CAS-NN: A Robust Cascade Neural Network Without Compromising Clean Accuracy

Zhuohuang Chen[1], Zhimin He[1(✉)], Yan Zhou[1], Patrick P. K. Chan[2(✉)], Fei Zhang[3], and Haozhen Situ[4]

[1] School of Electronic and Information Engineering, Foshan University, Foshan 528000, China
zhmihe@gmail.com

[2] Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510006, China
patrickchan.scut@gmail.com

[3] College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

[4] College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China

Abstract. Adversarial training has emerged as a prominent approach for training robust classifiers. However, recent researches indicate that adversarial training inevitably results in a decline in a classifier's accuracy on clean (natural) data. Robustness is at odds with clean accuracy due to the inherent tension between the objectives of adversarial robustness and standard generalization. Training a single classifier that combines high adversarial robustness and high clean accuracy appears to be an insurmountable challenge. This paper proposes a straightforward strategy to bridge the gap between robustness and clean accuracy. Inspired by the idea underlying dynamic neural networks, *i.e.,* adaptive inference, we propose a robust cascade framework that integrates a standard classifier and a robust classifier. The cascade neural network dynamically classifies clean and adversarial samples using distinct classifiers based on the confidence score of each input sample. As deep neural networks suffer from serious overconfident problems on adversarial samples, we propose an effective confidence calibration algorithm for the standard classifier, enabling accurate confidence scores for adversarial samples. The experiments demonstrate that the proposed cascade neural network increases the clean accuracies by 10.1%, 14.67%, and 9.11% compared to the advanced adversarial training (HAT) on CIFAR10, CIFAR100, and Tiny ImageNet while keeping similar robust accuracies.

Keywords: Adversarial training · Adversarial learning

1 Introduction

Deep neural networks (DNNs) are notoriously vulnerable to small and imperceptible perturbations in the input data known as *adversarial examples* [4,12,16].

B. Luo et al. (Eds.): ICONIP 2023, LNCS 14448, pp. 494–505, 2024.
https://doi.org/10.1007/978-981-99-8082-6_38

Adversarial training has been demonstrated to be the most effective approach to improving the accuracy on adversarial examples *robust accuracy* [4]. However, adversarial training leads to an undesirable reduction in natural unperturbed inputs (*clean accuracy*) [13,15]. There is an inherent tension between the goals of adversarial robustness and standard generalization due to the fundamental disparities in the features learned through standard training and robust training [13].

Subsequently, several vanilla adversarial training strategies have been proposed to achieve a better trade-off between clean accuracy and robust accuracy [8,9,14,15]. However, the goals of adversarial robustness are fundamentally at odds with standard generalization [13,15], Consequently, it is challenging and even impossible to train a single classifier to attain high adversarial robustness without compromising clean accuracy. Despite these recent advances, closing the gap between clean accuracy and robust accuracy remains an open challenge.

Dynamic neural network (NN) is an emerging research topic due to its high efficiency and adaptability [6]. It adapts model structures to allocate appropriate computation according to different inputs during inference, thereby reducing redundant computation on simple samples. Inspired by the concept of adaptive inference in dynamic NN, we propose a robust cascade neural network (CAS-NN) to alleviate the gap between clean accuracy and adversarial robustness. Rather than employing a single classifier, CAS-NN cascades two classifiers, *i.e.,* a standard classifier for clean samples and a robust classifier for adversarial samples. After obtaining the confidence score of the standard classifier on the input sample, early exiting occurs if the confidence score surpasses a predefined threshold; otherwise, the robust classifier is activated and makes the final decision. CAS-NN employs confidence-based criteria to determine the adopted classifier. The overconfidence problem of deep neural networks on adversarial samples [4,7] is an obstacle of such decision paradigm as adversarial samples obtain high confidences, leading to early exiting. In this paper, we address this problem by designing a confidence calibration on the standard classifier. The main contributions of this paper are as follows.

- we propose a cascade framework to alleviate the gap between clean accuracy and adversarial robustness. It can dynamically handle clean samples and adversarial samples using a standard model and a robust model.
- we propose an effective calibration method that provides low confidence scores for adversarial samples. The calibrated standard neural network outputs predictive scores that accurately reflect the actual likelihood of correctness.
- experiments on CIFAR10, CIFAR100, and Tiny ImageNet demonstrate that CAS-NN achieves comparable robust accuracy to state-of-the-art adversarial training algorithms while causing minimal or no decline in the clean accuracy.

2 Related Work

A variety of works have attempted to alleviate the gap between clean accuracy and adversarial robustness. An effective strategy is dual-domain training whose

loss function integrates both clean and adversarial objectives. Zhang *et al.* [15] characterized the trade-off by decomposing the robust error into the sum of the natural error and the boundary error. They proposed TRADES, which minimized the cross-entropy loss of clean samples and a KL divergence term that encourages smoothness in the neighborhood of samples. Rade *et al.* [8] observed that adversarial training resulted in a large margin along certain adversarial directions, leading to a significant drop in clean accuracy. They extended the objective function of TRADES by incorporating an additional helper loss, calculated using helper examples with proper labels in order to reduce the excessive margin. However, combining the optimization tasks of both clean and adversarial objectives into a single model and enforcing complete matching of the feature distributions between adversarial and clean examples may result in sub-optimal solutions.

More recent attempts to enhance the trade-off between clean accuracy and adversarial robustness have primarily focused on refining adversarial training by early-stopping [9], additional data [10] and weight initialization for training with large perturbations [11]. Despite these advancements, the problem remains far from being solved, with a substantial gap between clean accuracy and robustness observed in practical image datasets.

3 Preliminaries

Given a classification task with c classes, $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ represent a deep neural network parameterized by $\boldsymbol{\theta}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^c$ are the input and output space. The predicted label of an input sample $\boldsymbol{x}$ is given by $\arg\max_k f_{\boldsymbol{\theta}}(\boldsymbol{x})_k$, where $f_{\boldsymbol{\theta}}(\boldsymbol{x})_k$ is the k-th component of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. The confidence score of an input sample $\boldsymbol{x}$ is defined as the maximum softmax prediction probability, *i.e.,* $s(\boldsymbol{x}) = \max_k f_{\boldsymbol{\theta}}(\boldsymbol{x})_k$. Adversarial training is formulated as

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}} \Big(\max_{\boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x})} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}'), y), \tag{1}$$

where $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is the training set. $\mathcal{B}_\epsilon(\boldsymbol{x})$ denotes the l_p-norm ball centered at $\boldsymbol{x}$ with radius ϵ, which has been extensively employed in recent studies [7,14,15]. The inner maximization problem aims to generate an adversarial sample $\boldsymbol{x}'$ with respect to the training sample $\boldsymbol{x}$. Existing adversarial training algorithms approximately optimize the inner maximization problem using a gradient-based iterative solver, *e.g.,* multi-step PGD [7] and CW attacks [2].

Following the terminologies in Refs. [8,15], the *clean* (or *natural*) and *robust* (or *adversarial*) accuracies refer to the test accuracies on clean samples and adversarial samples, respectively.

4 Robust Cascade Neural Network

We propose a robust cascade neural network (CAS-NN) that combines two neural networks (NNs) to achieve both high standard accuracy and robust accuracy.

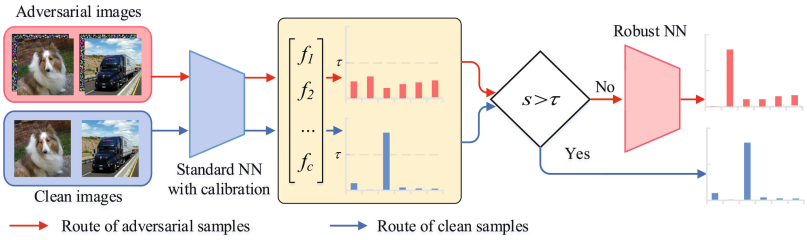

Fig. 1. Schematic of the robust cascade system.

Specifically, the standard NN is used to classify clean samples, while the robust NN is employed to classify adversarial samples. The route is determined by the confidence score of the input sample, which is measured by the standard NN. The confidence score is defined as the maximum softmax prediction probability of the standard NN. The diagram of the robust cascade system is illustrated in Fig. 1. Given a sample $\boldsymbol{x}$, the standard NN first makes a prediction and outputs the confidence score $s(\boldsymbol{x}) = \max_k f_{\boldsymbol{\theta}}(\boldsymbol{x})_k$. If the confidence score surpasses the threshold τ, the input sample is likely to be untainted. Then the inference terminates and the system outputs the prediction. If $s \leq \tau$, the standard NN is not confident of the prediction and the input sample is likely to be contaminated, prompting the activation of the robust NN to deliver the final prediction.

4.1 Confidence Calibration

DNNs suffer from a serious overconfident problem on adversarial samples [4, 7]. CAS-NN cannot solely depend on the confidence score of the standard NN to determine the activation of the robust NN. It is necessary to calibrate the confidence score of the standard NN on adversarial samples.

The confidence score should be indicative of the actual likelihood of correctness. Since the standard NN's predictions on adversarial samples are incorrect, it should output low confidence scores on these samples. Based on the low confidence scores, adversarial samples will be passed to the robust NN for the final decision. Since the confidence score is defined as the maximum softmax prediction probability, a uniform distribution of outputs results in the lowest confidence score. Therefore, we employ the Kullback-Leibler (KL) divergence to encourage the softmax prediction probability of the standard NN on adversarial samples to approximate a uniform distribution. This approach ensures that the standard NN produces the lowest confidence score for adversarial samples. Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with n samples, the loss function used to calibrate the confidence score of standard NN is defined as

$$\mathcal{L}_c = \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}}(\mathrm{KL}(P_u || f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(s)}) + \mathrm{KL}(P_u || f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(r)})), \tag{2}$$

where KL denotes the KL divergence, which quantifies the difference between two distributions. $\boldsymbol{x}^{(s)}$ and $\boldsymbol{x}^{(r)}$ refer to the adversarial samples generated by an adversarial attack against the standard NN and the robust NN, respectively.

In this paper, we use the PGD attack [7]. P_u represents a discrete uniform distribution, where $P_u(X = k) = 1/c$, $k \in \{1, 2, ..., c\}$.

4.2 Training of CAS-NN

CAS-NN comprises two neural networks: a standard NN and a robust NN. The robust NN can be trained using any adversarial training algorithm, making our extension compatible with state-of-the-art (SOTA) adversarial training techniques. During the training of a standard NN, confidence regularization should be taken into account, resulting in the loss defined as

$$\mathcal{L} = \mathcal{L}_{std} + \lambda \mathcal{L}_c, \tag{3}$$

where $\mathcal{L}_{std}$ is the original loss term of a standard NN. In this paper, we use the softmax cross entropy, which ensures a high confidence score for the clean sample. λ serves as a parameter that controls the trade-off between the original loss and the confidence regularization term. Although the training of the standard NN involves adversarial samples, we call it a standard NN because it does not incorporate the labels of adversarial samples during training.

Algorithm 1 outlines the pseudo-code for the training process of CAS-NN. We first train a robust NN $f_{\boldsymbol{\theta}_r}$ on the training dataset with an adversarial training algorithm. It is important to emphasize that various adversarial training algorithms can be employed to train the robust NN. In this paper, we use a standard adversarial training [7]. The calibrated standard NN $f_{\boldsymbol{\theta}_s}$ is trained by stochastic gradient descent. In each epoch, a mini-batch of samples is randomly selected, and their corresponding adversarial samples are generated through PGD attack against $f_{\boldsymbol{\theta}_s}$ and $f_{\boldsymbol{\theta}_r}$ as shown in Lines 8 and 9. The parameters of the calibrated standard NN $f_{\boldsymbol{\theta}}$ are optimized based on Eq. (3) (Line 10). $\mathcal{B}_\epsilon(\boldsymbol{x})$ is the closed ball with radius ϵ centered at $\boldsymbol{x}$, where $\mathcal{B}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' | \parallel \boldsymbol{x} - \boldsymbol{x}' \parallel_\infty \leq \epsilon\}$. $\prod_{\mathcal{B}_\epsilon(\boldsymbol{x})}$ denotes the projection function that maps the adversarial variant back to the ϵ-ball centered at $\boldsymbol{x}$.

5 Adaptive Attack Against CAS-NN

Since CAS-NN is composed of two neural networks, the intuitive attack is that we generate adversarial samples targeted on the standard NN or the robust NN. However, neither of these strategies exploits the structure and mechanism of CAS-NN. In this paper, we propose an adaptive attack against CAS-NN in order to better evaluate the robustness of CAS-NN. The detailed process is shown in Algorithm 2. The adaptive attack is a gradient-based method that utilizes the model's gradient to search for the perturbation direction. In each iteration, the gradient information is calculated with respect to different models based on the confidence score of the adversarial sample $s(\boldsymbol{x}')$. Specifically, if the confidence score exceeds the threshold τ, the gradient is calculated based on the robust NN $f_{\boldsymbol{\theta}_r}$; otherwise, the gradient is obtained based on the standard NN $f_{\boldsymbol{\theta}_s}$. $Attack(f_{\boldsymbol{\theta}}, \boldsymbol{x}')$ represents one-step update of the adversarial sample $\boldsymbol{x}'$

Algorithm 1. Training of a robust cascade neural network.

Require: $\mathcal{D}=\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$: training data; m: batch size; η: learning rate; ϵ: attack radius; α: attack step size; λ: the trade-off parameter between the original loss and the confidence regularization term. K: number of attack iterations.

Ensure: the cascade NN with a calibrated standard NN ($f_{\boldsymbol{\theta}_s}$) and a robust NN ($f_{\boldsymbol{\theta}_r}$)

train a robust NN $f_{\boldsymbol{\theta}_r}$ via an adversarial training algorithm
randomly initial the parameters $\boldsymbol{\theta}_s$ of the standard NN $f_{\boldsymbol{\theta}_s}$
repeat
  Sample a mini-batch of samples $\mathcal{D}_b = \{\boldsymbol{x}_j, y_j\}_{j=1}^m$ from $\mathcal{D}$
  for $j = 1, ..., m$ **do**
    $\boldsymbol{x}_j^{(s)}$ and $\boldsymbol{x}_j^{(r)}$ are initialized by a uniformly random point in the l_p-norm ball centered at $\boldsymbol{x}$ with radius ϵ
    for $k = 1, ..., K$ **do**
      $\boldsymbol{x}_j^{(s)} = \prod_{\mathcal{B}_\epsilon}(\boldsymbol{x}_j^{(s)} + \alpha \text{sign}(\nabla_{\boldsymbol{x}_j^{(s)}} \ell(f_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{(s)}), y_j)))$
      $\boldsymbol{x}_j^{(r)} = \prod_{\mathcal{B}_\epsilon}(\boldsymbol{x}_j^{(r)} + \alpha \text{sign}(\nabla_{\boldsymbol{x}_j^{(r)}} \ell(f_{\boldsymbol{\theta}_r}(\boldsymbol{x}_j^{(r)}), y_j)))$
  $\boldsymbol{\theta}_s = \boldsymbol{\theta}_s - \eta \nabla_{\boldsymbol{\theta}_s} \mathbb{E}_{(\boldsymbol{x}_j, y_j)\sim \mathcal{D}_b}(\text{CE}(f_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j), y_j) + \lambda \text{KL}(P_u || f_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{(s)})) + \lambda \text{KL}(P_u || f_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{(r)})))$
until training completed

using the gradient calculated by the model $f_{\boldsymbol{\theta}}$. Any gradient-based attack such as PGD, CW, and MM can be utilized to update the adversarial sample.

Algorithm 2. Adaptive attack against the robust cascade neural network.

Require: $(\boldsymbol{x}, y)$: target sample; $f_{\boldsymbol{\theta}_s}$: the calibrated standard NN; $f_{\boldsymbol{\theta}_r}$: the robust NN; τ: the threshold of confidence score; K: number of attack iterations.

Ensure: adversarial sample $\boldsymbol{x}'$

$\boldsymbol{x}'$ is initialized by a uniformly random point in the l_p-norm ball centered at $\boldsymbol{x}$
for $i = 0, ..., K-1$ **do**
  $s(\boldsymbol{x}') = \max_k f_{\boldsymbol{\theta}_s}(\boldsymbol{x}')_k$,
  if $s(\boldsymbol{x}') > \tau$ **then** $\boldsymbol{x}' \leftarrow Attack(f_{\boldsymbol{\theta}_s}, \boldsymbol{x}')$
  else $\boldsymbol{x}' \leftarrow Attack(f_{\boldsymbol{\theta}_r}, \boldsymbol{x}')$
return $\boldsymbol{x}'$

6 Experiments

Datasets. We extensively assess the proposed method on three datasets, *i.e.,* CIFAR10, CIFAR100, and Tiny ImageNet. Both CIFAR-10 and CIFAR-100 contain 50,000 training images and 10,000 test images that are in the size of 32×32. CIFAR-10 covers 10 classes while CIFAR-100 consists of 100 classes. Tiny ImageNet is a more complex dataset, which is a subset of ImageNet. It comprises 200 classes with each class containing 500 training images and 50 validation images that are in the size of 64×64.

Evaluation of Adversarial Robustness. Robust accuracy is evaluated using three adversarial attacks: PGD [7], CW [2], and Minimum-Margin(MM) attacks [3]. PGD is a reliable and computationally efficient method for assessing the robustness of a model. CW exhibits strong attack performance by introducing an alternative loss. Both PGD and CW employ 20 steps. The MM attack searches for the optimal adversarial sample with the minimum margin. It achieves comparable performance to AutoAttack while requiring only 3% of the computational time. The target selection numbers of the MM attack are set to 3 and 9. As CAS-NN comprises two neural networks, we employ three strategies to attack CAS-NN: adversarial samples generated according to the standard (Std) NN, the robust (Rob) NN, and the adaptive (Ada) path in Sect. 5. The attack radius ε is $8/255$ for all attacks.

Implementation Details. The calibrated standard NN is trained according to Algorithm 1. The robust NN in CAS-NN is trained with adversarial training [7], where the adversarial samples are generated by PGD20 with l_∞ norm. Both the standard NN and the robust NN share the same architecture and are optimized by an SGD optimizer with 0.9 momentum. The learning rate and weight decay are 0.02 and 5×10^{-4}, respectively. We use NF-ResNet18 [1] as the backbone of the neural network. The thresholds τ of CAS-NN are set to 0.2, 0.1, and 0.1 for the CIFAR10, CIFAR100, and Tiny ImageNet, respectively. The trade-off parameter λ between the original loss and the confidence regularization term is set to 1. The batch size m, attack step size α, and the number of attack iterations K are 256, $2/255$ and 10, respectively. Each experiment is independently run five times, and the average result is reported.

6.1 Comparisons with Prominent Adversarial Training Algorithms

Table 1 reports the clean accuracy and robust accuracy of CAS-NN and some prominent adversarial training algorithms. A high clean accuracy and robust accuracy indicate the model can well classify clean samples and adversarial samples, respectively. It can be observed that current adversarial training algorithms exhibit a significant decrease in clean accuracy compared to the standard training method (ST). Specifically, AT, TRADES, and HAT exhibit an average degradation of 14.21%, 13.44%, and 12.54% across three datasets, respectively. However, CAS-NN achieves comparable clean accuracy to ST on CIFAR10 and CIFAR100, with only a decrease of 3.23% on Tiny ImageNet.

We present the robust accuracies of CAS-NN against adversarial attacks based on the standard (Std) NN, robust (Rob) NN, and adaptive (Ada) path. CAS-NN achieves high robust accuracies on adversarial samples generated based on the standard NN. The result is intuitive as the adversarial samples are classified by the robust NN which can well classify adversarial samples generated based on the standard NN. In the case of PGD and CW attacks, CAS-NN has similar robust accuracies on adversarial samples generated based on the robust NN and the adaptive path. However, CAS-NN exhibits 11.71 to 19.96 % lower robust accuracies against the adaptive MM attack compared to the one targeted

Table 1. Clean accuracy (%) and robust accuracy (%) of different robust classifiers on CIFAR10, CIFAR100 and Tiny ImageNet. Robust accuracy is measured by different attacks, including PGD20 [7], CW20 [2], MM3 [3] and MM9 [3]. ST denotes standard training. ↑ suggests that higher values correspond to better performance.

Datasets	Classifiers	Clean acc(↑)	Robust acc (↑)											
			PGD20			CW20			MM3			MM9		
CIFAR10	ST	92.47	0.02			0.01			0.00			0.00		
	AT [7]	81.58	46.73			45.78			41.82			40.85		
	TRADES [15]	80.90	48.10			44.61			42.43			42.33		
	HAT [8]	82.61	46.38			43.22			41.19			41.13		
	CAS-NN	92.71	Std	Rob	Ada	Std	Rob	Ada	Std	Rob	Ada	Std	Rob	Ada
			81.50	46.73	46.73	81.49	45.80	45.78	78.59	55.38	43.67	78.43	54.47	40.54
CIFAR100	ST	73.33	7.44			0.00			0.00			0.00		
	AT [7]	56.30	24.86			23.79			21.32			20.58		
	TRADES [15]	56.83	25.14			21.67			19.29			18.84		
	HAT [8]	57.91	24.57			21.14			19.42			19.38		
	CAS-NN	72.58	Std	Rob	Ada	Std	Rob	Ada	Std	Rob	Ada	Std	Rob	Ada
			56.12	24.86	24.86	56.06	23.81	23.80	50.23	42.52	24.73	49.90	41.83	21.87
Tiny ImageNet	ST	60.56	8.04			0.00			0.00			0.00		
	AT [7]	45.84	18.95			16.72			14.38			13.99		
	TRADES [15]	48.30	18.73			14.42			12.33			11.91		
	HAT [8]	48.22	17.01			13.25			11.54			11.27		
	CAS-NN	57.33	Std	Rob	Ada	Std	Rob	Ada	Std	Rob	Ada	Std	Rob	Ada
			45.75	18.95	18.86	45.40	16.72	16.72	39.10	31.70	16.94	38.75	31.32	15.03

on the robust NN. The adversarial attack with an adaptive path proves to be the most effective among all the datasets. In the subsequent experiments, we only consider the adaptive attack as it is the most effective attack against CAS-NN. CAS-NN exhibits comparable or even higher robust accuracies than AT, TRADES, and HAT across four adversarial attacks (PGD20, CW20, MM3, and MM9). This indicates that CAS-NN can notably increase clean accuracy without compromising robust accuracy.

6.2 Performance of Confidence Calibration

A well-calibrated classifier ensures that the average confidence of a model aligns closely with its actual accuracy. The effectiveness of confidence calibration is evaluated using the Expected Calibration Error (ECE) [5], which measures how closely the average confidence of a model aligns with its actual accuracy. A better-calibrated classifier exhibits a lower ECE value. Table 2 shows the ECE values of the standard NN and the calibrated NN on adversarial samples. The standard NN has very high ECE values (greater than 89%) on adversarial samples. However, after confidence calibration, the average values of ECE decrease to 2.26%, 0.62%, and 1.92% in CIFAR10, CIFAR100, and Tiny ImageNet, respectively. The calibrated NNs achieve ECE values lower than 3.22% across all datasets.

After confidence calibration, the standard NN outputs low confidence scores for adversarial samples, accurately reflecting their actual accuracy. The majority of adversarial samples have confidence scores below the threshold, triggering the

Table 2. Expected Calibration Error (ECE) of the proposed confidence calibration on adversarial samples generated by PGD, CW, MM3, and MM9

Attack	CIFAR10		CIFAR100		Tiny ImageNet	
	Standard NN	Calibrated NN	Standard NN	Calibrated NN	Standard NN	Calibrated NN
PGD	98.15	2.71	89.33	0.75	91.85	1.50
CW	97.22	3.22	97.80	0.75	92.82	1.76
MM3	98.07	1.47	99.04	0.48	92.82	2.41
MM9	98.49	1.63	99.05	0.49	92.20	2.01

Table 3. Percentage of samples (%) that are classified by the robust NN.

Datasets	Clean	PGD	CW	MM3	MM9
CIFAR10	0.00	100.00	99.98	99.93	99.88
CIFAR100	0.00	100.00	99.93	99.83	99.74
Tiny ImageNet	0.87	100.00	99.90	99.90	99.88

activation of the robust classifier to make a final decision for these samples. Table 3 shows the percentage of samples whose confident scores given by the calibrated standard NN fall below the threshold. The majority of adversarial samples generated by PGD, CW, MM3, and MM9 fall below the threshold and are consequently transferred to the robust NN. Almost all clean samples have confidence scores exceeding the threshold, resulting in their classification by the standard NN.

6.3 Margin Along the Initial Adversarial Direction

Previous research [8] demonstrates that adversarial training leads to an excessive increment in the margin along initial adversarial directions to attain robustness, which prevents the neural network from using high discriminative features in those directions, leading to a substantial drop in clean accuracy. As described in Ref. [8], there exists a connection between the augmentation of the margin along initial adversarial directions and the reduction in clean accuracy. In contrast to adversarial training, the calibrated standard NN in CAS-NN does not require precise classification of adversarial samples, and its objective function does not involve the one-hot label vectors of the adversarial sample. Consequently, the confidence calibration has minimal impact on the margins between the training samples and the decision boundary.

Figure 2 shows the margin between clean samples and the decision boundary along initial adversarial directions. The horizontal axes denote the margins between clean samples and the decision boundary of the standard NN while the vertical axes are the margins between these samples and the decision boundaries of the adversarially trained NN (the upper row) and the calibrated standard NN (the bottom row). The standard NN typically exhibits margins ranging from 0 to

2, with an average margin of 1.36. However, after adversarial training, the margins have a substantial increment, resulting in an average margin of 11.56, which is 8.5 times greater than that of the standard NN. For the calibrated standard NN, there is either no or only a slight increment in the margin across different datasets. The average margin across the three datasets is 1.55, suggesting that confidence calibration minimally alters the geometry of the decision boundary, thereby exerting little effect on clean accuracy.

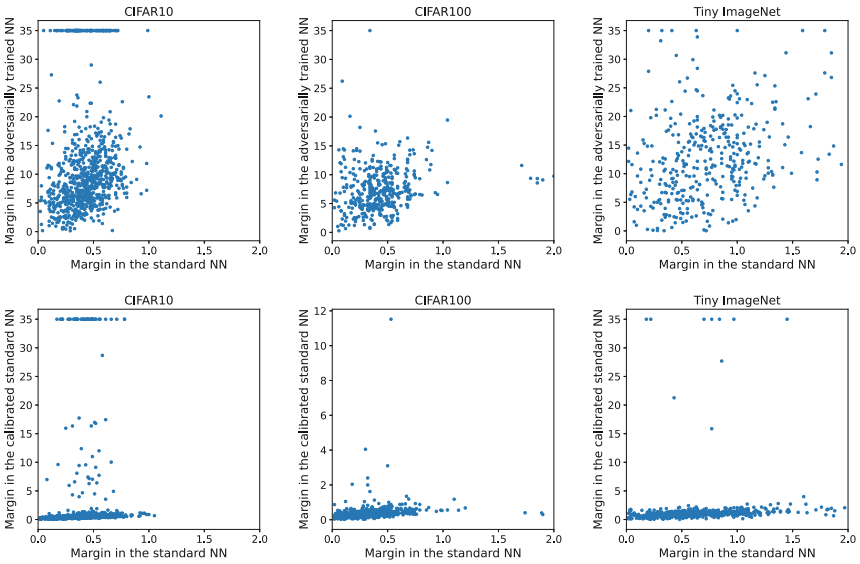

Fig. 2. Margins between clean samples and the decision boundaries of standard NN, adversarially trained NN and calibrated standard NN along initial adversarial directions. Each dot denotes a clean sample.

6.4 Analysis of the Hyper-parameter

Figure 3 illustrates the accuracies of CAS-NN on clean samples and adversarial samples as well as the pass rates for different values of the hyper-parameter τ. We are only interested in the pass rate of adversarial samples that are misclassified by the standard NN. With an increasing threshold, a greater number of samples exhibit confidence scores below the threshold, resulting in an increased pass rate. When the threshold is 1.0, all samples are classified by the robust NN, resulting in CAS-NN functioning as the traditional adversarial training algorithm. We can observe a significant degradation of clean accuracy in this setting. If the threshold is set to 0, all samples are classified by the standard NN, rendering CAS-NN comparable to the classifier with standard training. In this setting, CAS-NN has the highest clean accuracy but the lowest robust accuracy.

For CIFAR10, the confidence scores of the majority of clean samples surpass 0.6, whereas the confidence scores of all adversarial samples fall below 0.2. Despite a small fraction of clean samples being classified by the robust NN with thresholds of 0.6, 0.7, 0.8, or 0.9, the clean accuracy has no degradation as the robust NN accurately classifies these samples. CAS-NN attains identical clean accuracy and robust accuracy when the threshold is within the range of [0.2, 0.9]. However in the more challenging tasks such as CIFAR100 and Tiny ImageNet, a higher threshold substantially amplifies the pass rate of clean samples, consequently resulting in a more pronounced decline in clean accuracy. When the thresholds are 0.2, 0.1, and 0.1 for CIFAR10, CIFAR100, and Tiny ImageNet, respectively, the pass rates of adversarial samples reach 1.0, suggesting that the confidence scores of adversarial samples are significantly reduced through confidence calibration. Based on the aforementioned observations, the pass rate can guide the setting of the threshold. Under the premise of ensuring a high pass rate of adversarial samples, the threshold should be set as small as possible to achieve a high clean accuracy.

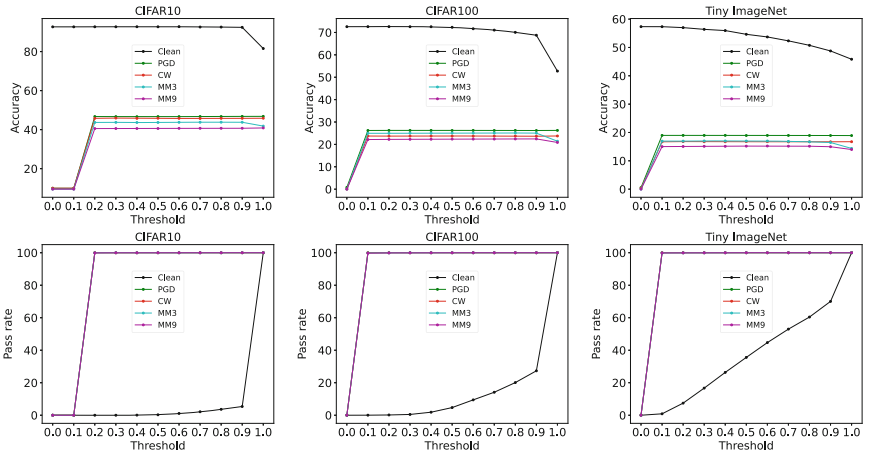

Fig. 3. Accuracies (%) of CAS-NN on clean samples and adversarial samples generated by PGD, CW, MM3, and MM9 as well as the pass rates (%) for different thresholds τ.

7 Conclusion

Considering the inherent contradiction between the adversarial robustness and standard generalization, we propose a novel strategy called CAS-NN, to bridge the gap between robustness and standard accuracy. CAS-NN dynamically classifies clean samples and adversarial samples with different neural networks according to the confidence scores of these samples. In order to mitigate the overconfidence problem in adversarial samples, we designed a confidence calibration

algorithm. Experimental results on CIFAR10, CIFAR100, and Tiny ImageNet show that CAS-NN achieves comparable or slightly lower clean accuracy compared to the standard neural network while attaining similar robust accuracy as existing adversarial training algorithms. In future work, we will develop more strategies based on adaptive inference to construct a safety-critical system.

Acknowledgements. This work is supported by Guangdong Basic and Applied Basic Research Foundation (Nos. 2022A1515140116, 2022A1515010101), National Natural Science Foundation of China (Nos. 61972091, 61802061).

References

1. Brock, A., De, S., Smith, S.L.: Characterizing signal propagation to close the performance gap in unnormalized resnets. In: ICLR (2021)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: S&P, pp. 39–57 (2017)
3. Gao, R., et al.: Fast and reliable evaluation of adversarial robustness with minimum-margin attack. In: ICML, pp. 7144–7163 (2022)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
5. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML, pp. 1321–1330 (2017)
6. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., Wang, Y.: Dynamic neural networks: a survey. TPAMI **44**(11), 7436–7456 (2021)
7. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
8. Rade, R., Moosavi-Dezfooli, S.M.: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In: ICLR (2022)
9. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: ICML, pp. 8093–8104 (2020)
10. Sehwag, V., et al.: Robust learning meets generative models: can proxy distributions improve adversarial robustness? In: ICLR (2022)
11. Shaeiri, A., Nobahari, R., Rohban, M.H.: Towards deep learning models resistant to large perturbations. arXiv:2003.13370 (2020)
12. Szegedy, C., et al.: Intriguing properties of neural networks. In: ICLR (2014)
13. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: ICLR (2018)
14. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: ICLR (2020)
15. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: ICML, pp. 7472–7482 (2019)
16. Zheng, J., Chan, P.P., Chi, H., He, Z.: A concealed poisoning attack to reduce deep neural networks' robustness against adversarial samples. Inf. Sci. **615**, 758–773 (2022)