Adversarial Feature Stacking for Accurate and Robust Predictions

Faqiang Liu, Rong Zhao, and Luping Shi†

***Abstract*—Deep Neural Networks (DNNs) have achieved remarkable performance on a variety of applications but are extremely vulnerable to adversarial perturbation. To address this issue, various defense methods have been proposed to enhance model robustness. Unfortunately, the most representative and promising methods, such as adversarial training and its variants, usually degrade model accuracy on benign samples, limiting practical utility. This indicates that it is difficult to extract both robust and accurate features using a single network under certain conditions, such as limited training data, resulting in a trade-off between accuracy and robustness. To tackle this problem, we propose an Adversarial Feature Stacking (AFS) model that can jointly take advantage of features with varied levels of robustness and accuracy, thus significantly alleviating the aforementioned trade-off. Specifically, we adopt multiple networks adversarially trained with different perturbation budgets to extract either more robust features or more accurate features. These features are then fused by a learnable merger to give final predictions. We evaluate the AFS model on CIFAR-10 and CIFAR-100 datasets with strong adaptive attack methods, which significantly advances the state-of-the-art in terms of the trade-off. Without extra training data, the AFS model achieves a benign accuracy improvement of ~6% on CIFAR-10 and ~9% on CIFAR-100 with comparable or even stronger robustness than the state-of-the-art adversarial training methods. This work demonstrates the feasibility to obtain both accurate and robust models under the circumstances of limited training data.**

***Index Terms*—adversarial robustness, stacking, trade-off**

I. Introduction

WITH the assistance of big data and powerful parallel computing platforms, DNNs [1] characterized by end-to-end training have achieved significant success in computer vision [2], [3], [4] and natural language processing [5], [6], [7]. The performance of DNNs on some domain-specific tasks is comparable or even superior to that of humans. However, DNNs are extremely vulnerable to adversarial perturbation, which is imperceptible by humans but can fool the state-of-the-art deep models to give wrong predictions [8], [9], [10]. The search of adversarial perturbation can be formulated as an optimization problem with a certain constraint of perturbation, which can be easily solved by gradient-based methods [11], [12] or gradient-free methods [13], [14]. The poor robustness of DNNs in the adversarial setting not only challenges the fundamental mechanism of deep learning, but also hinders applications of DNNs in security-critical scenarios. Therefore, it is crucial to develop both accurate and robust networks under the condition of limited training data.

F. Liu, R. Zhao and L. Shi are with the Department of Precision Instrument, Center for Brain Inspired Computing Research, Tsinghua University, Beijing, 100084, China

† Corresponding to: lpshi@mail.tsinghua.edu.cn

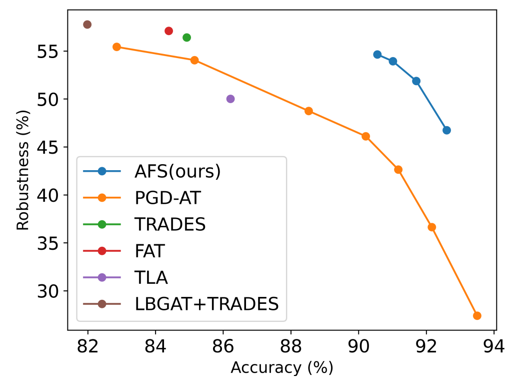


Fig. 1. Comparison of PGD-AT [15], TRADES [16], FAT [17], TLA [18], LBGAT+TRADES [19], and AFS. Robustness is evaluated on CIFAR-10 with PGD attack. Our model significantly improves the trade-off between accuracy and robustness.

Till now, a large body of work has been proposed to enhance model robustness. Some work tries to eliminate the adversarial perturbation by a denoising module before samples fed into networks [20], [21] or to suppress the effect of adversarial perturbation on hidden activations of networks [22], [23]. Another line of work takes some operations to mask the gradient or obfuscate the gradient of networks, which can hamper gradient-based attacks [24], [25], [26]. However, most of these methods can be defeated by adaptive attacks that take the defense methods into the consideration when generating adversarial perturbation, thus giving a false sense of robustness [27], [28]. On the contrary, certified defense methods are provably robust to certain norm-bounded adversarial perturbation with theoretical guarantees, but the computational cost of these methods is usually too high to be extended to large models and large data sets [29], [30], [31].

From a comprehensive consideration of feasibility and effectiveness, adversarial training based on projected gradient descent (PGD-AT) [12] remains one of the most promising and popular methods to improve model robustness. PGD-AT reformulates the training objective of DNNs as a robust optimization problem, which forces the network to be robust to adversarial perturbation during training. Practically, this method is implemented by injecting online-generated adversarial perturbation into benign samples at the training stage. Unfortunately, PGD-AT and its variants [32], [17], [33] substantially degrade model accuracy on benign samples, which limits their practical value for specific tasks. This issue raises the concern of whether there is an inherent trade-off between accuracy and robustness [34], [16], [35]. In essence, the final answer to the question depends on the property of considered data distribution and the capability of the used models, which is also influenced by the training algorithm to optimize the model. Remarkably, augmenting the training set with extra data can mitigate the trade-off to some extent, and achieves better accuracy and stronger robustness [36], [37], [38]. These results indicate that we can obtain both accurate and robust models. However, additional training data is not always available due to the heavy cost of data collecting and labeling. This raises a question worth exploring: can we obtain both accurate and robust models at the same time *without extra training data?*

To answer this question, we start by investigating how the features extracted by the networks with different training algorithms influence the model accuracy and robustness. For convenience, we refer to the accurate features as the features that are highly correlated to the labels, and refer to the robust features as the features that keep almost unchanged with adversarially perturbed inputs. Interestingly, a standardly trained network relies on some accurate but non-robust features to give accurate predictions, but an adversarially trained network relies on more robust features [39]. However, these robust features are usually insufficient to support high accuracy. Adversarial training with large perturbation budgets can hinder the network from learning non-robust but predictive features [34], which are important for accurate predictions of benign samples. In contrast, networks adversarially trained with small perturbation budgets have higher accuracy, but the features extracted by these networks are not robust enough, thus leading to lower robustness. From this perspective, the trade-off between accuracy and robustness can be interpreted as it is difficult to train a single network to extract both robust features and non-robust but accurate features.

To solve this dilemma, we develop an Adversarial Feature Stacking (AFS) model that combines the features extracted by multiple networks that are independently adversarially trained with varied perturbation budgets. These features are either more robust but less accurate or more accurate but less robust. Then, we adopt a linear merger to fuse the useful features to give final predictions. Due to the availability of both accurate features and robust features, the AFS model is facilitated to provide both accurate and robust predictions. We evaluate the AFS model on CIFAR-10 and CIFAR-100 datasets with advanced adaptive attack methods. The experimental results indicate that it is feasible to obtain a model with both high accuracy and strong robustness under the circumstances of limited training data. As presented in Figure 1, the proposed AFS model significantly improves the trade-off between accuracy and robustness. The work advances the development of practical deep models with both high accuracy and strong robustness. Our key contributions are summarized as follows:

1 We develop a stacking model to fuse the features extracted by multiple networks with different levels of robustness and accuracy. We propose a training method for the linear merger to balance the feature selection.
2 We verify the AFS model on CIFAR-10 and CIFAR-100 datasets with advanced attack methods. The experimental results demonstrate the feasibility to achieve both high accuracy and strong robustness without extra training data.
3 We conduct extensive ablation experiments and analyze the characteristics of the linear merger, which verify the effectiveness and reveals the mechanism of the AFS model.

II. Methods

In this section, we will introduce the overall architecture and the training methods of the proposed AFS model. The AFS model fuses the features extracted by several adversarially pre-trained networks with a linear merger to give the final prediction. The training procedure of the AFS model can be divided into two stages: the training of the feature extractors and the linear merger. In this work, we focus on the norm bounded adversarial perturbation for classification tasks. The AFS model can be generalized to other tasks and the adversarial perturbation with other constraints.

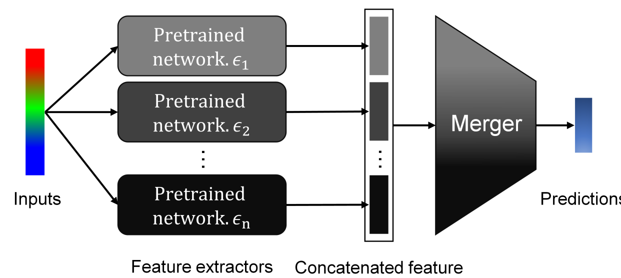


Fig. 2. The schematic of the AFS model. The AFS model fuses the features extracted by multiple networks with a learnable linear merger. These networks are adversarially trained with varied perturbation budgets. The learnable linear classifier is optimized to select useful features for both accurate and robust predictions.

A. Overall architecture

The AFS model consists of two parts: several pre-trained feature extractors with varied levels of robustness and accuracy and a linear merger to fuse the features, whose overall architecture is illustrated in Figure 2. Generally, as discussed above, there are two types of feature extractors. For a given benign input, some of these pre-trained extractors can extract accurate features, thus contributing more accurate predictions. However, these accurate features are not generally robust enough, which can change greatly with a slightly perturbed input. The others of these pre-trained extractors can extract more robust features that are not easily affected by the adversarial perturbation, but the correlation between these features and the true labels are not strong enough. The robustness and accuracy of pre-trained feature extractors can be determined by the strength of the defense methods. For example, the networks trained with different levels of perturbation budgets have different levels of robustness and accuracy. Ideally, if a feature extractor can extract both accurate and robust features, we will get a model with high accuracy and strong robustness. However, it is difficult to achieve under the condition of limited training data empirically. To address this dilemma, as the first step, we propose to use a trainable merger to fuse the features that are extracted by different pre-trained networks with varied levels of accuracy and robustness. The processing procedure of the AFS model can be formulated as follows:

$$z = w^T G(x;\Theta) + b, \tag{1}$$

where x and z denote the input and the prediction, respectively. w and b are the trainable parameters of the linear merger. $G(x;\Theta)$ is formed by concatenating the features generated by the pre-trained extractors along the feature dimension. Θ denotes the set of parameters of all the extractors, which is presented as the following equation:

$$G(x;\Theta) = \left[g_1(x;\theta_1)^T, g_2(x;\theta_2)^T, ..., g_n(x;\theta_n)^T\right]^T, \tag{2}$$

where g_i is the i-th feature extractor parameterized by θ_i, whose outputs are high dimensional vectors. These feature vectors are concatenated together to feed into the linear merger.

B. Adversarially trained feature extractor

In this work, we use the networks that are adversarially trained with different perturbation budgets as the feature extractors. The perturbation budgets should range from small values to large values. The networks adversarially trained with smaller perturbation budgets have higher accuracy but lower robustness. In contrast, the networks adversarially trained with larger perturbation budgets have stronger robustness but lower accuracy. We can infer that these networks trained with different perturbation levels can extract either more accurate features or more robust features with a large diversity. Therefore, we select these networks as the basic member of the proposed stacking model. The models enhanced by other defense methods with different levels of robustness and accuracy can also be considered as the candidates of the feature extractors. The feature extractors used in this work are separately trained with PGD-AT method [12]. The training objective of each extractor is formulated as follows:

$$\min_{\theta_i, w_i, b_i} \mathbb{E}_{(x,y)\in\mathcal{D}} \max_{\|\hat{x}-x\|_p \le \epsilon_i} \mathcal{L}\left[w_i^T g_i(\hat{x};\theta_i) + b_i, y\right], \tag{3}$$

where ϵ_i is the perturbation budget for the i-th feature extractor and w_i, b_i are the parameters of the corresponding linear classifier. $\mathcal{D}$ is the data distribution and y is the true label. Each feature extractor paired with its classifier is trained on the whole dataset. The extractors can generate diverse features due to the independent training with varied perturbation budgets. The parameters of the feature extractor remain unchanged after the adversarial training. The inner maximization is solved by PGD method, whose iterative formula is presented as follows:

$$\begin{aligned} \hat{x}_0 &= x + \text{Uniform}(-\epsilon_i, \epsilon_i) \\ \hat{x}_{k+1} &= P_S(\hat{x}_k + \eta \nabla_x \mathcal{L}\left[w_i^T g_i(\hat{x}_k;\theta_i) + b_i, y\right]), \end{aligned} \tag{4}$$

where η is the step size and P_S is a projection operator that makes $\hat{x}_k$ within the norm-bounded ball.

C. Linear merger

A naive approach to fuse the features is to take the average of the outputs of the original corresponding classifiers. However, this approach cannot adaptively select features. To address this issue, we propose a learnable linear merger to fuse the diverse features to give final predictions, which can balance the total accuracy and robustness when considering all the features. After obtaining a group of feature extractors, the linear merger is trained on the generated features. To balance the final accuracy and robustness of the whole stacking model, the features of both benign samples and adversarially perturbed samples are used to train the merger. The training objective of the linear merger is presented as follows:

$$\begin{aligned} &\min_{w,b} \mathbb{E}_{(x,y)\in\mathcal{D}} \{\alpha \mathcal{L}\left[w^T G(x;\Theta) + b, y\right] \\ &+ \max_{\|\hat{x}-x\|_p \le \epsilon} (1-\alpha)\mathcal{L}\left[w^T G(\hat{x};\Theta) + b, y\right]\}, \end{aligned} \tag{5}$$

where α is the ratio ranging in $[0,1]$. α can balance the loss for accuracy and robustness. ϵ denotes the perturbation budget for training the merger. The crafting method of the adversarial perturbation for adversarial training is the same as the Equation 4 except for targeting at the whole model. A small α encourages the merger to use more robust features to give robust predictions. On the contrary, a large α encourages the merger to use more accurate features to give accurate predictions. Remarkably, even with a too small or a too large α, the stacking model can still outperform the single member. This will be further investigated in the experiment section.

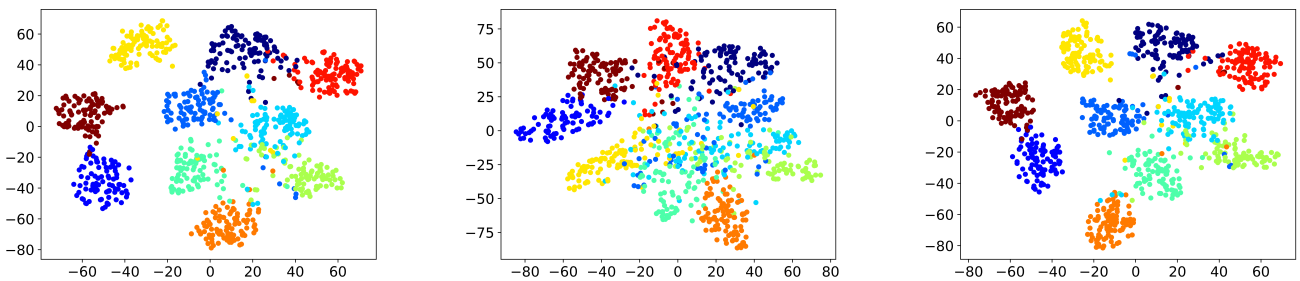
III. Experiments

In this section, we conduct a series of experiments to verify the effectiveness of the AFS model. The effects of different parameter settings on the performance are also extensively studied. Firstly, we identify some candidates that are used to form the stacking model, and evaluate the accuracy and robustness of them. The features extracted by these networks are also analyzed. Secondly, we conduct experiments to study the effect of α in training the linear merger and identify a parameter setting that can better balance the model robustness and accuracy. Thirdly, we investigate the effectiveness of the AFS model by stacking different extractors, providing practical guidelines for selecting the number of networks to be stacked. Fourthly, we compare the AFS model with the state-of-the-art defense methods, which achieves a significant improvement in terms of the trade-off between accuracy and robustness. At last, we conduct extensive ablation experiments and analyze the weight matrix of the linear merger, demonstrating the mechanism of the AFS model. Without loss of generality, we evaluate the AFS model with the adversarial perturbation bounded by infinity norm. Our code is available at https://github.com/anonymous1s8f2o/afs_code. The experimental settings are summarized as follows.

Dataset. We conduct experiments on widely adopted CIFAR-10 and CIFAR-100 datasets for evaluating adversarial robustness. CIFAR-10 contains 10 classes of 32×32 colorful images, with 50k samples for training and 10k samples for test. CIFAR-100 is similar to CIFAR-10 except for containing 100 classes.

TABLE I
ACCURACY AND ROBUSTNESS OF BASIC EXTRACTORS TRAINED WITH DIFFERENT PERTURBATION BUDGETS (%)

Model Pert.	#0 0/255	#1 1/255	#2 2/255	#3 3/255	#4 4/255	#5 5/255	#6 6/255	#7 7/255	#8 8/255	#9 9/255
Clean	94.62	93.92	93.50	92.16	91.17	90.21	88.52	87.30	85.15	83.85
PGD-10	0	15.10	31.48	39.12	44.63	47.76	50.16	52.88	54.76	55.02
PGD-20	0	10.71	27.41	36.64	42.65	46.12	48.76	51.74	54.06	54.37



(a) (b) (c)

Fig. 3. Visualization of t-SNE embeddings. Different colors denote different classes. (a) The embeddings of the features extracted by the network #1. (b) The embeddings of the features extracted by the network #8. (c) The embeddings of the concatenation of these two features along the feature dimension.

We train feature extractors on the standard training set without extra training data. The model accuracy and robustness are evaluated on the whole test set.

Network. We adopt the prevalent wide ResNet backbone with 28 layers and a width factor of 10 (WRN-28-10) [40] as the feature extractor. The hidden activations after the last average pooling layer are considered as the extracted features, which is a 640 dimensional vector. We use a linear classifier as the merger. The dimension of its inputs is equal to 640 times the number of stacked extractors.

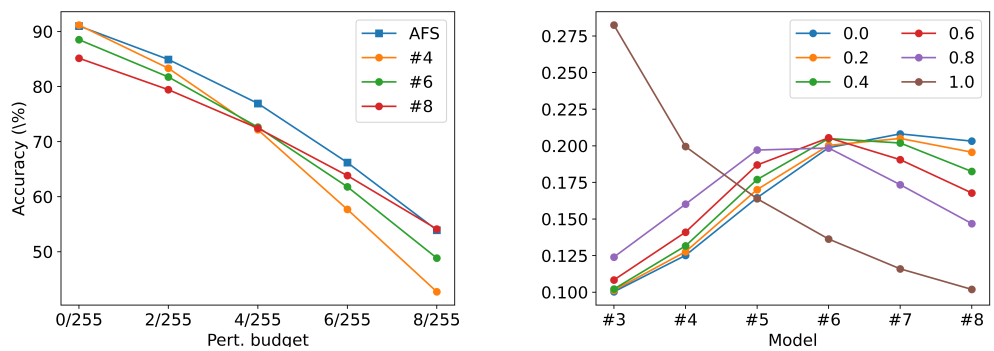
Training. The training setting of CIFAR-10 is the same as CIFAR-100. During training, standard data augmentation methods such as random horizontal flipping and random center cropping are used. We train feature extractors paired with their classifier 70 epochs with early stopping [15]. The optimizer is Stochastic Gradient Descent (SGD) with a momentum of 0.9. The initial learning rate is set to 0.1 and decays to 0.01 at 60 epoch. Weight decay factor is set to 5e-4. Batch size is set to 128. For PGD-AT, the step size is set to 2/255 and the number of iterations is set to 10. We train 10 feature extractors with different perturbation budgets ranging from 0/255 to 9/255. We train the linear merger 5 epochs. The initial learning rate is set to 0.1 and decays to 0.01 at 3 epoch. Batch size is set to 100. The perturbation budget for training the merger is set to 8/255. Other settings are the same as those of training extractors. We train multiple mergers with stacking different extractors and varied α.

Evaluation protocol. We evaluate the model robustness with PGD method [12]. The perturbation budget is set to 8/255, and the step size is 2/255. We denote by PGD-10 and PGD-20 for PGD with 10 steps and 20 steps, respectively. Stronger attack such as auto attack (AA) [41] is also applied. AA ensembles multiple gradient-based and black-box gradient-free attack methods, which can give a more reliable evaluation even facing with gradient obfuscation. For simplicity, we refer to the model robustness as the model accuracy on samples with a perturbation budget of 8/255. When evaluating the stacking model, the adversarial perturbation is generated for the whole model, thereby providing adaptive attacks.

A. Analysis of basic feature extractors

To form the stacking model, we firstly train multiple WRN-28-10 with evenly-spaced perturbation budgets from 0/255 to 9/255. The backbone of the adversarial trained WRN-28-10 is used as the feature extractor. The accuracy and robustness of these networks are summarized in Table I. It can be seen that the network trained with a larger perturbation budget has lower accuracy and stronger robustness. Trained with the perturbation budgets larger than 8/255, the growth of the model robustness becomes saturated, but the accuracy still degrades remarkably. Thus, from the perspective of a better trade-off between robustness and accuracy, we select the networks trained with the perturbation budgets less than 9/255 as the candidates of the feature extractors.

As a preliminary analysis of the features extracted by these candidates, we visualize the 2-dimensional embeddings of some benign test samples using t-SNE method. As shown in Figure 3, we present the t-SNE embeddings of the features generated by the network #1 and the network #8, and the embeddings of the concatenation of these features along the feature dimension, respectively. The features extracted by the network trained with small perturbation budgets are separated well, but the robustness of these networks is lower. In contrast, the robustness of the network trained with large perturbation budgets is stronger, but



(a) (b)

Fig. 4. (a) Robustness of different models under different perturbation budgets. (b) The importance ratio of each stacked network under different α.

the features extracted by these networks are not well separated, thus leading to lower accuracy on benign samples. Notably, the features formed by simply concatenating these two types of features are also separated well, indicating improved accuracy. The feature concatenation with a learnable merger forms the proposed AFS model, whose robustness and accuracy will be further investigated in the following sections.

B. *Effect of α in training the merger*

To better select features generated by different extractors for final predictions, a learnable merger is deployed to fuse the features. In this part, we study the effect of α that balances the robustness and accuracy of the whole stacking model when training the merger. In these experiments, we select 6 networks trained with perturbation budgets ranging from 3/255 to 8/255 as the feature extractors. Based on these features extracted from both benign samples and adversarial samples, we train several linear mergers to give final predictions. The accuracy and robustness of the stacking models trained with varied α are summarized in Table II. Similar to the results in Table I, there is a trade-off between accuracy and robustness: the larger the α, the higher the accuracy and the lower the robustness. However, it should be noted that the trade-off is significantly improved. The stacking model trained with an α of 1 has an accuracy of 92.60% on benign samples and a robustness of 46.76%. In contrast, the single adversarially trained network with the same level of accuracy has a robustness of 36.64%. The stacking model with the strongest robustness ($\alpha = 0.0$) exceeds the single network with the same level of robustness (network #8) more than 6% on accuracy and 1% on robustness. Additionally, the variance of the robustness and accuracy of the stacking model with different α is not large, leading to easy parameter selection in practical use. We evaluate the robustness of the stacking model with an α of 0.5 under different levels of perturbation budgets, and compare it with several single networks. The experimental results are depicted in Figure 4(a). The AFS model significantly improves the accuracy without sacrificing the robustness. Practically, the α can be set to 0.5 for a better trade-off.

To investigate how the features of different extractors influence the performance, we analyze the characteristic of the weight matrix of the linear merger trained with different α. The weight matrix can be divided into several sub-matrices according to the corresponding weighted features. The sum of the absolute values of each sub-matrix reflects the importance of the feature generated by the corresponding extractor. Thus, we calculate the normalized sum as the importance ratio and plot it versus the stacked single model under different settings of α in Figure 4(b). A smaller α promotes the merger to rely on more robust features, thus facilitating stronger robustness. Notably, the features extracted by the network trained with moderate perturbation budgets, such as network #5, #6, and #7, account for a larger proportion to give final predictions. In contrast, the features extracted by the network trained with too small perturbation budgets such as network #3 have limited impact on the predictions. Additionally, the importance ratio of the features of the network #8 varies greatly under different settings of α, indicating the strong relation between these features and the model robustness. These results also demonstrate that the AFS model indeed utilizes diverse features generated by different extractors to give predictions.

TABLE II
ACCURACY AND ROBUSTNESS OF STACKING MODELS WITH DIFFERENT α (%)

α	0.0	0.2	0.4	0.6	0.8	1.0
Clean	90.06	90.55	90.80	91.01	91.70	92.60
PGD-10	56.16	55.74	55.21	54.61	53.18	48.20
PGD-20	54.94	54.65	53.79	53.33	51.89	46.76

C. *Effect of stacking different extractors*

In this section, we study the effect of the number of different networks to be stacked. In this experiment, we select 9 basic networks (#0˜#8) as the candidates to be stacked. We use a 9-dimensional binary vector to denote whether a single network is adopted. For example, '100000001' denotes the setting that network #0 and #8 are adopted. All the stacking models are trained with an α of 0.5. The accuracy and robustness of different settings are summarized in Table III. Generally, the more networks are stacked, the higher accuracy and the stronger robustness are. However, the computation budgets and storage usage increase linearly with the number of stacked networks. Moreover, the performance improvement of the AFS model will gradually saturate as stacking more networks. From the both consideration of performance and cost, we choose to stack the 6 networks (#3˜#8) as the default setting. These networks can provide diverse features with varied levels of accuracy and robustness. Additionally, as discussed above, the perturbation budgets used to train the feature extractors should include both small values and large values.

TABLE III
ACCURACY AND ROBUSTNESS OF STACKING DIFFERENT NETWORKS (%)

Setting	100000001	100010001	101010101	111111111
Clean	86.25	90.00	90.79	91.08
PGD-10	53.30	53.01	53.77	54.90
PGD-20	52.44	51.96	52.44	53.38

D. Comparison with state-of-the-art methods

The default AFS model is evaluated on CIFAR-10 and CIFAR-100 with advanced attacks such as AA, and compared with state-of-the-art defense methods, whose results are summarized in Table IV and Table V. The trade-off is measured by the maximum of the mean of accuracy and robustness. The results listed at the bottom part of the table are our implementations. Notably, the robustness of the Adversarial Vertex Mixup method (AVM) degrades significantly when evaluated by AA, indicating that it relies on gradient obfuscation. The AFS model in this work is based on PGD-AT. Compared to the PGD-AT with a perturbation budget of 8/255, our AFS model achieves a benign accuracy improvement of $\sim$6% on CIFAR-10 and $\sim$9% on CIFAR-100 with comparable or even stronger robustness. These results demonstrate a better trade-off between accuracy and robustness, indicating that we can obtain a model with both high accuracy and strong robustness without extra training data. Additionally, as shown in Figure 1, the AFS model outperforms other methods in terms of the trade-off. Since the AFS model is orthogonal to the adversarial training methods that are used to train the extractors. It is believed that AFS model can be further combined with more advanced adversarial training methods and other defense methods to achieve more robust and more accurate models.

TABLE IV
COMPARISON OF DIFFERENT METHODS ON CIFAR-10 (%)

Method	Clean	PGD	AA	Trade-off
FAT [17]	84.39	57.12	53.51	68.95
TLA [18]	86.21	50.03	47.41	66.81
TRADES [16]	84.92	56.43	53.08	69
LBGAT+TRADES [19]	81.98	57.78	53.14	67.56
AVM [42]	93.14	58.04	27.06	60.1
PGD-AT [15]	85.15	54.06	51.02	68.08
AFS(ours)	91.01	53.94	52.55	**71.78**

TABLE V
COMPARISON OF DIFFERENT METHODS ON CIFAR-100 (%)

Method	Clean	PGD	AA	Trade-off
LBGAT+TRADES [19]	60.43	35.50	29.34	44.88
AVM [42]	73.82	32.95	13.61	43.71
PGD-AT [15]	61.39	30.24	26.58	43.98
AFS(ours)	70.28	28.43	26.97	**48.62**

E. Ablation study

To investigate the key points that make AFS model work, we conduct several ablation experiments on CIFAR-10. The parameters, computational costs, and performance, are presented in Table VI. The computational cost is evaluated with a input size of $1 \times 3 \times 32 \times 32$. All the models are trained by PGD-AT method. The baseline model is a WRN-28-10 trained with a perturbation budget of 8/255. For comparison, we train a single WRN-28-10 with random perturbation budgets ranging from 3/255 to 8/255. The accuracy of this model improves about 4% at the cost of the same level robustness degradation, thus failing to improve the trade-off between accuracy and robustness than the baseline model. The results indicate that training a single network with different perturbation budgets cannot alleviate the trade-off.

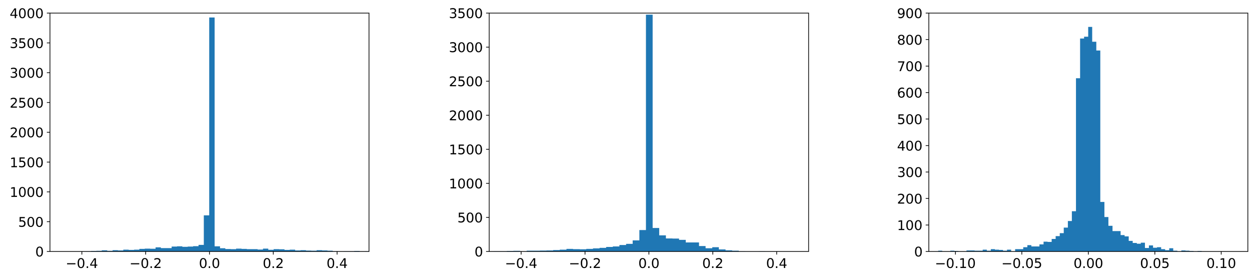
To investigate the effect of the model capacity on the trade-off, we train a large WRN-28-30 model, the parameters and computational costs are 1.5 times larger than the default AFS model for comparison. The accuracy of this large model improves about 2.5% without degrading robustness. The results demonstrate that increasing model capacity can improve the trade-off within certain limits but the improvement is not remarkable. In contrast, with even fewer parameters and smaller computational costs than the above mentioned large model, the AFS model can significantly improve the accuracy about 6% without degrading robustness. Considering the results of these ablation experiments, we can infer that separately training multiple networks with different levels of perturbation budget and then merging the extracted features for predictions are both key ingredients that make the AFS model work.

TABLE VI
COMPARISON OF DIFFERENT ABLATION SETTINGS

Setting	Para.(M)	Flops(G)	Clean	PGD-20
Baseline	36.47	5.24	85.15	54.06
Rand.pert.	36.47	5.24	89.02	49.73
Large model	328	47.04	87.67	54.08
AFS	218.82	31.44	91.01	53.94

F. Weight analysis and gradient visualization

We conduct more analysis on the weight matrix of the linear merger, which is the key component of the stacking model. We present the histogram of the weight matrix of the AFS model with the default setting (Figure 5(c)), and compare it with that of the weight matrix of the classifier of



(a) (b) (c)

Fig. 5. Histograms of the weight matrix of linear classifiers. (a) The histogram of the network #1. (b) The histogram of the network #8. (c) The histogram of the AFS model with the default setting.

the network #1 and the network #8 (Figure 5(a),5(b)). The weight matrices of the network #1 and the network #8 are relatively sparse and have a small part of larger values, indicting that the corresponding classifier heavily depends on a few key features to give predictions. The small part of key features are usually not both accurate and robust, leading to a trade-off between accuracy and robustness. In contrast, the weight matrix of the linear merger of the AFS model is distributed more evenly, demonstrating that the AFS model uses more diverse features rather than excessively relies on a small part of key features. The concatenated features with varied levels of accuracy and robustness can form both accurate and robust features, which is not easily realized using a single network especially under the condition of limited training data. This characteristic facilitates the AFS model to achieve a much better trade-off between accuracy and robustness.

To show the key features that the model relies on to give predictions, we visualize the gradient of the AFS model w.r.t. the input, and compare it with that of its individual constituent networks. As shown in Figure 6, the gradients of the AFS model align well with the human perception. Compared to the gradients of its constituent networks, the gradients of the AFS model are more consistent to the shape of the objects, which are usually accurate and robust features for predictions. Notably, the gradient visualization also indicates that the AFS model does not rely on gradient obfuscation, for the gradient contains interpretable semantic information.

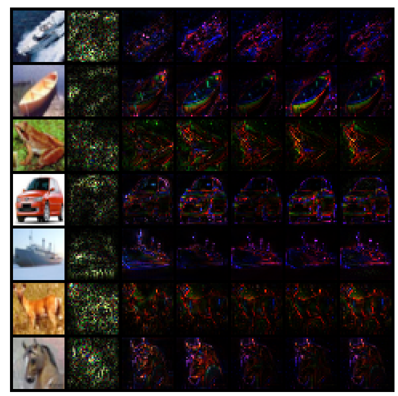


Fig. 6. Gradient visualization of different models. For better visualization, we take the absolute value of the gradient, normalize it to [0,1], and perform a log transformation. From left to right, the images are samples form CIFAR-10 dataset, the corresponding gradient of network #0, #2, #4, #6, #8, and the AFS model stacking these individual network.

IV. Related work

Attack methods. The adversarial attack methods can be generally categorized into gradient-based methods and gradient-free methods. [43] proposed the Fast Gradient Sign Method (FGSM) that uses the gradient of the cost function of the network w.r.t. the inputs to efficiently generate adversarial perturbation. Basic Iterative Method (BIM) extended the FGSM with multi-step gradient update to give a stronger attack [44]. To overcome the problem of suboptimal adversarial perturbation, a Projected Gradient Descent method (PGD) with random initialization was adopted to further enhance the attack strength [12]. [41] proposed two variants of PGD attack with budget-aware step size and a difference of logits ratio loss, which provides a stronger parameter-free attack that is immune to the gradient masking caused by the logits shifting and scaling. [13] proposed a query-efficient gradient-free adversarial attack via random search, which can penetrate gradient obfuscation defense. Auto-attack (AA) [41] ensembles multiple diverse gradient-based and gradient-free methods to provide a more reliable evaluation of model robustness.

Defense methods. The adversarial defense methods can be generally categorized into denoising-based, gradient obfuscation-based, and adversarial training-based methods. Feature squeezing was proposed to reduce the search space available to an adversary by color depth reduction and spatial smoothing, thus hardening DNNs by detecting adversarial examples [45]. [20] tried to eliminate adversarial perturbation by a high-level representation guided denoiser, which overcomes the error amplification effect of the standard denoiser. [46] proposed stochastic activation pruning to defense gradient-based attack by randomizing gradients. Adversarial training [43], [12] feeds on-line generated adversarial samples into the training set, promoting networks to be immune to the adversarial perturbation, which remains one of the most effective defense methods. [16] proposed TRADES that optimizes networks by a standard classification loss for accuracy and a regularization term to minimize the difference between the prediction of natural samples and adversarial samples. [15] identified an overfitting problem in adversarial training, and demonstrated that the performance improvement achieved by TRADES than vanilla PGD-AT largely relies on early stopping. To address the overfitting in adversarial training, [42] proposed the Adversarial Vertex Mixup (AVM) method that adopts a soft-labeled data augmentation scheme, which achieves significant improvement on both model accuracy and robustness. However, this method results in gradient obfuscation due to the heavy label smoothing, and can be defeated by adaptive attacks such as AA.

Adversarial ensembles. Ensemble learning is a powerful method to improve model performance in classic machine learning, whose success relies heavily on the diversity of the features learned by the members. A lot of work incorporates ensemble methods to further improve model robustness. [47] proposed an ensemble adversarial training method that incorporates the adversarial samples transferred from other pre-trained models, exhibiting stronger robustness to black-box attacks with a slight sacrifice of accuracy. [48] proposed to train the entire ensemble as a single model and demonstrated that the adversarial training of the ensemble provides stronger robustness than the ensemble of adversarially trained models. Another series of work tried to improve the diversity of the ensembles by some regularization schemes to enhance the robustness [49], [50]. Most of these methods aim to enhance model robustness by an ensemble of multiple networks, but the improvement is not substantial due to the low diversity of the ensembles. In contrast, inspired by the stacking method, our AFS model fuses several separately trained networks with varied levels of robustness and accuracy to achieve a better trade-off. The stacking model has both higher accuracy and stronger robustness enabled by different models.

V. Conclusion

Developing accurate and robust networks under the condition of limited training data is a crucial problem, which remains unsolved. In this work, to alleviate the trade-off between accuracy and robustness, we propose the AFS model that deploys a linear merger to fuse the features extracted by multiple networks adversarially pre-trained with different levels of perturbation budgets. The stacking model can comprehensively utilize both robust features and accurate features, thus significantly alleviating the trade-off and facilitating both accurate and robust predictions. We verify the effectiveness of the AFS model on CIFAR-10 and CIFAR-100 datasets with advanced attack methods, which significantly outperforms the state-of-the-art methods in terms of the trade-off between accuracy and robustness. The experimental results demonstrate the feasibility to obtain models with both high accuracy and strong robustness without extra training data, which promotes the development of practical robust models. The AFS model provides a new route to simultaneously improve the model robustness and accuracy with stacking-based methods. In the future, the AFS model is expected to be combined with more advanced defense methods and trained on more data to further improve the accuracy and robustness.

VI. Acknowledgements

This work is partially supported by the Project of NSFC No. 61836004 and the Brain-Science Special Program of Beijing under Grant Z181100001518006.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *14th European Conference on Computer Vision, ECCV 2016*, 2016, pp. 21–37.

[4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2015.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2018, pp. 4171–4186.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, and D. Amodei, “Language models are few-shot learners,” 2020.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*, 2014.

[9] J. Su, D. V. Vargas, and S. Kouichi, “One pixel attack for fooling deep neural networks,” *IEEE Transactions on Evolutionary Computation*, 2017.

[10] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” 2017.

[11] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, “Boosting adversarial attacks with momentum,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.

[13] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient black-box adversarial attack via random search,” in *European Conference on Computer Vision*. Springer, 2020, pp. 484–501.

[14] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, “Query-efficient hard-label black-box attack: An optimization-based approach,” *arXiv preprint arXiv:1807.04457*, 2018.

[15] L. Rice, E. Wong, and Z. Kolter, “Overfitting in adversarially robust deep learning,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 8093–8104.

[16] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, “Theoretically principled trade-off between robustness and accuracy,” *arXiv preprint arXiv:1901.08573*, 2019.

[17] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, “Attacks which do not kill training make adversarial learning stronger,” *arXiv preprint arXiv:2002.11242*, 2020.

[18] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, “Metric learning for adversarial robustness,” in *Advances in Neural Information Processing Systems*, 2019, pp. 480–491.

[19] J. Cui, S. Liu, L. Wang, and J. Jia, “Learnable boundary guided adversarial training,” *arXiv preprint arXiv:2011.11164*, 2020.

[20] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, “Defense against adversarial attacks using high-level representation guided denoiser,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

[21] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, “A study of the effect of jpg compression on adversarial images,” *arXiv preprint arXiv:1608.00853*, 2016.

[22] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, “Feature denoising for improving adversarial robustness,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 501–509.

[23] Anonymous, "Improving adversarial robustness via channel-wise activation suppressing," in *Submitted to International Conference on Learning Representations*, 2021, under review. [Online]. Available: https://openreview.net/forum?id=zQTezqCCtNx

[24] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[25] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[26] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv preprint arXiv:1710.10766*, 2017.

[27] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[28] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *arXiv preprint arXiv:2002.08347*, 2020.

[29] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5286–5295.

[30] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344*, 2018.

[31] K. Dvijotham, S. Gowal, R. Stanforth, R. Arandjelovic, B. O'Donoghue, J. Uesato, and P. Kohli, "Training verified learners with learned verifiers," *arXiv preprint arXiv:1805.10265*, 2018.

[32] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[33] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "Mma training: Direct input space margin maximization through adversarial training," in *International Conference on Learning Representations*, 2019.

[34] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *arXiv preprint arXiv:1805.12152*, 2018.

[35] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri, "A closer look at accuracy vs. robustness," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[36] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 192–11 203.

[37] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, and P. Liang, "Understanding and mitigating the tradeoff between robustness and accuracy," *arXiv preprint arXiv:2002.10716*, 2020.

[38] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli, "Uncovering the limits of adversarial training against norm-bounded adversarial examples," *arXiv preprint arXiv:2010.03593*, 2020.

[39] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, 2019, pp. 125–136.

[40] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[41] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," *arXiv preprint arXiv:2003.01690*, 2020.

[42] S. Lee, H. Lee, and S. Yoon, "Adversarial vertex mixup: Toward better adversarially robust generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 272–281.

[43] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[44] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[45] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[46] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," *arXiv preprint arXiv:1803.01442*, 2018.

[47] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[48] E. Grefenstette, R. Stanforth, B. O'Donoghue, J. Uesato, G. Swirszcz, and P. Kohli, "Strength in numbers: Trading-off robustness and computation via adversarially-trained ensembles," *arXiv preprint arXiv:1811.09300*, 2018.

[49] A. Mirzaeian, M. Sabokrou, M. Khalooei, J. Kosecka, H. Homayoun, T. Mohsening, and A. Sasan, "Learning diverse latent representations for improving the resilience to adversarial attacks," *arXiv preprint arXiv:2006.15127*, 2020.

[50] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," *arXiv preprint arXiv:1901.08846*, 2019.