal reduction in CA of 0.4%, AdvA of 3.84%, and with no degradation in NPA. In fact we observe that some amount

Dataset	Model	State	RLNet				SeNet [21]				#ReLU	Comm.
			CA(%)	NPA(%)	mCE↓	AdvA(%)	CA(%)	NPA(%)	mCE↓	AdvA(%)	(k)	Savings
CIFAR-10	ResNet18	AR	**95.88**	**88.15**	**0.12**	**51.18**	95.23	74.98	0.25	0.01	557	1×
		PR	**93.71**	**85.62**	**0.14**	**44.95**	93.41	73.78	0.26	0.58	50	11.14×
		PR	**94.43**	**84.83**	**0.15**	**45.75**	94.24	72.99	0.27	0.28	82	6.79×
		PR	**95.36**	**88.35**	**0.12**	**46.52**	95.12	74.53	0.25	0.13	120	4.64×
		PR	**95.67**	**89.04**	**0.11**	**47.04**	95.0	75.17	0.25	0.12	150	3.71×
CIFAR-100	ResNet18	AR	**78.68**	**63.31**	**0.37**	**26.99**	77.59	48.67	0.51	0.04	557	1×
		PR	**74.87**	**60.72**	**0.39**	**22.07**	74.51	48.71	0.51	0.07	50	11.14×
		PR	**77.00**	**64.45**	**0.36**	**22.99**	76.67	49.90	0.50	0.09	100	5.57×
	ResNet34	AR	**79.71**	**65.25**	**0.35**	**26.57**	78.25	49.63	0.50	0.29	967	1×
		PR	**73.72**	**61.27**	**0.39**	**22.03**	72.84	51.67	0.48	0.30	80	12×
		PR	75.76	**63.90**	**0.36**	**23.17**	**76.07**	51.59	0.48	0.26	200	4.8×
	WRN22-8	AR	**80.96**	**66.38**	**0.34**	**29.69**	79.77	49.62	0.50	0.03	1393	1×
		PR	**80.53**	**66.58**	**0.33**	**25.85**	79.75	50.73	0.49	0.04	240	5.8×
		PR	**80.66**	**67.49**	**0.33**	**25.61**	79.95	51.11	0.49	0.05	300	4.64×
Tiny-ImageNet	ResNet18	AR	**67.22**	**37.99**	**0.81**	**20.52**	66.1	26.91	0.96	0.08	2228	1×
		PR	58.72	**31.31**	**0.90**	**12.85**	**59.31**	24.01	0.99	0.15	150	14.85×
		PR	**66.57**	**39.45**	**0.79**	**15.99**	66.18	27.54	0.95	0.19	300	7.43×

Table 1. Performance evaluation of RLNet models on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets and comparison with SeNet [21]. The higher accuracy and lower mCEs are highlighted in bold.

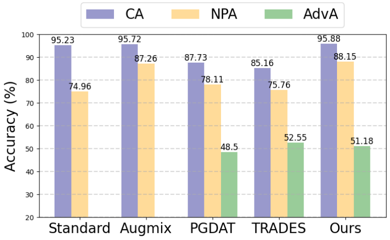


Figure 4. CA, NPA, and AdvA for ResNet18 on CIFAR-10 for different training modes

of ReLU reduction actually boosts NPA. NPA for ResNet18 with 150k ReLU on CIFAR-10-C, 100k ReLU on CIFAR-100-C, and 300k ReLU on Tiny-ImageNet-C are higher than the baseline AR models. For more aggressive ReLU pruning up to 11.14×, our model suffers a degradation in CA of only 2.17%, NPA of 2.52%, and AdvA of 6.23% on CIFAR-10. For the same ReLU budget on CIFAR-100, degradation in CA, NPA, and AdvA are 3.81%, 2.59%, and 4.92% respectively. For Tiny-Imagenet, we achieve 7.43× ReLU reduction with close to baseline CA, NPA and AdvA reduction of 4.5%. We try to ensure minimum degradation in CA through our choice of hyperparameters and loss function, as discussed in Section 4.3. This explains the higher degradation in AdvA compared to CA and NPA.

We also compare RLNet models with a SOTA linearized network SeNET [21]. RLNet consistently outperforms SeNet not only in NPA and AdvA but also in CA. RLNet achieves an improvement in CA over SeNet by 0.67% and 0.3% for ResNet18 on CIFAR-10 for ReLU reduction of 3.7× and 11.14× respectively, 0.78% for WRN22-8 on CIFAR-100 and 0.39% for ResNet18 on Tiny-Imagenet, for a ReLU reduction of 5.8× and 7.43× respectively. RLNet yields up to 16.38% improvement in NPA and 47% improvement in AdvA over SeNet models.

4.3. Ablation Studies

Study of Robust Distillation Loss Adversarial robust distillation techniques [8, 48] demonstrate that learning from teacher predictions significantly improves adversarial robustness as compared to regular AT using hard labels. The increase in adversarial robustness is accompanied with a drop in clean accuracy. We try to build a robust distillation approach for our dual BN framework that prioritizes CA, and at the same time achieves a reasonable trade-off between CA, NPA, and AdvA. We formulate three different robust training techniques, inspired from PGD AT [27], ARD [8], and RSLAD [48]. Train$_{CE}$ uses CE loss for x, x_{aug}, and x_{adv}, similar to PGD AT [27] for x and x_{adv}, which requires no teacher model. Train$_{CEKL}$ uses distillation only for x_{adv}, following ARD [8], and CE loss for x and x_{aug}. Train$_{KL}$, inspired from RSLAD [48], uses distillation for all three inputs x, x_{aug}, and x_{adv}, where the model learns from the clean predictions of the teacher. The loss functions for these robust training methods are presented in Table 2.

Table 3 presents the evaluation results for our PR model, trained using these robust training techniques. We observe that KL divergence loss for x and x_{aug} improves CA and NPA by ∼3% compared with CE loss. Hence, we formulate the loss $\mathcal{L}_{kl}$ (Equation 5) according to Train$_{KL}$. The degradation in AdvA for Train$_{KL}$ may be attributed to the choice of distillation temperature, favourable towards CA

Method	**Loss**
Train$_{CE}$	$(CE(S^{\lambda=0}(x),y)+CE(S^{\lambda=0}(x^{aug}),y)+CE(S^{\lambda=1}(x^{adv}),y))/3$
Train$_{CEKL}$	$(CE(S^{\lambda=0}(x),y)+CE(S^{\lambda=0}(x^{aug}),y)+KL(S^{\lambda=1}(x^{adv}),T^{\lambda=0}(x)))/3$
Train$_{KL}$	$(KL(S^{\lambda=0}(x),T^{\lambda=0}(x))+KL(S^{\lambda=0}(x^{aug}),T^{\lambda=0}(x))+KL(S^{\lambda=1}(x^{adv}),T^{\lambda=0}(x)))/3$

Table 2. Robust distillation losses for conditional model training

and NPA, as discussed in the next section.

State	**Accuracy**			**#ReLU**	**Loss**
	CA(%)	**NPA(%)**	**AdvA(%)**	**(k)**	
AR	78.68	63.31	26.99	557	-
PR	70.21	58.08	**23.91**	50	Train$_{CE}$
PR	70.53	58.75	22.21	50	Train$_{CEKL}$
PR	**73.47**	**61.13**	21.82	50	Train$_{KL}$

Table 3. Study of robust distillation techniques for ResNet18 PR models on CIFAR-100

Choice of Distillation Temperature The choice of distillation temperature plays a critical role in the trade-off between CA and AdvA. In line with RSLAD [48], we observe that using a distillation temperature of 1.0 results in improved AdvA. However, this reduces CA, both with and without PRAM loss, as observed in Table 4. Since we prioritize CA, we choose a distillation temperature of 4.0.

Accuracy			**Loss**	**Temperature**
CA(%)	**NPA(%)**	**AdvA(%)**		
73.47	**61.13**	21.82	$\mathcal{L}_{kl}$	4.0
71.08	58.6	**23.32**	$\mathcal{L}_{kl}$	1.0
74.16	**61.19**	21.81	$\mathcal{L}_{kl}+\mathcal{L}_{PRAM}$	4.0
73.5	59.11	**24.55**	$\mathcal{L}_{kl}+\mathcal{L}_{PRAM}$	1.0

Table 4. Study of distillation temperature for ResNet18 PR models (ReLU count = 50k) on CIFAR-100

Study of Robust Feature Similarity Loss In this section, we explore the effectiveness of PRAM loss [21] in our robust training setup. PRAM(x), PRAM(x_{aug}), and PRAM(x_{adv}) are the PRAM losses with original, augmented, and adversarial images as input. In Table 5, we perform an ablation to understand the contribution of each of these PRAM loss terms. We observe that the presence or absence of PRAM loss causes a variation in adversarial accuracy by less than 0.5%. PRAM(x), PRAM(x_{aug}), and PRAM(x_{adv}) boosts CA, NPA, as well as AdvA, in the absence of $\mathcal{L}_{ce}$. However, when we incorporate $\mathcal{L}_{ce}$, PRAM(x_{adv}) is found to degrade both CA and AdvA. Hence, we only use PRAM(x) and PRAM(x_{aug}) for feature similarity during fine-tuning (Equation 7).

$\mathcal{L}_{kl}$	$\mathcal{L}_{ce}$	$PRAM$ (x)	$PRAM$ (x_{aug})	$PRAM$ (x_{adv})	**Accuracy**		
					CA (%)	**RA (%)**	**AdvA (%)**
✓	✗	✗	✗	✗	73.47	61.13	21.82
✓	✗	✓	✗	✗	73.85	60.48	21.73
✓	✗	✓	✓	✗	74.16	61.19	21.81
✓	✗	✓	✓	✓	74.20	**61.53**	**22.19**
✓	✓	✓	✓	✗	**74.87**	60.72	22.07
✓	✓	✓	✓	✓	74.28	61.12	21.95

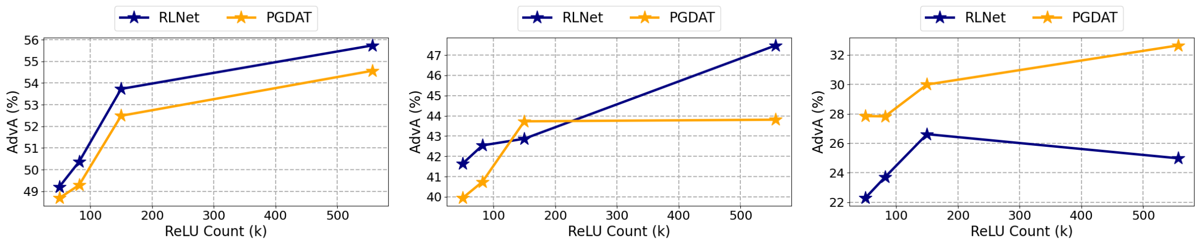
Table 5. PRAM ablation for ResNet18 PR models (ReLU count = 50k) on CIFAR-100

Necessity Analysis of CE Loss In this section, we analyze whether $\mathcal{L}_{ce}$ helps or hurts performance when used together with $\mathcal{L}_{kl}$ and $\mathcal{L}_{PRAM}$. In Table 6, we present results with and without using $\mathcal{L}_{ce}$ on a number of models and datasets. We observe that $\mathcal{L}_{ce}$ improves CA as well as AdvA for both ResNet18 and ResNet34 models on CIFAR-100, whereas it retains the same CA for ResNet18 on CIFAR-10 dataset. Since our main focus is to minimize degradation in CA, we incorporate $\mathcal{L}_{ce}$ in our training framework.

Model/ Dataset	**State**	**Accuracy**			**#ReLU**	$\mathcal{L}_{ce}$
		CA(%)	**NPA(%)**	**AdvA(%)**	**(k)**	
ResNet18 CIFAR-10	AR	95.88	88.15	51.18	557	-
	PR	**93.71**	**85.62**	**44.95**	50	✗
	PR	**93.70**	85.13	43.57	50	✓
ResNet18 CIFAR-100	AR	78.68	63.31	26.99	557	-
	PR	74.16	**61.19**	21.81	50	✗
	PR	**74.87**	60.72	**22.07**	50	✓
ResNet34 CIFAR-100	AR	79.71	65.25	26.57	967	-
	PR	73.30	61.23	21.69	80	✗
	PR	**73.72**	**61.27**	**22.03**	80	✓

Table 6. Fine-tuning PR model with and without CE loss

Performance on Other Attacks We evaluate RLNet models against existing SOTA white-box attacks FGSM [9], PGD-20 [27] and Auto-PGD, a variant of AutoAttack [3]. For ResNet18 on CIFAR-10, we train a robust teacher using PGDAT [27] and distill PR models for different ReLU budgets using RSLAD [48]. In Figure 5, we compare these PR models, distilled from a PGD trained teacher, and focused solely on improving AdvA, to our RLNet models for different ReLU budgets. We still outperform PGDAT PR models for all ReLU counts on FGSM attacked images



(a) FGSM (b) PGD-20 (c) Auto-PGD

Figure 5. Performance comparison of RLNet vs ResNet-18 models, distilled from a PGD trained teacher, for different ReLU budgets, on FGSM [9], PGD-20[27] and Auto-PGD [3] attacked images on CIFAR-10 dataset

and for ReLU count lower than 100k on PGD-20 generated images. PGDAT models demonstrate higher robustness against Auto-PGD on CIFAR-10. Attack evaluation performance for RLNet models on CIFAR-100 are presented in Figure 6.

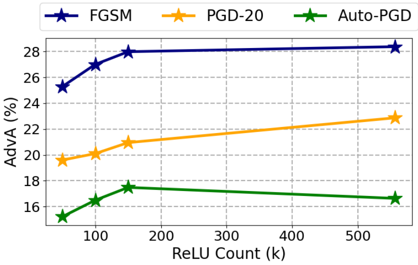


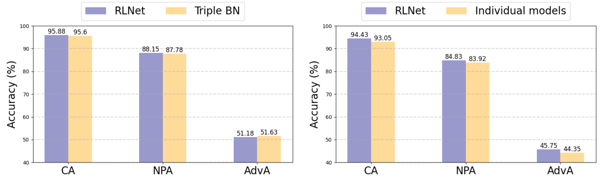
Figure 6. Performance of ResNet18 for different ReLU budget on FGSM [9], PGD-20[27], and AutoPGD [3] attack generated CIFAR-100 test images

Sufficiency Test of Dual BN We demonstrate results using a triple BN formulation, using three separate BN for clean, augmented and adversarial images and compare it with dual BN in Figure 7a. We observe that triple BN yields no extra benefits, as hypothesized in Section 3.3

Sufficiency Analysis of Single ReLU Mask The performance of RLNet models is compared with three separate models, trained using standard, Augmix [13] and PGDAT [27] training, for the same ReLU count, in Figure 7b. RLNet achieves improved accuracy over separately-trained individual models in all three cases. This confirms that a shared mask shared weight dual BN model is sufficient to achieve improved performance on all three fronts.

5. Conclusion

The latency overhead for PI in CNNs can be largely attributed to the presence non-linear ReLU units. Latency efficient PI methods have devised ReLU reduction techniques in CNNs. However, robustness of these partially linearized models remain unexplored. In this paper, we propose RLNet, a class of shared-mask shared-weight conditional models that yields close to baseline accuracy against clean, naturally perturbed as well as adversarial images with up to 11.14× fewer ReLU. Compared with its SoTA non-robust counterpart, RLNet models improve adversarial accuracy up to $\sim$47%, naturally perturbed accuracy up to $\sim$16.4%, while improving clean image accuracy up to $\sim$1.5%. Exploring robustness of vision transformer models for latency efficient PI can be an interesting future research in this direction.



(a) RLNet vs Triple BN (b) RLNet vs Individual Models

Figure 7. (a) Dual vs triple BN for ResNet18 on CIFAR-10 (b) CA, NPA, and AdvA of RLNet vs separate ResNet18 PR models (ReLU count=82k) trained using standard, Augmix [13] and PGDAT [27], respectively.

References

[1] Dan Andrei Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andr'as Gyorgy, Timothy A. Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *ArXiv*, 2021. 2

[2] Minsu Cho, Ameya Joshi, Siddharth Garg, Brandon Reagen, and Chinmay Hegde. Selective network linearization for efficient private inference. In *International Conference on Machine Learning*, 2022. 3

[3] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2020. 5, 7, 8

[4] Roshan Dathathri, Olli Saarikivi, Hao Chen, Kim Laine, Kristin Lauter, Saeed Maleki, Madanlal Musuvathi, and Todd Mytkowicz. CHET: an optimizing compiler for fully-homomorphic neural-network inferencing. In *Proceedings*

of the 40th ACM SIGPLAN conference on programming language design and implementation, 2019. 3

[5] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *ArXiv*, 2017. 2

[6] Zahra Ghodsi, Akshaj Kumar Veldanda, Brandon Reagen, and Siddharth Garg. CryptoNAS: Private inference on a ReLU budget. *Neural Information Processing Systems*, 2020. 1, 3

[7] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, 2016. 3

[8] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *AAAI Conference on Artificial Intelligence*, 2019. 2, 6

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, 2014. 1, 2, 7, 8

[10] Lucas Hansen. Tiny imagenet challenge submission. *CS 231N*, 5, 2015. 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 5

[12] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ArXiv*, 2019. 1, 2, 5

[13] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *ArXiv*, 2019. 2, 3, 5, 8

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020. 2

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[16] Nandan Kumar Jha, Zahra Ghodsi, Siddharth Garg, and Brandon Reagen. DeepReDuce: ReLU reduction for fast private inference. In *International Conference on Machine Learning*, 2021. 1, 3

[17] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, 2018. 3

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[19] Souvik Kundu, Mahdi Nazemi, Peter A. Beerel, and Massoud Pedram. Dnr: A tunable robust pruning framework through dynamic network rewiring of dnns. *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020. 2

[20] Souvik Kundu, Qirui Sun, Yao Fu, Massoud Pedram, and Peter A. Beerel. Analyzing the confidentiality of undistillable teachers in knowledge distillation. *Advances in Neural Information Processing Systems*, 2021. 1

[21] Souvik Kundu, Shunlin Lu, Yuke Zhang, Jacquelin Liu, and Peter A. Beerel. Learning to linearize deep neural networks for secure and efficient private inference. *ICLR*, 2023. 1, 2, 3, 4, 6, 7

[22] Souvik Kundu, Sairam Sundaresan, Massoud Pedram, and Peter A. Beerel. FLOAT: Fast learnable once-for-all adversarial training for tunable trade-off between accuracy and robustness. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023. 2, 3, 4

[23] Souvik Kundu, Sairam Sundaresan, Sharath Nittur Sridhar, Shunlin Lu, Han Tang, and Peter A. Beerel. Sparse mixture once-for-all adversarial training for efficient in-situ trade-off between accuracy and robustness of DNNs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. 3

[24] Souvik Kundu, Yuke Zhang, Dake Chen, and Peter A. Beerel. Making models shallow again: Jointly learning to reduce non-linearity and depth for latency-efficient private inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023. 3

[25] Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. Oblivious neural network predictions via MiniONN transformations. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017. 3

[26] Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Undistillable: Making a nasty teacher that cannot teach students. *arXiv preprint arXiv:2105.07381*, 2021. 1

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, 2017. 1, 2, 3, 5, 6, 7, 8

[28] Pratyush Mishra, Ryan T. Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca A. Popa. Delphi: A cryptographic inference service for neural networks. In *IACR Cryptology ePrint Archive*, 2020. 1

[29] Jong Jin Park. The science behind Astro's graceful, responsive motion, 2023. 2

[30] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2017. 3

[31] Brandon Reagen, Wooseok Choi, Yeongil Ko, Vincent T. Lee, Hsien-Hsin S. Lee, Gu-Yeon Wei, and David M. Brooks. Cheetah: Optimizing and accelerating homomorphic encryption for private inference. *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2020. 1

[32] Brandon Reagen, Woo-Seok Choi, Yeongil Ko, Vincent T Lee, Hsien-Hsin S Lee, Gu-Yeon Wei, and David Brooks. Cheetah: Optimizing and accelerating homomorphic encryption for private inference. In *2021 IEEE International*

Symposium on High-Performance Computer Architecture (HPCA), 2021. 3

[33] Amartya Sanyal, Matt Kusner, Adria Gascon, and Varun Kanade. TAPAS: Tricks to accelerate (encrypted) prediction as a service. In *International conference on machine learning*, 2018. 3

[34] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 2019. 2

[35] Wenting Zheng Srinivasan, PMRL Akshayaram, and Popa Raluca Ada. Delphi: A cryptographic inference service for neural networks. In *Proc. 29th USENIX Secur. Symp*, 2019. 3

[36] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning*, 2020. 3

[37] Manoj-Rohit Vemparala, Nael Fasfous, Alexander Frickenstein, Sreetama Sarkar, Qi Zhao, Sabine Kuhn, Lukas Frickenstein, Anmol Singh, Christian Unger, Naveen-Shankar Nagaraja, Christian Wressnegger, and Walter Stechele. Adversarial robust model compression using in-train pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 66–75, 2021. 2

[38] Haotao Wang, Tianlong Chen, Shupeng Gui, TingKuei Hu, Ji Liu, and Zhangyang Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. *Advances in Neural Information Processing Systems*, 2020. 2, 3

[39] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. AugMax: Adversarial composition of random augmentations for robust training. In *Neural Information Processing Systems*, 2021. 2, 3, 5

[40] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 2

[41] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Loddon Yuille, and Quoc V. Le. Adversarial examples improve image recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4

[42] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (Sfcs 1986)*, 1986. 3

[43] Jiahui Yu, L. Yang, N. Xu, Jianchao Yang, and Thomas S. Huang. Slimmable neural networks. *ArXiv*, 2018. 3

[44] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Young Joon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[45] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[46] Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ArXiv*, 2017. 2

[47] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. *ArXiv*, 2019. 2, 5

[48] Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. Revisiting adversarial robustness distillation: Robust soft labels make student better. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 6, 7