FACM: Correct the Output of Deep Neural Network with Middle Layers Features against Adversarial Samples

Xiangyuan Yang, Jie Lin, Hanlin Zhang, Xinyu Yang, and Peng Zhao

Abstract—In the strong adversarial attacks against deep neural network (DNN), the output of DNN will be misclassified if and only if the last feature layer of the DNN is completely destroyed by adversarial samples, while our studies found that the middle feature layers of the DNN can still extract the effective features of the original normal category in these adversarial attacks. To this end, in this paper, a middle Feature layer Analysis and Conditional Matching prediction distribution (FACM) model is proposed to increase the robustness of the DNN against adversarial samples through correcting the output of DNN with the features extracted by the middle layers of DNN. In particular, the middle Feature layer Analysis (FA) module, the conditional matching prediction distribution (CMPD) module and the output decision module are included in our FACM model to collaboratively correct the classification of adversarial samples. The experiments results show that, our FACM model can significantly improve the robustness of the naturally trained model against various attacks, and our FA model can significantly improve the robustness of the adversarially trained model against white-box attacks with weak transferability and black-box attacks where FA model includes the FA module and the output decision module, not the CMPD module.

Index Terms—Adversarial samples, correction model, diversity property, deep neural network.

I. INTRODUCTION

Deep neural networks (DNN) were shown to be susceptible against adversarial examples: adversarial perturbed samples will cause mis-classification while being nearly "imperceptible", i.e., very close to the original samples. Ding et al. [1] has explored that the added imperceptible noise of the input will be amplified layer by layer till the output layer makes a wrong classification. However, the noise generated by white-box attacks with weak transferability (e.g., \( \text{CW}_\infty \)) [2], \( \text{CW}_2 \) [2] and DeepPool L2 [3]) may just destroy the output layer of DNN to make a wrong classification, while the output of the middle layers of DNN still contains the correct features of the original category. Hence, the features extracted by the middle layers of DNN can be used to correct the output of the DNN against adversarial samples generated by white-box attacks with weak transferability.

Corresponding author: Jie Lin.
Xiangyuan Yang, Jie Lin, Xinyu Yang and Peng Zhao are with School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China (e-mail: ouyang_xy@stu.xjtu.edu.cn; {jielin, xyyphd, p.zhao}@mail.xjtu.edu.cn)
Hanlin Zhang is with Qingdao University, Qingdao, China (e-mail: hanlin@qdu.edu.cn)

Adversarial training (AT) is currently one of the most effective defenses against adversarial attacks for deep learning models. Various variants of adversarial training methods are proposed. For example, Adversarial Training with Hypersphere Embedding (ATHE) [4], Fast Adversarial Training (Fast-AT) [5], Friendly Adversarial Training (FAT) [6], Adversarial Training with Transferable Adversarial examples (ATTA) [7], You Only Propagate Once (YOPO) [8], Fair-Resilient Robustness (FRR) [9], TRadeoff-inspired Adversarial DEFense via Surrogate-loss minimization (TRADES) [10] were proposed to enhance the robustness of DNN. However, these adversarial training methods focused on accelerating adversarial training and mitigating low efficiency, low generalization and unfairness, but remain to achieve poor performance against both strong white-box and black-box attacks, e.g., \( \text{CW}_\infty \), square and NATACK, etc.

Recently, Bai et al. [11] and Yan et al. [12] investigated the adversarial robustness of DNNs from the perspective of channel-wise activations, which found that adversarial training can align the activation magnitudes of adversarial examples with those of their natural counterparts, but the over-activation of adversarial examples is still exist. To further improve the robustness of DNNs, Bai et al. [11] proposed Channel-wise Activation Suppressing (CAS) to suppress redundant activation of adversarial examples. Yan et al. [12] proposed Channel-wise Importance-based Feature Selection (CIFS) to suppress the channels that are negatively-relevant to predictions when processing adversarial examples. However, these methods also remain to achieve poor performance against both white-box attacks with weak transferability and black-box attacks.

To fill this gap, we propose a middle Feature layer Analysis and Conditional Matching prediction distribution (FACM) model to correct the output of DNN against adversarial samples with features extracted by the middle layers of DNN. The proposed FACM model consists of three modules: the middle feature layer analysis (FA) module, the conditional matching prediction distribution (CMPD) module and the output decision module, respectively. The FA module directly takes the output of the middle layers as the input to build the middle layer classifiers, and then the middle layer correction model can be constituted by the ensemble of the middle layer classifier and the DNN model. The CMPD module is used to build an autoencoder that is conditioned on the outputs of several middle layer classifiers, in order to reconstruct the input of the DNN with the objective of mitigating the effect of the perturbation. Particularly, the conditions involved in the
The autoencoder can improve the efficiency of reconstruction. In addition, the loss function of the autoencoder is generated by the Kullback-Leibler divergence between the predictions of the DNN on the original and reconstructed inputs. The output decision module can randomly select one or more correction models generated in both the FA and CMPD modules to correct the output through weighted fusion of these selected correction models. Because correction and diversity properties are introduced in FACM, the DNN’s defense ability against both black-box attacks and white-box attacks with weak transferability can be improved. Our main contributions are summarized as:

- The correction property that the middle layers of the DNN can still extract the effective features of the original category of the adversarial sample is proposed and theoretically proved that can be used to correct the DNN’s output.
- The middle Feature layer Analysis and Conditional Matching prediction distribution (FACM) model is proposed to correct the output of the DNN against adversarial samples with features extracted by middle layers of the DNN.
- The diversity property that the correction model in FACM is diverse against adversarial samples is proposed. The greater the attack strength, the more obvious the diversity.
- Extensive experiments are conducted to evaluate the effectiveness of our FACM model and FA model, and the evaluation results show that our FACM model can efficiently improve the classification accuracy of the naturally trained DNN model against various attacks and FA model can efficiently improve the classification accuracy of the adversarially trained DNN model against white-box attacks with weak transferability and black-box attacks. Note that FA model includes the FA module and the output decision module, not the CMPD module.

II. PRELIMINARIES

In this section, the adversarial training, TRADES and matching prediction distribution, are mentioned briefly.

Standard Adversarial Training: Adversarial training is an effective defense method to train robust DNN models against adversarial attacks. With adversarial attacks as the data augmentation method, the model trained with adversarial examples achieves considerable robustness. Madry et al. [13] first formulated adversarial training as a min-max optimization problem:

$$
\min_{f \in H} \mathbb{E}_{(x,y) \sim D} \left[ \max_{\delta \in B} L(f(x + \delta), y) \right]
$$

where $H$ is the hypothesis space, $D$ is the distribution of the training dataset, $L$ is a loss function, and $B$ is the allowed perturbation space that is usually selected as an $L_p$ norm ball around $x$. The basic idea of adversarial training is to find a perturbed image $x + \delta$ based on a given original image $x$, with the objective of maximizing the loss with respect to correct classification. Then, the model is trained by generated adversarial examples.

TRADES: To trade off natural and robust errors, Zhang et al. [10] trained a DNN model with both natural and adversarial data, and changed the min-max formulation as follows:

$$
\min_{f \in H} \mathbb{E}_{(x,y) \sim D} \left[ L(f(x), y) + \beta \cdot \max_{\delta \in B} KL(f(x + \delta), f(x)) \right]
$$

where $L_{KL}$ is the Kullback-Leibler loss, $\beta$ is a regularization parameter that controls the trade-off between standard accuracy and robustness. As shown in Eq. [2], when $\beta$ increases, standard accuracy will decrease while robustness will increase, and vice versa.

Matching Prediction Distribution: To correct the output of the DNN against adversarial samples, Vacanti et al. [14] presents a novel adversarial correction model including an autoencoder that is trained with a custom loss function generated by the Kullback-Leibler divergence between the classifier predictions on the original and reconstructed instances:

$$
\min_{\phi^e, \theta^e} \mathbb{E}_{(x,y) \sim D} \left[ L_{KL}(f(x), f(\Psi(\Phi(x)))) \right]
$$

where $\Phi(\cdot)$ and $\Psi(\cdot)$ are the encoder and decoder of autoencoder, respectively, $\theta^e$ and $\theta^o$ are the weights of $\Phi(\cdot)$ and $\Psi(\cdot)$ that need to be optimized. This method is unsupervised, easy to train and does not require any knowledge about the underlying attack.

III. PROPOSED APPROACH

In this section, we present the FACM model, which consists of three modules: the middle feature layer analysis (FA) module, the conditional matching prediction distribution (CMPD) module and the output decision module. In this section, first the construction steps and the working principles of the FACM model is introduced, and then the working principle and theoretical analysis of FA, CMPD, output decision modules and FACM model are described in details.

A. Construction Steps and Working Principles of FACM Model

Construction Steps: Firstly, the middle layer correction model, denoted as $\{(f + f_i)(\cdot)\}_{1 \leq i \leq n - 1}$, is constructed. Then, the conditional matching prediction distribution correction model, denoted as $\{g_i(\cdot)\}_{0 \leq i \leq n - 1}$, is constructed. Finally, the output decision model, denoted as $h(\cdot)$, is constructed.

Working Principles: Taking an instance $(x, y)$ as an example, firstly the weights vector $\omega = S(h(x))$ is calculated by the output decision model, where $S(\cdot)$ is the sigmoid function. Then, $\omega$ is normalized to $\bar{\omega}$. Thirdly, $\tau$ correction models are randomly extracted from all correction models (i.e. FACM, including the DNN model itself) according to probability vector $\bar{\omega}$. Finally, the average output of $\tau$ correction models is taken as the final prediction of the instance $x$.

B. Middle Feature Analysis Module

**Definition 1:** (Middle layer classifier) For an instance $(x, y) \sim D$, the output of the $i^{th}$ middle layer is denoted
Therefore, adversarial samples only make the output layer of the DNN classify the input $x$ into wrong categories other than $y$. The following two cases occur because the disturbance against the middle layer is less than that on the last layer.

Definition 2: (Middle layer correction model) For the DNN $f()$ and the $i$th middle layer classifier $f_i()$, the $i$th middle layer correction model, denoted as $(f + f_i)()$, can be constituted by the ensemble of $f()$ and $f_i()$.

In the middle layer correction model $(f + f_i)()$, the middle layer classifier $f_i()$ can correct the output of the DNN $f()$ without any effect on the classification accuracy of $f()$ on normal samples (i.e., the classification accuracy of $(f + f_i)()$ is basically equal to that of $f()$ on normal samples).

In the following, Definitions 3 and 4 are mentioned to define the classification sequence and corresponding classification sub-space which are used in the proof of Proposition 1 to analyze the effectiveness of the middle layer correction model.

Definition 3: (Classification sequence) For an instance $(x, y) \sim D$, the top $i$ middle layer classifiers’ classification sequence $S_i$ can be denoted as $[\arg\max(f_1(x)), \arg\max(f_2(x)), \ldots, \arg\max(f_i(x))]$.

Definition 4: (Classification sequence sub-space) In the input space $D$ of the DNN $f()$, the sub-space $D^i_k$ indicates that the sequence $[\arg\max(f_1(x)), \arg\max(f_2(x)), \ldots, \arg\max(f_i(x))]$ is equal to $S_i$ when $x \in D^i_k$.

The following two cases discuss that the correct classification of the classifiers $f_i(\hat{x})$ or $f_j(\hat{x})(j < i)$ can make up for the wrong classification of the classifier $f_{i+1}(\hat{x})$ where $\hat{x}$ is an adversarial examples, $k$ is the original category:

- **Case 1:** $\hat{x} \notin D^{[S_{i-1}, k]}_{i+1} \land \hat{x} \notin \left(D^{[S_{i-1}, k]}_i - D^{[S_{i-1}, k]}_{i+1}\right)$, where $\hat{x}$ causes the classifier $f_{i+1}$ to misclassify, while $\hat{x}$ is still belong to $D^{[S_{i-1}, k]}_i$ so that the classifier $f_i$ can make the correct classification to $\hat{x}$.

- **Case 2:** $\exists j < i, \hat{x} \notin D^{[S_{i-1}, k]}_{i+1} \land \hat{x} \notin \left(D^{[S_{i-1}, k]}_i - D^{[S_{i-1}, k]}_{i+1}\right) \land \hat{x} \notin D^{[S_{i-1}, [1 \cdot j - 1], k]}_j$, where $\hat{x}$ causes the classifier $f_{j+1}, \ldots, f_{i-1}$ to misclassify, while $\hat{x}$ is still belong to $D^{[S_{i-1}, [1 \cdot j - 1], k]}_j$ so that the classifier $f_j$ can make the correct classification to $\hat{x}$, where $S_{i-1}[1 \cdot j - 1]$ denotes the sub-sequence with indexes within $[1 \cdot j - 1]$. These two cases occur because the disturbance against the original normal sample is very small and as close to the original normal sample as possible in the definition domain. Therefore, adversarial samples only make the output layer of deep neural network output wrong classification, but have little impact on the middle layer classifier (i.e., $f_i, i \leq n, n$ is the depth of the DNN). Proof over.

However, a problem has arisen as follows: the middle layer classifier cannot completely make up for all adversarial samples because the outputs of the middle layer classifiers may not have the effective features of the original normal category. To solve the problem, a conditional matching prediction distribution module is proposed in Section III-C which transforms the adversarial sample $\hat{x}$ into $\hat{x}'$ that can be classified correctly by the output layer of DNN.

C. Conditional Matching Prediction Distribution Module

The conditional matching prediction distribution (CMPD) correction model is actually an autoencoder, in which the input consists of a sample $x$ and the concatenation of the outputs of several middle layer classifiers, and the output is a transformed sample $x'$. Note that, the different number of middle layers features as the conditions constructs the different CMPD correction model.

Definition 5: (The $i$th CMPD correction model $g_i()$) which emerges an encoder $\Phi()$, a decoder $\Psi()$ and the outputs of the top $i$ middle layer classifiers as the conditions, i.e., $g_i = \Psi(\Phi([f_1(), \ldots, f_i()])[f_i(), \ldots, f_1()])$. The $g_i$ is built by:

$$
\min_{g_i \in \mathcal{H}_g} \mathbb{E}_{(x,y) \sim D} [\text{KL}(f(x), f(g_i(x)))]
$$

where $\mathcal{H}_g$ is the hypothesis space of $g_i$.

Definition 6 defines the composition and objective function of the $i$th CMPD correction model. Note that, in our FACP model, because the dimensions of the input conditions are different, all the CMPD correction models share weights except for layers with conditional inputs in the encoder and decoder. When CMPD correction model $g_0$ does not have any conditional input, it can be considered as the MPD correction model $g() = \Psi(\Phi())$ (i.e., $g_0 = g$). Proposition 2 verify that the CMPD correction model is easier to converge than MPD correction model.

Proposition 2: For a learning task, in the case of conditional constraints, the less the mapping of learning, the less the complexity of the task, thereby leading to the faster and more stable convergence of the model training. With the conditions of given outputs of $f_1(), f_2(), \ldots, f_i()$, our CMPD correction model $g_i$ simplifies the difficulty of the autoencoder learning task in comparison with MPD correction model $g$.

Proof 2: For the MPD correction model $g$, to correctly classify the sample $x$, the compound function $f \circ g(x)$ needs to find a classification sequence $[\arg\max(f_1 \circ g(x)), \arg\max(f_2 \circ g(x)), \ldots, \arg\max(f_n \circ g(x))]$ from $m^n$ classification sequences where $m$ is the number of categories and $n$ is the layer depth of the DNN. While, for the CMPD correction model $g_i$, to correctly classify the sample $x$, because the classification sequence $S_i$ is the known conditions, the compound function $f \circ g_i(x)$ only needs to find a classification subsequence $\arg\max(f_{i+1}(g_i(x)), \ldots, \arg\max(f_n(g_i(x))))$ from $m^{n-1}$ classification sequences. According to the concept of the hypothesis space and growth function in computational learning theory [5], the growth functions of the hypothesis space of $g$ and $g_i$ are calculated as Eq. 5, where the hypothesis...
space is the set of all possible mappings and the growth function represents the maximum number of possible results that $M$ samples are labeled in the hypothesis space.

\begin{equation}
\begin{cases}
\Pi_{H_g}(M) = M^{m^n} \\
\Pi_{H_{g_i}}(M) = M^{m^{n-i}}
\end{cases}
\end{equation}

where $H_g$ and $H_{g_i}$ denote hypothesis space of the MPD correction model $g$ and the CMPD correction model $g_i$, respectively. $\Pi_{H_g}$ and $\Pi_{H_{g_i}}$ denote the growth function of $H_g$ and $H_{g_i}$ respectively, $M$ is the size of train dataset. Due to $\Pi_{H_g}(M) \gg \Pi_{H_{g_i}}(M)$, the CMPD correction model converges faster than the MPD correction model.

According to Theorem 12.2 [15], for any $M$, $0 < \nu < 1$, $g \in H_g$ and $g_i \in H_{g_i}$, we have

\begin{equation}
\begin{cases}
P(|E(g) - \hat{E}(g)| > \nu) \leq 4\Pi_{H_g}(2M) \exp\left(-\frac{M\nu^2}{8}\right) \\
P(|E(g_i) - \hat{E}(g_i)| > \nu) \leq 4\Pi_{H_{g_i}}(2M) \exp\left(-\frac{M\nu^2}{8}\right)
\end{cases}
\end{equation}

where $P(|E(g) - \hat{E}(g)| > \nu)$ and $P(|E(g_i) - \hat{E}(g_i)| > \nu)$ respectively denote the probabilities of the MPD correction model $g$ and the CMPD correction model $g_i$ which do not converge to the expectation error $\nu$. Due to $4\Pi_{H_{g_i}}(2M) \exp\left(-\frac{M\nu^2}{8}\right) \ll 4\Pi_{H_g}(2M) \exp\left(-\frac{M\nu^2}{8}\right)$, the possible values and ranges of $P(|E(g_i) - \hat{E}(g_i)| > \nu)$ are smaller, i.e., $g_i$ has a smaller probability of non-convergence. Therefore, the CMPD correction model converges more stable than the MPD correction model. Proof over.

When $i \geq \arg\max \{\arg\max f_j(x) \neq \arg\max f_j(\hat{x})\}$, the CMPD correction model $g_i$ cannot be used to transfer the adversarial sample $\hat{x}$ because the classification sequence of the conditions hardly changes, i.e., the sequence $[\arg\max f_j(x), \arg\max f_j(x), \cdots, \arg\max f_j(x)]$ hardly recovers to the original sequence $[\arg\max f_j(x), \arg\max f_j(x), \cdots, \arg\max f_j(x)]$. Meanwhile, according to two cases analysis in the proof of Proposition [1] different adversarial samples need different middle layer correction models. Hence, output decision module in Section [III-D] calculate the weights of each correction model in the FA and CMPD modules for each adversarial sample, then select the correction model with the weights.

**D. Output Decision Module**

Because the adversarial samples will make the deep neural network take an unusual activation path [16], the outputs of all middle layer classifier can be used to decide whether use the correction model or not and determine the weight of each correction model. Therefore, for a sample $(x, y)$, the training data of output decision model $h(\cdot)$ can be defined as $X^h(x)$ and $Y^h(x)$, which are the input and output annotation function of $h$, respectively. $X^h(x)$ indicates the concatenation of all middle layer classifiers’ output. $Y^h(x)$ represents a 0-1 vector indicating whether the DNN or each correction model (i.e., each element in $\{f(x), f + f_1(x), \cdots, f + f_{n-1}(x), f \circ g_0(x), \cdots, f \circ g_{n-1}(x)\}$) is classified correctly or not. Algorithm [1] indicates the training of output decision model where $F(\cdot)$ is the ensemble model of all correction models and the

**Algorithm 1** (Output Decision model training): PGD adversarial training for $T$ epochs, given some radius set $\mathcal{E}$, adversarial step size $A$, $N$ PGD steps and a dataset of size $M$, robust output decision model $h$ and its weights $\theta_h$

\[
\mathcal{D} = \{x_i | i \leq M\} \\
\text{for } \epsilon, \alpha \in \mathcal{E}, A \text{ do} \\
\quad \text{//Perform PGD adversarial attack} \\
\quad \delta = 0 \text{ //or randomly initialized} \\
\quad \text{for } j = 1 \text{ to } N \text{ do} \\
\quad\quad \delta = \delta + \alpha \cdot \text{sign}(\nabla_{\delta} L_CE(F(x_i + \delta), y_i)) \\
\quad\quad \delta = \max(\min(\delta, \epsilon), -\epsilon) \\
\quad \text{end for} \\
\quad \mathcal{D} = \mathcal{D} \cup \{x_i + \delta\} \\
\text{end for} \\
\text{for } t = 1 \text{ to } T \text{ do} \\
\quad \text{// Update model weights} \\
\quad \theta_h = \theta_h - \nabla_{\theta_h} L_{MFL}(x) \\
\text{end for} \\
\]

DNN model. Because the accuracies of different correction models are different on adversarial samples, for the output decision model $h$, the amount of training data of different labels is unbalanced. Hence, the multi-label focal loss is used as the loss function, which is defined as:

\[
L_{MFL}(x) = \sum_i -Y^h(x)_i \cdot (1 - S(h(X^h(x)))_i)\gamma \cdot \log(S(h(X^h(x)))_i) \\
\]

where $S(\cdot)$ is the sigmoid function, $\gamma$ is an adjustable factor, $S(h(X^h(x)))_i$ and $Y^h(x)_i$ are the $i^{th}$ element in these vectors, respectively.

**E. FACM Model**

Besides the correction property introduced in Sections [III-B] and [III-C] the diversity property of FACM is introduced in Proposition [3].

**Proposition 3**: The correction model in FACM is diverse against adversarial samples. The greater the attack strength, the more obvious the diversity.

**Proof 3**: Due to the complementarity of FA module and CMPD module in FACM, we discuss the diversity of the correction models in FA and CMPD, respectively.

In FA, Figs. [5] [6] and [7] show that there are differences between the correction models against adversarial samples on the naturally trained DNN model and the adversarially trained DNN model for MNIST, CIFAR10/100 datasets, and the greater the attack strength, the more obvious the differences. Hence, the diversity property of FA is correct with high confidence.

In CMPD, Figs. [9] [10] [11] and [12] also prove the diversity property of CMPD with high confidence. Proof over.

Based on Proposition [3] due to the diversity of the correction models in FACM and the randomness of the output decision
module, FACM can significantly improve the robustness of DNN against white-box attacks with weak transferability and black-box attacks. Two reasons briefly explain why the combination of diversity and randomness can defend white-box attacks with weak transferability and black-box attacks. (i) Because the attacked DNN model (i.e., the DNN f or the ensemble correction models) is not equal to the random selected correction model (i.e., the prediction model), the white-box attack on FACM model is not the real white-box attack, but almost the black-box transfer attack. Therefore, the performance of white-box attacks on FACM is decreased, especially white-box attacks with weak transferability [17]. The diversity of the correction models in FACM further decreases the performance of white-box attacks with weak transferability. (ii) The existence of diversity and randomness will lead to inaccurate gradient estimation and model optimization of black-box attacks.

IV. EXPERIMENTS AND RESULTS

In this section, the experiment is conducted to validate the effectiveness of the proposed model (FACM and FA) on achieving the robust output of the DNN model.

A. Experimental Setting

Our experiments are conducted on benchmark adversarial learning datasets, including MNIST [18], CIFAR10 [19] and CIFAR100 [19] datasets. For MNIST dataset, the algorithms with the model architecture MNISTNet [20] are evaluated, where MNISTNet includes 4 convolutional layers and 3 fully connected layers. For both CIFAR10 and CIFAR100 datasets, the algorithms with the model architecture WRN-16-4 [21] are evaluated, where WRN-16-4 includes 4 basic blocks. The architecture of middle layer classifier is a fully connected layer, in which the input is the output of middle layer and the output is the probability vector of each category. The auto-encoder architecture of the CMPD module is similar across the datasets and consists of 3 convolutional layers in both the encoder and the decoder. As baselines, 7 adversarial training methods are chosen, i.e., Fast Adversarial Training (Fast-AT) [5], You Only Propagate Once (YOPO) [8], Adversarial Training with Hypersphere Embedding (ATHE) [4], Fair Robust Learning (FRL) [9], Friendly Adversarial Training (FAT) [6], TRADES [10] and Adversarial Training with Transferable Adversarial examples (ATTA) [7], and 2 channel-wise activation suppressing methods, i.e., Channel-wise Activation Suppressing (CAS) [11] and Channel-wise Importance-based Feature Selection (CIFS) [12]. Note that the middle layer can be a single layer or a network block.

In our implementation for FACM model, during the training process, we firstly train the DNN model using natural training or TRADES adversarial training, then the FA module and the CMPD module are trained by stochastic gradient descent (SGD) optimizer with learning rate as 0.0005 and 0.001 for 10 and 30 epochs, respectively. Finally, the output decision module is trained by the SGD optimizer with learning rate as 0.1 for 20 epochs. Note that, in order to avoid overfitting, the CMPD module for the MNISTNet is trained for 5 epochs. Multistep learning rate scheduler is used in every training phase in FACM, and we decay the learning rate by 0.1 at the 1/4 and 3/4 of the total epochs. During the evaluation phase, we report each trained model’s classification accuracy, model build time, attack time and inference time on clean samples, 4 white-box attacks with good transferability, (i.e., Fast Gradient Sign Method (FGSM) [22], Projected Gradient Descent (PGD) [23], Momentum Iterative FGSM (MIFGSM) [24], AutoAttack [25]), 3 white-box attacks with weak transferability (i.e., Carlini Wagner $L_\infty$ and $L_2$ (CW$_\infty$ and CW$_2$) [2] and DeepFool $L_2$ [3]) and 2 black-box attacks (Square [26] and NATTACK [27]). All test samples in MNIST, CIFAR10 and CIFAR100 are used to evaluate the accuracy of model with the white-box attacks. For black-box attacks, the first 1000 samples in the test samples are used for evaluation with the Square attack, and the first 200 samples for evaluation with the NATTACK attack. Note that the attack time and inference time are the total calculation time on the test set, not the average calculation time. The average or avg. in Tables I, II, III, IV, V and VI is the average classification accuracy on the white-box attacks with weak transferability and the black-box attacks.

The aforementioned white-box attacks and black-box attacks are implemented in [28]. The hyperparameters of each attack are introduced as follows. The attack strength $\epsilon$ is set to 8/255 on CIFAR10/100 and 0.3 on MNIST for FGSM, PGD, MIFGSM, AA, CW$_\infty$ and NATTACK, and 0.05 on CIFAR10/100 and 0.3 on MNIST for Square. The step size $\alpha$ is set to 0.8/255 on CIFAR10/100 and 0.03 on MNIST for PGD, and 2/255 on CIFAR10/100 and 0.1 on MNIST for MIFGSM. The number of steps is set to 20 on CIFAR10/100 and 40 on MNIST for PGD, 10 on CIFAR10/100 and 50 on MNIST for CW$_2$ and CW$_\infty$, 5 on all datasets for MIFGSM, 50 on all datasets for DeepFool $L_2$, 5000 on all datasets for Square and 500 on all datasets for NATTACK. The learning rate is set to 0.2 on CIFAR10/100 and 1.0 on MNIST for CW$_2$, and 0.01 on all datasets for CW$_\infty$. The overshoot is 0.02 on all datasets for DeepFool $L_2$. The population is 100 on all datasets for NATTACK. The box-constraint is 0.005 on CIFAR10/100 and 10 on MNIST for CW$_2$. In addition, the hyperparameters of all the baselines are set to the optimal parameters in their original papers.

To explain the diversity of the correction model in FACM against adversarial samples, a difference metric $\xi (\rho_1, \rho_2, S)$ between two correction models $\rho_1$ and $\rho_2$ on test set $S$ is introduced as follows:

$$\xi (\rho_1, \rho_2, S) = \frac{||v_2^S \cdot v_2^{S_1}||_1 - ||v_2^S \cdot \& v_2^{S_1}||_1}{||v_2^S||}$$

(9)

where $\rho_1, \rho_2$ are two correction models, $S$ is the test set, $v_2^S$ denotes 0-1 vector in which each element represents whether the correction model $\rho_i$ correctly predicts a test sample in $S$, $\&$ and $\|$ represent OR and AND, respectively, $||\cdot||_1$ denotes the l1 norm. Note that, the CMPD model should be combined with the DNN model $f$ for correction, i.e., $f \circ g_i$.

We consider two settings on the attacks: grey-box and white-box. The grey-box attacks of FACM (FACM-grey) have full knowledge of the DNN model but not the defense model
of the FA module, the CMPD module and output decision module. The white-box attacks of FACM (FACM-white) not only have the full knowledge of the DNN model, but also the defense model about FACM. All experiments are run on single machine with four GeForce RTX 2080tis using Pytorch.

B. Correction Effect of Middle Layers Features

To verify that the middle layers features can correct the output of the DNN $f$ when $f$ is attacked (i.e., FACM-grey), we compared the test accuracy of $f$ with the middle layer correction models $f + f_i$, on MNIST, CIFAR10 and CIFAR100 under FGSM with different attack strength $\epsilon$. As shown in Fig. 1, the blue curve represents the test accuracy of $f$ varies with the attack strength $\epsilon$ and the other curves represent the middle layer correction models $\{ f + f_i \}$. The results show that the test accuracies of all the middle layer correction models $\{ f + f_i \}$ are higher than the DNN model $f$ on all datasets with all attack strengths. In addition, Fig. 5 verifies the diversity property of FA when $f$ is attacked (i.e., FACM-grey). As shown in Fig. 5, the difference matrix heatmaps on different datasets with different attack strengths show that: as the attack strength gradually increases, the differences between the correction models become more and more significant.

Fig. 3 shows that the test accuracies of all the middle layer correction models $\{ f + f_i \}$ are higher than the DNN model $f$ on MNIST with various attack strengths, less or equal to the DNN model $f$ on CIFAR10 and CIFAR100 with any attack strength when FACM-white is attacked by FGSM. Therefore, the correction property of FA is effective on MNIST, but ineffective on CIFAR100 when FACM-white is attacked by FGSM. In addition, Fig. 7 verifies the diversity property of FA when FACM-white is attacked by FGSM. As shown in Fig. 7, the difference matrix heatmaps on different datasets under different attack strengths show that: as the attack strength gradually increases, the differences between the correction models become more and more significant.

To further verify the diversity property of FACM on the adversarially trained DNN model, Figs. 6 and 8 represent the difference matrix heatmaps of the middle layer correction models on the TRADES trained DNN when FACM-grey and FACM-white are attacked by FGSM, respectively. As the attack strength gradually increases, the differences between the middle layer correction models become larger but not significant because the size of the common correction samples set is large (the test accuracy of the middle layer correction models on the TRADES trained DNN is higher than the naturally trained DNN under the FGSM attack) so that the difference calculated by Eq. 9 has slight increased.

C. Comparison between CMPD and MPD

To verify the effectiveness of the CMPD correction models when the DNN model $f$ is attacked (i.e., FACM-grey), we
Fig. 5. The difference (which is calculated by Eq. 9) matrix heatmap between different middle layer classifiers \( \{f + f_i\} \) and the DNN model \( f \) on MNIST, different middle layer classifiers \( \{f + f_i\} \) and the DNN model \( f \) on MNIST, CIFAR10 and CIFAR100 when the naturally trained DNN model \( f \) (i.e., FACM-grey) is attacked by FGSM with different \( \epsilon \) (255 on CIFAR10/100). FACM-grey is attacked by FGSM with different \( \epsilon \) (255 on CIFAR10/100).

Fig. 6. The difference (which is calculated by Eq. 9) matrix heatmap between different middle layer classifiers \( \{f + f_i\} \) and the DNN model \( f \) on MNIST, CIFAR10 and CIFAR100 when the TRADES trained DNN model \( f \) (i.e., FACM-grey) is attacked by FGSM with different \( \epsilon \) (255 on CIFAR10/100).

Fig. 7. The difference (which is calculated by Eq. 9) matrix heatmap between different middle layer classifiers \( \{f + f_i\} \) and the DNN model \( f \) on MNIST, CIFAR10 and CIFAR100 when the ensemble model of the naturally trained DNN model \( f \) and the middle layer classifiers \( \{f + f_i\} \) and the CMPD models \( \{g_i\} \) (i.e., FACM-white) is attacked by FGSM with different \( \epsilon \) (255 on CIFAR10/100). CMPD models \( \{g_i\} \) (i.e., FACM-white) is attacked by FGSM with different \( \epsilon \) (255 on CIFAR10/100), is attacked by FGSM with different \( \epsilon \) (255 on CIFAR10/100).

compared the test accuracies of the CMPD correction models \( \{g_i\} \) with the MPD correction model \( g_0 \) on MNIST, CIFAR10 and CIFAR100 under FGSM with different attack strength \( \epsilon \). As shown in Fig. 2, the blue curve represents the test accuracy of \( g_0 \) and the other curves represent \( \{g_i\} \). Fig. 2(1),(3) show that the test accuracy of the CMPD correction model \( g_i \) using the outputs of the shallow middle layer classifiers (i.e., \( i \) is small) as a condition is higher than that of the MPD correction model \( g_0 \) on MNIST and CIFAR100 datasets. Fig. 2(2) shows that the CMPD correction model \( g_i \) using the output of the shallow middle layer classifiers as a condition has comparable performance to the MPD correction model \( g_0 \). In addition, Fig. 9 verifies the diversity property of the CMPD correction model when FACM-grey is attacked by FGSM. As shown in Fig. 9, the difference matrix heatmaps on different datasets under different attack strengths show that: as the attack strength gradually increases, the differences between the CMPD correction models become more and more significant.

When FACM-white is attacked by FGSM, Fig. 11(1),(3) show that the test accuracies of all CMPD correction models \( \{g_i\} \) are higher than that of the MPD correction model \( g_0 \) on MNIST and CIFAR100, and Fig. 11(2) shows that the majority of the CMPD correction models \( \{g_i\} \) have comparable performance to the MPD correction model \( g_0 \). Therefore, the correction property of CMPD is effective on MNIST and CIFAR100. In addition, Fig. 11 verifies the diversity property of the CMPD correction model when FACM-white is attacked by FGSM. As shown in Fig. 11, the difference
Fig. 10. The difference (which is calculated by Eq. 9) matrix heatmap between different CMPD correction models built in the TRADES trained DNN under the FGSM attack, so that the size of the common correction samples set is large and the difference calculated by Eq. 9 has slight increased.

D. Comparison between FACM and FA

Fig. 13 evaluate the effect of TRADES trained model with different $\epsilon$ on FA and FACM for CIFAR10 under different types of attacks, and Fig. 14 for MNIST. As shown in Figs. 13 and 14 when $\epsilon$ is equal to 0, the TRADES trained model (i.e., the naturally trained DNN model) with FACM is better than FA against all types of attacks. However, when $\epsilon$ is greater than 0, the TRADES trained model with FA is better than FACM for a majority of white-box attacks. As $\epsilon$ in TRADES increases, the test accuracy of FA gradually approaches FACM against CW$_{\infty}$ and all black-box attacks. In conclusion, combining the aforementioned statements, the
TABLE I
Evaluations (test accuracy(\%)) of the naturally trained DNN model with or without FACM on CIFAR10/100 and MNIST datasets. F1, P1, Sq, NA and DF2 represent FGSM ($\epsilon=2/255$ on CIFAR10/100 and $\epsilon=0.1$ on MNIST), PGD ($\epsilon, \alpha, \text{step number}=2/255, 0.2/255, 20$ on CIFAR10/100 and $\epsilon, \alpha, \text{step number}=0.1, 0.01, 40$ on MNIST), SQUARE, NATTACK and DeepFool L2, respectively. Note that the attack strength of SQUARE is 0.05 on CIFAR10/100 rather than 0.031.

| Method            | Clean | Good transferability | Weak transferability | Black-box attacks | Avg. |
|-------------------|-------|----------------------|----------------------|------------------|------|
|                   |       | F1       | P1    | PGD    | DF2 | CW2 | CW∞ | Sq | NA |
| CIFAR 10          |       |          |       |        |     |     |     |    |    |
| Natural           | 94.41 | 24.86    | 9.69  | 1.28   | 0   | 3.96| 9.59| 0  | 1.5| 3.01|
| FACM-white($\tau=3$) | 91.86 | 60.26    | 32.86 | 0.66   | 12.24| 17.71| 23.37| 0  | 1  | 40.78|
| FACM-grey($\tau=3$) | 92.09 | 49.61    | 48.43 | 25.93 | 10.67| 79.7 | 29.51| 51.87| 70.1 | 80.5 | 62.34|
| CIFAR 100         |       |          |       |        |     |     |     |    |    |
| Natural           | 75.84 | 11.08    | 5.3   | 0.6    | 0   | 5.99| 4.06| 0  | 0  | 2.01|
| FACM-white($\tau=3$) | 73.44 | 28.78    | 10.64 | 10.22  | 0.32| 40.92| 10.29| 20.73| 34.3 | 49.5 | 31.15|
| FACM-grey($\tau=3$) | 72.94 | 26.07    | 24.06 | 11.6  | 19.71| 55.62| 12.76| 35.71| 37.58 |     |      |
| MNIST             |       |          |       |        |     |     |     |    |    |
| FACM-white($\tau=1$) | 99.53 | 65.78    | 9.7   | 19.48  | 0   | 46.13| 2.7  | 0  | 32 | 16.17|
| FACM-grey($\tau=1$) | 98.68 | 82.91    | 22.45 | 59.49  | 8.08| 86.37| 47.62| 72.09| 61.8 | 99 | 60.52|
| Avg.              |       |          |       |        |     |     |     |    |    |

TABLE II
Evaluations (test accuracy(\%)) of the adversarially trained DNN model with or without FA on CIFAR10/100 and MNIST datasets. M1, AA, Sq, DF2 and NA represent MIFGSM, AUTOAttack, SQUARE, DeepFool L2 and NATTACK, respectively. Note that the attack strength of SQUARE is 0.05 on CIFAR10/100 rather than 0.031.

| Method            | Clean | Good transferability | Weak transferability | Black-box attacks | Avg. |
|-------------------|-------|----------------------|----------------------|------------------|------|
|                   |       | F1       | P1    | PGD    | DF2 | CW2 | CW∞ | Sq | NA |
| CIFAR 10          |       |          |       |        |     |     |     |    |    |
| TRADES [10]       | 80.98 | 55.8     | 51.83 | 53.76  | 47.2| 0.59| 25.2| 48.54| 34.4 | 69 | 35.55|
| FA-white($\tau=1$)+TRADES | 79.42 | 55.09    | 51.63 | 53.66  | 47.1| 33.66| 27.96| 61.78| 72.6 | 76 | 54.4 |
| FA-grey($\tau=1$)+TRADES | 79.47 | 55.59    | 51.86 | 53.99  | -   | 35.42| 27.69| 59.1 | 54.16 |     |      |
| CIFAR 100         |       |          |       |        |     |     |     |    |    |
| TRADES [10]       | 55.91 | 29.23    | 26.91 | 27.82  | 21.7| 0.36| 8.9  | 22.36| 13.3 | 39.5 | 16.88|
| FA-white($\tau=3$)+TRADES | 51.17 | 27.62    | 26.33 | 26.83  | 22.0| 25.46| 12.67| 36.53| 46.7 | 51 | 32.17|
| FA-grey($\tau=3$)+TRADES | 51.34 | 28.81    | 27.11 | 27.91  | -   | 32.99| 13.99| 38.17| 36.57 |     |      |
| MNIST             |       |          |       |        |     |     |     |    |    |
| FA-white($\tau=2$)+TRADES | 99.48 | 93.36    | 70.69 | 82.32  | 27.7| 4.43| 68.17| 96.19| 19.7 | 95.5 | 56.8 |
| FA-grey($\tau=2$)+TRADES | 98.62 | 91.09    | 81.93 | 83.68  | 52.9| 40.79| 96.37| 97.03| 93.4 | 97.5 | 85.02|
| Avg.              |       |          |       |        |     |     |     |    |    |

TABLE III
Evaluations (test accuracy(\%)) of different adversarial training methods on CIFAR10 using white-box attacks with weak transferability and black-box attacks. DF2 represents DeepFool L2.

| Method            | Clean | Good transferability | Weak transferability | Black-box attacks | Average |
|-------------------|-------|----------------------|----------------------|------------------|---------|
|                   |       | F1       | P1    | PGD    | DF2 | CW2 | CW∞ | SQUARE (\epsilon=0.031) | SQUARE (\epsilon=0.035) | NATTACK |
| CIFAR 10          |       |          |       |        |     |     |     |                           |                           |         |
| Fast-AT [5]       | 0.71  | 21.3     | 45.03 | 49.0   | 26.3| 68.5 | 35.14 |                             |                             |         |
| YOPO-5-3 [8]      | 2.13  | 11.73    | 33.59 | 38.9   | 17.7| 60  | 27.34 |                             |                             |         |
| ATHE [4]          | 0.42  | 24.13    | 48.13 | 52.6   | 33.8| 69.5 | 38.10 |                             |                             |         |
| FRL [9]           | 1.82  | 5.4      | 21.44 | 26.3   | 9.3 | 50.5 | 19.13 |                             |                             |         |
| FAT [6]           | 0.48  | 25.13    | 48.11 | 51.7   | 32  | 69  | 37.74 |                             |                             |         |
| ATTA [7]          | 0.58  | 21.78    | 44.97 | 46.8   | 30.9| 64  | 34.84 |                             |                             |         |
| FA-white($\tau=1$)+TRADES | 33.66 | 27.96    | 61.78 | 76.2   | 72.6| 76  | 58.03 |                             |                             |         |

E. Effect of FACM on the Naturally Trained DNN Model

To verify that FACM can effectively improve the robustness of naturally trained DNN model against various attacks, especially white-box attacks with weak transferability and black-box attacks, under the slight decrease of performance on clean samples, we compare the classification accuracy of the naturally trained DNN model with and without FACM on CIFAR10/100 and MNIST. As shown in Table I for the setting of FACM-white, FACM can improve the classification accuracy against white-box attacks with good transferability, and the smaller the attack strength $\epsilon$ in white-box attacks with good transferability is, the more obvious the increase of classification accuracy on each datasets. FACM can significantly improve the classification accuracy against white-box attacks with weak transferability and black-box attacks, specifically, the average improvement is 37.77\% on CIFAR10, 29.14\% on CIFAR100 and 44.35\% on MNIST. The classification accuracy of clean samples decreases by 1.56\%, 4.74\% and 0.86\% on CIFAR10/100 and MNIST, respectively. For the setting of FACM-grey, the classification accuracy against clean samples naturally trained DNN model with FACM is better than FA. On the contrary, the TRADES trained model with FA is better than FACM. Therefore, in the following experiments, FACM is used for the naturally trained DNN model and FA is used for the adversarially trained DNN model.
TABLE IV
Evaluations (test accuracy(\%)) of different adversarial training methods on CIFAR100 using white-box attacks with weak transferability and black-box attacks. DF$_2$ represents DeepFool L$_2$.

| Method       | DF$_2$ | CW$_2$ | CW$_\infty$ | White-box attacks with weak transferability | Black-box attacks | Average |
|--------------|--------|--------|-------------|---------------------------------------------|-------------------|---------|
|              |        |        |             | Square (\(\varepsilon=0.03\)) | Square (\(\varepsilon=0.05\)) | NAI TACK |
| Fast-AT [5]  | 2.49   | 17.78  | 0           | 0                                           | 0                 | 0       |
| YOPO-5-3 [8] | 0.82   | 6.86   | 20.83       | 23.5                                        | 11                | 36      |
| ATHE [4]     | 0.43   | 10.99  | 24.82       | 26.8                                        | 14.8              | 38.5    |
| FRL [9]      | 1.93   | 2.67   | 6.94        | 8.1                                         | 2.8               | 19      |
| FAT [6]      | 0.57   | 9.03   | 22.43       | 23.6                                        | 13.8              | 36      |
| ATTA [7]     | 0.63   | 8.17   | 13.98       | 13.9                                        | 9.7               | 20      |
| FA-white(\(\tau=3\)) + TRADES | 25.46  | 12.67  | 36.53       | 50.6                                        | 46.7              | 51      |

TABLE V
Evaluations (test accuracy(\%)) of different adversarial training methods on MNIST using white-box attacks with weak transferability and black-box attacks. DF$_2$ represents DeepFool L$_2$.

| Method       | DF$_2$ | CW$_2$ | CW$_\infty$ | White-box attacks with weak transferability | Black-box attacks | Average |
|--------------|--------|--------|-------------|---------------------------------------------|-------------------|---------|
|              |        |        |             | Square (\(\varepsilon=0.3\)) | Square (\(\varepsilon=0.4\)) | NAI TACK |
| Fast-AT [5]  | 67.16  | 51.21  | 90.61       | 70.3                                        | 0                 | 95.5    |
| YOPO-5-10 [8]| 33.82  | 46.36  | 85.66       | 69.6                                        | 5.3               | 90      |
| ATHE [4]     | 1.77   | 95.23  | 95.76       | 86.2                                        | 0                 | 97      |
| FRL [9]      | 16.8   | 73.11  | 93.8        | 67.2                                        | 0                 | 94.5    |
| FAT [6]      | 0.79   | 76.06  | 95.67       | 59.3                                        | 0                 | 96      |
| ATTA [7]     | 2.5    | 89.27  | 96.81       | 92.4                                        | 0                 | 97      |
| FA-white(\(\tau=2\)) + TRADES | 40.79  | 96.37  | 97.03       | 93.4                                        | 27.9              | 97.5    |

TABLE VI
Evaluations (test accuracy(\%)) of different channel-wise activation suppressing methods on CIFAR10 using various attacks. CL, F, P, MI, AA, DF$_2$, SQ and NA represent CLEAN, FGSM, PGD, MIFGSM, AUTOAttack, DeepFool L$_2$, SQUARE and NATACK.

| Method       | Cl     | F     | P     | MI    | AA    | DF$_2$ | CW$_2$ | CW$_\infty$ | White-box attacks with good transferability | Black-box attacks | Avg. |
|--------------|--------|-------|-------|-------|-------|--------|--------|-------------|---------------------------------------------|-------------------|------|
|              |        |       |       |       |       |        |        |             | Square (\(\varepsilon=0\))                      | Square (\(\varepsilon=0\))                      |       |
| WRN-16-4 CAS+TRADES [11] | 80.92  | 51.36 | 46.92 | 49.22 | 42.7  | 0.48   | 20.4   | 44.44       | 65.2                                        | 80     | 42.10 |
| ResNet-20 CIFS [12] | 82.46  | 61.07 | 54.66 | 58.02 | -     | 0.66   | 37.99  | 53.74       | 39.8                                        | 66.5   | 39.74 |
| 18 Our FA-white(\(\tau=3\)) + CIFS | 82.20  | 59.48 | 53.98 | 56.51 | -     | 9.59   | 37.43  | 63.48       | 73.4                                        | 78     | 52.38 |

TABLE VII
The build time (MINS) of different methods for building robust models on MNIST and CIFAR10/100.

| Method       | Natural Train | Fast-AT | YOPO | ATHE | FRL | FAT | ATTA | TRADES | Our FACM | Our FACM+TRADES |
|--------------|---------------|---------|------|------|-----|-----|------|--------|----------|-----------------|
|              | MNIST         | 8.5     | 13.6 | 43.3 | 122.6 | 39.9 | 161.8 | 63.6   | 93.1     | 163.2           |
|              | CIFAR10       | 50.7    | 61.4 | 134.8 | 601.5 | 708.1 | 507.9 | 183.6  | 280.7    | 236.6           |
|              | CIFAR100      | 114.8   | 45.6 | 66.3 | 558.2 | 1485  | 1015  | 165.2  | 498.5     | 235.8           |

and various attacks is higher than that of FACM-white.

F. Effect of FA on the Adversarially Trained DNN Model

To verify that FA can effectively improve the robustness of adversarially trained DNN model against white-box attacks with weak transferability and black-box attacks, under the usually slight decrease of performance against clean samples and white-box attacks with good transferability, we compare the classification accuracy of the TRADES trained model with and without FA on CIFAR10/100 and MNIST. As shown in Table [11] for the setting of FA-white, FA can significantly improve the classification accuracy against white-box attacks with weak transferability and black-box attacks, specifically, the average improvement is 18.85% on CIFAR10, 15.29% on CIFAR100 and 28.22% on MNIST. The average classification accuracy against clean samples and white-box attacks with good transferability decreases by 0.53%, 1.52% on CIFAR10/100, respectively, increases by 6.93% on MNIST. For the setting of FA-grey, the classification accuracy against clean samples and various attacks is higher than that of FA-white.

G. Comparison between FA and the Other Adversarial Training Methods

To verify that TRADES trained model with FA is more robust than the other adversarial training methods against white-box attacks with weak transferability and black-box attacks,
TABLE VIII
THE ATTACK TIME (MINS) COMPARISON BETWEEN NATURALLY TRAINED DNN MODEL WITH AND WITHOUT FACM ON MNIST, CIFAR10/100. THE LONGER THE ATTACK TIME, THE GREATER THE AMOUNT OF COMPUTATION, AND THE MORE DIFFICULT THE MODEL IS TO BE ATTACKED.

| Method | CIFAR10 | CIFAR100 | MNIST |
|--------|---------|----------|-------|
| Natural Train | 0.03 | 0.07 | 0.17 |
| Our FACM | 0.21 | 0.35 | 0.29 |

Table IX
THE INFERENCE TIME (MINS) COMPARISON BETWEEN THE MODEL WITH AND WITHOUT FACM OR FA ON MNIST, CIFAR10 AND CIFAR100.

| Method | CIFAR10 | CIFAR100 | MNIST |
|--------|---------|----------|-------|
| TRADES | 0.051 | 0.050 | 0.028 |
| FA(τ=2)+TRADES | 0.066 | 0.068 | 0.036 |
| FACM(τ=3)+TRADES | 0.170 | 0.156 | 0.052 |

we compare the classification accuracy of the TRADES trained model with FA and the other six adversarial training methods on CIFAR10/100 and MNIST. As shown in Tables III and IV for the setting of FA-white, the classification accuracy of TRADES with FA is higher than the other six adversarial training methods against the majority white-box attacks with weak transferability and black-box attacks. In comparison with the best one in the other six adversarial training methods, the average improvement is 19.93%, 17.77% and 12.50% on CIFAR10/100 and MNIST, respectively. In addition, for Square, the increase of the attack strength has little impact on the performance of TRADES with FA and has a great impact on the other six adversarial training methods without FA.

H. Comparison between FA and the Channel-wise Activation Suppressing Methods

To further verify that the adversarially trained DNN model with FA is more robust than the channel-wise activation suppressing methods against white-box attacks with weak transferability and black-box attacks, we compare the classification accuracy of TRADES with CAS and TRADES with FA, and CIFS with and without FA on CIFAR10. As shown in Table VI in comparison with CAS, the classification accuracy of TRADES with FA is higher than that of TRADES with CAS against various attacks except NATTACK under the slight decrease of performance on clean samples. The average classification accuracy against white-box attacks with weak transferability and black-box attacks increases by 12.3%. In comparison with CIFS, under the slight decrease of performance against clean samples and white-box attacks with good transferability, FA can significantly improve the classification accuracy against white-box attacks with weak transferability and black-box attacks, specifically, the average reduction is slight, i.e. 1.01%, and the average improvement is significant, i.e. 12.64%.

I. The Sensitivity Analysis of τ

Fig 15 investigates the influence of the size of the parameter τ on the performance of the FACM-white on MNIST under different types of attacks. In the subgraph (1), the DNN is obtained through natural training; in the subgraph (2), the DNN is obtained through TRADES adversarial training.

J. Build Time, Attack Time and Inference Time Comparisons

As shown in Table VII the build time of the naturally trained DNN model with FACM and the TRADES trained...
ADVERSARIAL TRAINING

Adversarial Training: Besides several variants of AT have been introduced in Section I, Sriramanan et al. introduced a relaxation term to the the standard loss, that finds more suitable gradient-directions, increase attack efficacy and leads to more efficient adversarial training. Wang et al. proposed Once-for-all adversarial training methods with a controlling hyper-parameter as the input where trained model could be adjusted among different standard and robust accuracies at testing time. Stutz et al. tackled the problem of the robustness generalization on the unseen threat model by biasing the model towards low confidence predictions on adversarial examples. Laidlaw et al. developed perceptual adversarial training against a perceptual attack gives robustness against many other types of adversarial attacks. Pang et al. provide comprehensive evaluations on CIFAR10, focusing on the effects of mostly overlooked training tricks and hyperparameters for adversarially trained DNN models.

Robust Architectures: To obtain robust network architectures, Du et al. attempts to trade off exploring diverse structures and exploiting the best structures, and a new stochastic neural networks, an intrinsically sparse rewiring approach, implicit Euler skip connections. LayerCert framework, the deep Bayes classifier are proposed to enhance robustness.

Randomness: Randomness is an effective approach to defense black-box attacks. Dhillon et al. proposed stochastic activation pruning, a mixed strategy for adversarial defense. Xie et al. proposed to utilize randomization at inference time to mitigate adversarial effects.

VI. CONCLUSION

In this paper, the middle Feature layer Analysis and Conditional Matching prediction distribution (FACM) model is proposed to improve the model robustness through correcting the output of the DNN with features extracted by middle layers of DNN. Specifically, we theoretically demonstrate the advantages of FACM in terms of the correction and diversity properties, and empirically justify our FACM on benchmark datasets. The experimental results show that our FACM can effectively improve the robustness of the naturally trained DNN model against various adversarial attacks, especially black-box attacks and white-box attacks with weak transferability. In addition, our FA module can improve the accuracy of the adversarially trained DNN model against both black-box attacks and white-box attacks with weak transferability.

REFERENCES

[1] Y. Ding, L. Wang, H. Zhang, J. Yi, D. Fan, and B. Gong, “Defending against adversarial attacks using random forest,” in CVPR Workshops, 2019, pp. 105–114.
[2] N. Carlini and D. A. Wagner, “Towards evaluating the robustness of neural networks,” in IEEE Symposium on Security and Privacy, 2017, pp. 39–57.
[3] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: A simple and accurate method to fool deep neural networks,” in CVPR, 2016, pp. 2574–2582.
[4] T. Pang, X. Yang, Y. Dong, T. Xu, J. Zhu, and H. Su, “Boosting adversarial training with hypersphere embedding,” in NeurIPS, 2020.
[5] E. Wong, L. Rice, and J. Z. Kolter, “Fast is better than free: Revisiting adversarial training,” in ICLR, 2020.
[6] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. S. Kankanahalli, “Attacks which do not kill training make adversarial learning stronger,” in ICML, 2020, pp. 11 278–11 287.
[7] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash, “Efficient adversarial training with transferable adversarial examples,” in CVPR, 2020, pp. 11 178–11 187.
[8] D. Zhang, T. Zhang, Y. Lu, Z. Zha, and B. Dong, “You only propagate once: Accelerating adversarial training via maximal principle,” in NeurIPS, 2019, pp. 227–238.
[9] H. Xu, X. Liu, Y. Li, A. K. Jain, and J. Tang, “To be robust or to be fair: Towards fairness in adversarial training,” in ICLR, 2021, pp. 11 492–11 501.
[10] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. El Ghaoui, and M. I. Jordan, “Theoretically principled trade-off between robustness and accuracy,” in ICML, 2019, pp. 7472–7482.
[11] Y. Bai, Y. Zeng, Y. Jiang, S. Xia, X. Ma, and Y. Wang, “Improving adversarial robustness via channel-wise activation suppressing,” in ICLR (Poster), 2021.
[12] H. Yan, J. Zhang, G. Niu, J. Feng, V. Y. F. Tan, and M. Sugiyama, “CIFS: improving adversarial robustness of cnns via channel-wise importance-based feature selection,” in ICML, 2021, pp. 11 693–11 703.
[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in ICLR (Poster), 2018.
[14] G. Vacanti and A. V. Looveren, “Adversarial detection and correction by matching prediction distributions,” CoRR, vol. abs/2002.09364, 2020.
[15] Z.-H. Zhou, “Computational learning theory,” in Machine Learning, 2021, pp. 287–313.
[16] S. Ma, Y. Liu, G. Tao, W. Lee, and X. Zhang, “NIC: detecting adversarial samples with neural network invariant checking,” in NDSS, 2019.
[17] W. Wu, Y. Su, X. Chen, S. Zhao, I. King, M. R. Lyu, and Y. Tai, “Boosting the transferability of adversarial samples via attention,” in CVPR, 2020, pp. 11 586–11 596.
[18] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist, vol. 2, 2010.
[19] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.
[20] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in IEEE Symposium on Security and Privacy, 2016, pp. 582–597.
[21] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in CVPR, 2016.
[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in ICLR (Poster), 2015.
[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in ICLR (Poster), 2018.
