Vanilla Feature Distillation for Improving the Accuracy-Robustness Trade-Off in Adversarial Training

Guodong Cao, Zhibo Wang, Senior Member, IEEE, Xiaowei Dong, Zhifei Zhang, Hengchang Guo, Zhan Qin, Senior Member, IEEE, and Kui Ren, Fellow, IEEE

Abstract—Adversarial training has been widely explored for mitigating attacks against deep models. However, a critical limitation of existing works is that robustness enhancement is at the cost of noticeable accuracy degradation. To achieve a better trade-off between robustness and accuracy, we propose the Vanilla Feature Distillation Adversarial Training (VFDAT), which conducts knowledge distillation from a pre-trained model (optimized towards high accuracy) to guide adversarial training model towards generating high-quality and well-separable features by constraining the obtained features of natural and adversarial examples. More specifically, both adversarial examples and their natural counterparts are forced to be aligned in feature space by distilling predictive representations from a pre-trained natural model. In this way, the adversarial training model can be updated towards maximally preserving the accuracy as gaining robustness. A key advantage of our method is that it can be universally adapted to and boost existing works. Exhaustive experiments on various datasets, classification models, and adversarial training algorithms demonstrate the effectiveness of our proposed method.

Index Terms—Adversarial training, well-separable features, feature alignment, knowledge distillation.

I. INTRODUCTION

Deep neural networks (DNNs) have widely deployed in various of daily tasks, e.g., image classification [1], [2], object detection [3], [4], autonomous driving [5], etc. However, DNNs are known to be vulnerable to adversarial examples generated by overlaying carefully designed perturbation into original/natural examples [6], [7], [8]. Against those adversarial examples, the adversarial training is explored to improve the robustness of DNNs, typically by feeding adversarial examples to the model during the training stage [9], [10], [11]. Adversarial training can be formulated as a min-max optimization problem, where perturbation is generated to maximize the original loss, and then the model is optimized against the perturbation/attacks by minimizing the loss [12]. Its goal is to make the model robust and prompt the model to correctly classify the input samples, even with the adversarial perturbation.

There are a large literature devoted to improving the model robustness by adversarial training. They can be roughly divided into two categories: 1) data augmentation by increasing the diversity of training data with adversarial examples [13], and 2) adversarial regularization by constraining adversarial training with well-designed objective functions to enhance robustness [6], [7], [9], [14], [15]. These methods have successfully improved the robustness of the models at the cost of large accuracy drops during adversarial training, because the trained models pay too much attention to the robust feature information of the sample, while ignoring other feature information [16]. Recent studies [14], [17], [18] found that knowledge distillation can improve the accuracy and robustness in adversarial training. These methods only guide the student model to learn the decision boundary of the adversarial examples by resorting to third-party models, while ignoring constraints on natural examples. The common disadvantage of the above methods is that the learned features are mixed in the feature space, which could not achieve a better trade-off between accuracy and robustness.

To solve the above problem, a basic idea is that if the features of adversarial examples and their natural counterparts are clustering in the same class and separable from other classes during adversarial training, the trade-off between robustness and accuracy could be reconcilable. The typical adversarial training and feature-alignment adversarial training methods can improve the robustness of the model, but the features they learn are still mixed in the feature space as shown in Fig. 1(b) and (c). We want to constrain the features of natural and adversarial examples and make them well-separable, so the features have large inter-class distance and less intra-class distance, as shown in Fig. 1(d). One intuition is that we can utilize the high-quality features, e.g., dots in Fig. 1(a) generated by a natural training model, through
knowledge distillation, in which the features of natural examples are aggregated in the same class and well-separable from the different classes in the natural training model.

Based on the above idea, this paper proposes Vanilla Feature Distillation Adversarial Training (VFDAT) to achieve a better trade-off between accuracy and robustness by distilling vanilla representations from a high-accuracy pre-trained model. More specifically, a vanilla model is trained only with natural examples and has high accuracy on natural dataset. It learns the well-separable features, that have large inter-class distance and less intra-class distance, from the natural samples, which is called vanilla features. We take advantage of the vanilla/well-separable features and propose a new training strategy VFDAT that forces those natural features and adversarial features close to the corresponding vanilla/well-separable features. In this way, the extracted features are clustering and well-separable, thus the adversarial training model can achieve a better trade-off between accuracy and robustness. Please note that the vanilla model is only trained with natural examples, while the adversarial training model is fed by both natural and adversarial examples. Exhaustive experiments on various datasets, classification models, and adversarial training methods demonstrate the effectiveness of our proposed method. In summary, our main contributions are four-fold:

- To the best of our knowledge, we are first to define vanilla/well-separable features that have large inter-class distance and less intra-class distance in the feature space, which could be used to guide the adversarial training model to achieve a more favorable balance between model robustness and accuracy.
- We propose Vanilla Feature Distillation Adversarial Training to improve the accuracy and robustness of the adversarial training model by distilling vanilla features. Specifically, we not only force the adversarial features to be close to the corresponding well-separable features but also force the natural features close to the well-separable features.
- The proposed method could be a universally adaptable plug-in for existing related methods to boost accuracy and robustness.
- Extensive experiments on diverse classification models, datasets, and adversarial training methods demonstrate the superior performance of our method in terms of accuracy and robustness.

II. RELATED WORKS

In this section, we introduce related work about adversarial attacks and adversarial training methods to explain the attacks and defenses against attacks in AI models.

A. Adversarial Attack

Since Adversarial attacks were first introduced in [7], many attack algorithms have been proposed to analyze the vulnerability of deep neural networks. This line of research could be roughly divided into two categories according to the knowledge of the adversary, i.e., white-box attacks and black-box attacks. Some studies focus on white-box attacks, where the adversary has full access to the parameters of the target model. With the full knowledge of the target model, attacks can be conducted by Fast Gradient Sign Method (FGSM) [7], Momentum Iterative Method (MIM) [19], Projected Gradient Descent (PGD) [6], and many other methods [8], [20]. Besides, several works focus on black-box attacks, in which the adversary cannot access the target model [21]. In this case, the adversary carries out the attack with the transferability of adversarial examples, i.e., adversarial examples crafted from one model can effectively attack the other model. Without knowing the parameters of the target model, Carlini Wagner Attack (C&W) [22], GAP
B. Adversarial Training

In order to defend against adversarial attacks, adversarial training was proposed to improve the robustness of a model by augmenting the training dataset with adversarial examples when training. Ilyas et al. [16] reported that adversarial examples were features, rather than defects. They posited that the decision of the model relies on the “non-robust” features, which were vulnerable to adversarial perturbation. The decision of the robust model was more dependent on the robust features, which could be learned correctly from adversarial examples. The robust model may be ignored on non-robust features, which results in decreased accuracy. Such that, there is a trade-off between accuracy and robustness for the adversarial training model. Zhang et al. [12] theoretically proved a trade-off is label space. At layer \( l \) and \( L \) are metric criterion, is input space and \( y \) can be denoted as \( y \) is the ground truth label and \( y^* \) is the ground truth of \( F \).

C. Knowledge Distillation

Knowledge distillation is a technique and used in machine learning for improving model performance. Its goal is to transfer knowledge from a complex model (teacher) to a simple model (student) so that the simple model can achieve comparable performance to the complex model [29]. Recently, some researchers found that adversarial training can be improved with knowledge distillation, e.g., Helper-based adversarial training (HAT) [14]. The teacher model generated the mutual relations within the output embeddings as the ‘soft’ knowledge to finetune the student model [30]. In [18] used the logistics of the teacher model output to guide the adversarial training model learning a reasonable decision boundary. In [17] used smoothed labels from Knowledge Distillation (KD) to calibrate the notorious overconfidence of logits generated by pre-trained models. The common disadvantage of the above methods is that the learned features are mixed in the feature space, which could not achieve a better trade-off between accuracy and robustness.

By contrast, we propose a Vanilla Feature Distillation Adversarial Training (VFDAT) that transfers the high-accuracy knowledge from the vanilla model to the adversarial training model by approximating the extracted features of natural and adversarial examples to the vanilla features. In this way, the adversarial training model learns to extract predictive/well-separable features from natural and adversarial examples and achieves a better trade-off between accuracy and robustness.

III. PRELIMINARY

In this section, we will briefly introduce deep neural networks, adversarial attacks, adversarial training, and knowledge distillation to better understand our proposed method.

Deep Neural Networks: In this paper, we focus on deep neural network based images classification models. A Deep Neural Network \( F \) with parameters \( \theta \) can be denoted as \( F(x; \theta) : \mathcal{X} \rightarrow \mathcal{Y} \), where \( \mathcal{X} \) is input space and \( \mathcal{Y} \) is label space. At layer \( l \in \{1, 2, \ldots, L\} \), \( L \) is the number of layers, we denote the output at layer \( l \) as \( F^l(x; \theta) \). Usually, the training process of a neural network is to minimize loss function \( L = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(F(x_i; \theta), y_i) \) on training data \( D \) where \( N \) is the number of training instances and \( y_i \) is the ground truth of \( x_i \). \( \mathcal{L}(\cdot, \cdot) \) is usually the cross-entropy for a classification model.

Adversarial Attack: Recent studies show that deep learning models are vulnerable to adversarial examples [31]. In this paper, we focus on untargeted adversarial attacks. Given a classification model \( F(x; \theta) \), the goal of untargeted adversarial attack is to find a small perturbation to generate an adversarial example \( x^{adv} \), to mislead the classifier \( F(x^{adv}; \theta) \neq y \). Typically, the \( \ell_p \)-norm of the perturbation should be less than \( \epsilon \), i.e. \( \|x^{adv} - x\|_p \leq \epsilon \).

Adversarial Training: Adversarial training is a method for defending against adversarial attacks by augmenting the training dataset with adversarial examples. Adversarial training can be formulated as a min-max optimization problem as:

\[
\arg\min_{\theta} \max_{x^{adv} \sim D} \mathbb{E}_{(x, y) \sim D} \left[ \mathcal{L}(F(x^{adv}; \theta), y) \right], \quad \text{s.t.} \|x^{adv} - x\|_p \leq \epsilon
\]

where \( x^{adv} \) is the adversarial example that can maximize loss within \( \ell_p \)-norm distance \( \epsilon \), \( \theta \) is parameters of model \( F \) that needs to be updated to minimize the loss.

Knowledge Distillation: Knowledge distillation is a popular and successful model compression technique that can transfer knowledge from a large pre-trained teacher model to a smaller student model. Given a teacher model \( F_t \) with parameters \( \theta_t \) and a student model \( F_s \) with parameters \( \theta_s \), knowledge distillation can be formulated as minimizing a combined loss of soft and hard labels by updating \( \theta_s \):

\[
\mathcal{L}_{KD} = \rho \mathcal{L}_{soft}(z_s, z_t) + (1 - \rho) \mathcal{L}_{hard}(z_s, y),
\]

where \( \mathcal{L}_{soft} \) and \( \mathcal{L}_{hard} \) are metric criterion, \( \mathcal{L}_{soft} \) is Kullback-Leibler divergence [32], \( \mathcal{L}_{hard} \) is cross-entropy, \( z_s \) and \( z_t \) are logits of \( F_s \) and \( F_t \), respectively, \( y \) is the ground truth label and \( \rho \) is a hyper-parameter to control the ratio between \( \mathcal{L}_{soft} \) and \( \mathcal{L}_{hard} \). After distillation, the student network can achieve better performance than training only with the dataset.
IV. VANILLA FEATURE DISTILLATION ADVERSARIAL TRAINING

In this paper, we propose Vanilla Feature Distillation Adversarial Training (VFDAT) to realize a better trade-off between robustness and accuracy. This section will first overview the proposed VFDAT and then describe the detailed design of network and loss functions. Finally, we will further discuss the training strategy of VFDAT.

A. Overview of VFDAT

Existing adversarial training methods (e.g., TRADE, MART) incorporate adversarial examples into the training dataset and supervise the model training with true ground truth or soft labels (e.g., logits from a teacher model). They have improved model robustness at the cost of large accuracy drops during adversarial training. Previously, few researchers pay attention to the variation of sample feature space in adversarial training, especially the features of natural samples, resulting in large accuracy degradation on natural examples. If a model can extract identical features for both adversarial examples and their natural counterparts, both of them are predictive/well-separable, i.e., that have large inter-class distance and less intra-class distance, thus mitigating the trade-off between robustness and accuracy.

In this paper, we propose a Vanilla Feature Distillation Adversarial Training (VFDAT) to alleviate the dilemma between gaining robustness and preserving high accuracy for existing adversarial training methods. It could guide the adversarial training model to learn the vanilla/well-separable features from the natural and adversarial examples. When feeding adversarial examples to the model during the adversarial training stage, the different categories features are mixed, as shown in the second to fifth columns of Fig. 4. Although the adversarial training improves model robustness, the accuracy of the model drops.

In this paper, we improve the accuracy and robustness of the adversarial training model and aim to learn the well-separable features from natural and corresponding adversarial examples. To achieve our goal, we are facing two nontrivial challenges. First, since the adversarial examples are generated based on the opposite gradient of the natural samples, the features of the natural samples are far from the features of the corresponding adversarial examples. Therefore, it is difficult for the adversarial training model to generate well-separable features, that have large inter-class distance and less intra-class distance in the feature space. Second, the existing adversarial training clusters the features of natural samples and adversarial examples, but the features of different categories are mixed in the feature space. How to take advantage of the well-separable features to separate different category features is a challenge.

To solve these challenges, we propose the vanilla feature distillation adversarial training method to achieve a better trade-off between accuracy and robustness for the adversarial training model by learning the well-separable features from natural and corresponding adversarial examples. Specifically, to generate the well-separable features, we train a high-accuracy vanilla model only with natural examples. It has the same architecture as the adversarial training model, which promises that the feature size is identical in the same convolutional layer. Importantly, it learns the well-separable features from the natural samples, that have large inter-class distance and less intra-class distance. Then, to take advantage of the well-separable features, we propose the vanilla feature distillation adversarial training method to constrain model adversarial training, which forces the natural and adversarial features close to the corresponding well-separable features. Specifically, we minimize the feature vector distance between the well-separable feature and the corresponding natural and adversarial example feature.

The pipeline of proposed VFDAT is overviewed in Fig. 2, where VFDAT has two components: 1) the high accuracy vanilla model $F_{van}$ trained only with merely natural examples, and 2) the adversarial training model $F_{adv}$ that needs to be updated.
(a) Accuracy and robustness trade-off of different adversarial training methods on ResNest18. We apply the PGD ($\text{step} = 20, \ell_\infty, \varepsilon = 8/255$) to generate adversarial examples in white-box attack. Our approach can achieve the best trade-off between accuracy and robustness compared to original TRADES and optimized TRADES, e.g., TRADES accompanied by KD, TRADES accompanied by GBAT, TRADES accompanied by LGBAT. (b) Robust accuracy with different attack radius ($\varepsilon = 0/255 \sim 10/255$) on ResNest18. Our approach outperforms the original TRADES, TRADES accompanied by KD, TRADES accompanied by GBAT and TRADES accompanied by LGBAT.

Fig. 4. Feature visualization of natural examples and adversarial examples in CIFAR-10 on ResNet-18 and VGG-16. Each color represents a class. Dots and triangles represent natural and adversarial examples respectively. The first and third rows are the feature visualization of natural examples on ResNet-18 and VGG-16. The second and fourth rows are the feature visualization of adversarial examples on ResNet-18 and VGG-16.
in VFDAT. A vanilla model can be one of those off-the-shelf models or pre-trained by ourselves. And the adversarial training model is the model we want after training of VFDAT. Please note that the vanilla model and the adversarial training model have the same architecture, while the parameters of the vanilla model should be frozen once pre-trained. Sharing the spirit of general knowledge distillation and adversarial training during the training process. The vanilla model provides vanilla/well-separable features by extracting from natural examples, while the adversarial training model extracts features from both natural examples and adversarial examples. By matching the extracted features to vanilla features, the adversarial training model learns the features from natural and adversarial examples, that are close to vanilla/high-accuracy features in adversarial training.

B. Loss Functions

In this part, we detail the loss functions of the above mentioned VFDAT. As illustrated in the overview, there are three loss functions: 1) the natural loss $L_{\text{natural}}$ for supervision of natural examples, and 2) the robustness loss for adversarial training $L_{\text{adv}}$, we design 3) the knowledge distillation loss of the feature $L_{\text{KD}}$ for guiding adversarial training model towards a better trade-off between accuracy and robustness.

1) The Natural Loss $L_{\text{natural}}$. The natural loss measures the difference between the model prediction and the real label, which aims to enable the model to better fit natural samples. Since we aim to get a model $F_{\text{adv}}$ with high accuracy and robustness, we take natural examples and their labels for one of the supervisions as general adversarial training. This loss can be expressed as:

$$L_{\text{natural}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}(F\_{\text{adv}}(x_i; \theta_{\text{adv}}), y_i),$$

where $x_i$ denotes a natural example, $y_i$ is ground truth for $x_i$, $\mathcal{J}(\cdot, \cdot)$ is cross-entropy for multi-label classification.

2) The Robustness Loss $L_{\text{adv}}$. Meanwhile, we need to improve the robustness of model. A typical way to improve robustness is adversarial training, i.e., incorporating adversarial examples into the model training process. The robustness loss is a function of the existing adversarial training loss, such as TRADES, ALP, etc., which aims to correctly classify the adversarial examples as much as possible by approximating natural samples output. The loss for adversarial training can be expressed as:

$$L_{\text{adv}} = \frac{1}{N} \sum_{i=1}^{N} \arg \max_{x_i^{\text{adv}}} \phi(F\_{\text{adv}}(x_i^{\text{adv}}; \theta_{\text{adv}}), \hat{y}_i),$$

s.t. $\|x_i^{\text{adv}} - x_i\|_p \leq \epsilon,$

Where $\hat{y}_i$ can be either soft labels (e.g., logits) or hard labels (ground truth), $\phi(\cdot, \cdot)$ is a distance metric (e.g., Kullback-Leibler Divergence, Cross-Entropy). $L_{\text{adv}}$ itself tends to force the adversarial training model to learn similar outputs from the natural and corresponding adversarial example.

3) The Knowledge Distillation Loss $L_{\text{KD}}$: Model predictions depend on the extracted features of input samples [33]. Adversarial perturbation disrupts the feature extraction process and mislead the model to map adversarial examples to a false class. Even if adversarial examples can be recognized correctly after adversarial training, compared with the vanilla model, the features that the model extracts from the natural examples are more mixed in feature space, which results in decreasing the accuracy of the model. To solve this problem, we design a knowledge distillation loss $L_{\text{KD}}$ to extract predictive/well-separable features from adversarial examples and their natural counterparts in adversarial training.

More specifically, the design of $L_{\text{KD}}$ begins with knowledge distillation. Knowledge distillation is one of the most popular techniques used to transfer knowledge from one model to another. In particular, when the architectures are identical, this is called self-distillation. The self-distilled model achieves higher accuracy on held-out data [34], [35]. To get predictive/well-separable features, we pre-trained a vanilla model that is optimized and towards high accuracy only with natural examples. Then, we distill knowledge from its intermediate layer. We call this knowledge vanilla features (i.e., output at layer $k$ of the model for natural examples in our setting). We constrain features from the adversarial training model that takes adversarial examples and natural examples as input and make them similar to those from the vanilla model fed with the same natural examples. In this way, the model learns to map them close to vanilla features. The design of $L_{\text{KD}}$ is defined as:

$$L_{\text{KD}} = \frac{1}{N} \sum_{i=1}^{N} \left( \Gamma(F\_{\text{adv}}(x_i; \theta_{\text{adv}}), F\_{\text{van}}(x_i; \theta_{\text{van}})) + \Gamma(F\_{\text{adv}}(x_i; \theta_{\text{adv}}), F\_{\text{van}}(x_i; \theta_{\text{van}})) \right),$$

Where $\Gamma(\cdot, \cdot)$ is a distance metric, e.g., $l_1$ norm distance and $l_2$ norm distance, $k$ denotes layer number, $F\_{\text{adv}}$ is the output of adversarial training model at layer $k$, $F\_{\text{van}}$ is the output of natural examples in vanilla model at layer $k$ (i.e., well-separable/vanilla features). The first term aims to match features of adversarial examples extracted by the adversarial model to well-separable features, while the second term will force features of natural examples extracted by the adversarial model to approximate vanilla features. Different from traditional distillation that only distills features of adversarial examples, $L_{\text{KD}}$ forces both the features $F\_{\text{adv}}(x)$ of natural examples and features $F\_{\text{adv}}(x^{\text{adv}})$ of adversarial examples to match well-separable features. In more detail, we maximize the reduction of perturbation in adversarial examples in adversarial training, which is a way of noise filtering at the feature level. Meanwhile, the training model learns the well-separable features and correctly classifies examples by them.

C. Training Strategy

In this part, we demonstrate the training strategy of VFDAT. Instead of proposing a new adversarial training paradigm, our method is a plug-in for existing adversarial training methods. Based on (3), (4) and (5), we first pre-train a high accuracy vanilla
Algorithm 1: Training of VFDAT.

Input: natural example \(x\), adversarial example \(x^{\text{adv}}\), model parameters \(\theta\), labels \(y\), adversarial training model \(F_{\text{adv}}\), vanilla model \(F_{\text{van}}\), the model layer output \(F^k(\cdot)\), perturbation radius \(\varepsilon\), data set \(D\), \(J\) is the loss function of adversarial training, such as cross-entrop, \(\phi\) is the loss function of adversarial training, such as Kullback Leibler Divergence, number of epochs \(N\), hyperparameter \(\lambda\) and \(\beta\), learning rate \(\eta\).

Output: Model \(F_{\text{adv}}(\theta_{\text{adv}})\)

1: Initialize the adversarial training model \(F_{\text{adv}}\) and freeze vanilla model \(F_{\text{van}}\)
2: For epoch = 1 to \(N\):
3: For batch \((x, y)\) in \(D\):
4: Extract features of natural examples \(F_{\text{van}}^k(x; \theta_{\text{van}})\) from \(F_{\text{van}}\) given \(x\)
5: \(x^{\text{adv}} = \max J(F_{\text{adv}}(x; \theta_{\text{adv}}), y)\)
6: Clip \(x^{\text{adv}}\) to meet \(\|x^{\text{adv}} - x\|_2 \leq \varepsilon\)
7: \(J_{\text{naturat}} = J(F_{\text{adv}}(x; \theta_{\text{adv}}), y)\)
8: \(L_{\text{adv}} = \phi(F_{\text{adv}}(x^{\text{adv}}; \theta_{\text{adv}}), y)\)
9: Extract features of natural examples \(F_{\text{adv}}^k(x; \theta_{\text{adv}})\) from model \(F_{\text{adv}}\) given \(x\)
10: Extract features of adversarial examples \(F_{\text{adv}}^k(x^{\text{adv}}; \theta_{\text{adv}})\) from model \(F_{\text{adv}}\) given \(x^{\text{adv}}\)
11: \(L_{KD} = \|F_{\text{van}}^k(x; \theta_{\text{van}}) - F_{\text{adv}}^k(x^{\text{adv}}; \theta_{\text{adv}})\|_2 + \|F_{\text{van}}^k(x; \theta_{\text{van}}) - F_{\text{adv}}^k(x^{\text{adv}}; \theta_{\text{adv}})\|_2\)
12: \(L_{\text{total}} = L_{\text{naturat}} + \beta \cdot L_{\text{adv}} + \lambda \cdot L_{KD}\)
13: \(\theta_{\text{adv}} = \theta_{\text{adv}} - \eta \cdot \nabla_{\theta_{\text{adv}}} L_{\text{total}}\)
14: End For
15: End For

V. EXPERIMENTS

In this section, we design and conduct experiments on different datasets and models to evaluate the performance of the proposed VFDAT.

A. Experimental Setting

Datasets and Classifiers: For a fair comparison, we follow the previous works [17], [18], [26] to conduct experiments on CIFAR-10 and CIFAR-100 [36]. The CIFAR-10 dataset has 10 classes and per class contains 60,000 images, while the CIFAR-100 dataset has 100 classes and per class contains 600 images. For each dataset, they are split into one training set and one test set in a ratio of 5:1. In addition, we evaluate our method on ResNet-18 [1] and VGG-16 [37], which are widely used in adversarial robustness benchmarks.

Attack Methods: We consider three typical attack methods, i.e., PGD, FGSM, C&W and AutoAttack (AA) [38]. Moreover, we evaluate the robustness of the model in two attack settings (i.e., white-box setting and black-box setting), and we use WidResNet [39] as the surrogate model to generate adversarial examples in the black-box setting.

Baselines and Parameter Settings: To verify the effectiveness of the proposed method, we compare it with four related state-of-the-art adversarial training methods (i.e., ALP [40], TRADES [12], MART [26], ADT [28] and three enhanced methods (i.e., KD [17], BGAT and LBGAT [18]) which can further improve the performance. For these methods, we follow their default parameter settings and use the SGD optimizer [41] with a learning rate of 0.1 and a batch size of 128 to train corresponding robust models. In this paper, we take \(l_1\) norm and \(l_2\) norm to constrain vanilla feature distillation. We conduct all the experiments on a server with one NVIDIA Tesla V100 16 GB GPU.

B. Evaluation on Robustness and Accuracy

In this part, we provide the trade-off between accuracy and robustness with different adversarial training methods, e.g., TRADES and Optimized TRADES, to illustrate the effectiveness of our method. Fig. 3(a) shows the comparison of the accuracy and robustness trade-off between the original TRADES and its corresponding enhanced versions. We vary the robustness parameter \(\beta\) (from 1 to 6) in TRADES to obtain models with different robustness. We can observe that the models trained by different methods exhibit similar robustness when using the same \(\beta\). But our method (i.e., TRADES accompanied by VFDAT) constrained with the \(l_2\) norm outperforms the others significantly in natural accuracy, achieving the best trade-off between accuracy and robustness.
Fig. 3(b) shows the robust accuracy with different attack radius. We apply the PGD (step = 20, $\ell_\infty$, $\varepsilon = 0/255 \sim 10/255$) to generate adversarial examples. $\varepsilon = 0/255$ means no perturbation, which is equivalent to the natural accuracy on the ordinate axis. We can observe that our model outperforms others. The TRADES accompanied by VFDAT can improve the robust accuracy by 2.9% and natural accuracy by 5.4% on average. Meanwhile, the TRADES accompanied by VFDAT also outperforms those state-of-the-art methods (e.g., equipped with KD, LBGAT, etc.) 2.6% in robust accuracy and 4.6% in natural accuracy on average.

To quantitatively compare the performance between the proposed VFDAT and the baselines, we present the robust and natural accuracy of corresponding trained robust ResNet-18 and VGG-16 under several typical attack methods. Table I shows the results on adversarial robustness against white-box attacks and Table II shows the results on adversarial robustness against black-box attacks.

1) Adversarial Robustness Against White-Box Attacks: We tested white-box attacks on CIFAR-10 and CIFAR-100 datasets by PGD, FGSM C&W and AA attacks, which allow full access to the architectures and weights of DNN models. As shown in Table I, our method significantly improves the robust and natural accuracy of the original adversarial training methods (ALP, TRADES, MART and ADT) in different model architectures (ResNet-18 and VGG-16).

For ResNet-18, compared with the original TRADES on CIFAR-10, the TRADES accompanied by VFDAT (i.e., equipped with our proposed method) can further improve the robust accuracy by 2.0% on average and natural accuracy by 4.8%. Meanwhile, the TRADES accompanied by VFDAT also outperforms those state-of-the-art methods 0.7% in robust accuracy (e.g., equipped with the KD) and 3.3% in natural accuracy (e.g., equipped with the BGAT). Compared with the original TRADES on CIFAR-100, the TRADES accompanied by VFDAT can further improve the robust accuracy by 1.8% on average and natural accuracy by 16.0%. Meanwhile, the TRADES accompanied by VFDAT also outperforms those state-of-the-art methods by 1.1% on average in robust accuracy (e.g., equipped with the KD) and 8.0% in natural accuracy (e.g., equipped with the LBGAT).

For VGG-16, compared with the original TRADES on CIFAR-10, the TRADES accompanied by VFDAT can further improve the robust accuracy by 2.1% and natural accuracy by 7.0% on average. Meanwhile, the TRADES accompanied by VFDAT also outperforms those state-of-the-art methods by 0.9% in robust accuracy (e.g., equipped with the LBGAT) and 5.7% in natural accuracy (e.g., equipped with the KD) on average. Compared with the original TRADES on CIFAR-100, the TRADES accompanied by VFDAT can further improve the robust accuracy by 1.6% and natural accuracy by 12.7% on average. Meanwhile, the TRADES accompanied by VFDAT also outperforms those state-of-the-art methods by 0.5% in robust accuracy (e.g., equipped with the KD) and 7.4% in natural accuracy (e.g., equipped with the LBGAT) on average. Moreover, we can draw similar conclusions when combining our method with other adversarial training methods (i.e., ALP, MART and ADT), which indicates the scalability and effectiveness of our method.

2) Adversarial Robustness Against Black-Box Attacks: To test the adversarial robustness against transferred adversarial examples, we report the result of PGD, FGSM C&W and AA based black-box attacks, where the network architectures and weights were not obtainable for the attacker. The black-box robustness of all defense models is reported as shown in Table II. We use the Wide-ResNet as the adversarial example generation model.

For ResNet-18, compared with the original TRADES on CIFAR-10, the TRADES accompanied by VFDAT can further improve the robust accuracy by 6.1% on average, the TRADES accompanied by VFDAT also outperforms the state-of-the-art method 4.2% in robust accuracy (e.g., equipped with LBGAT) on average. Compared with the original TRADES on CIFAR-100, the TRADES accompanied by VFDAT can further improve the robust accuracy by 11.6% on average, the TRADES accompanied by VFDAT also outperforms the state-of-the-art method 8.6% in robust accuracy (e.g., equipped with LBGAT) on average.

For VGG-16, compared with the original TRADES on CIFAR-10, the TRADES accompanied by VFDAT can further improve the robust accuracy by 6.7% on average, the TRADES accompanied by VFDAT also outperforms the state-of-the-art method by 5.2% in robust accuracy (e.g., equipped with KD) on average. Compared with the original TRADES on CIFAR-100, the TRADES accompanied by VFDAT can further improve the robust accuracy by 12.3% on average, the TRADES accompanied by VFDAT also outperforms the state-of-the-art method by 7.2% in robust accuracy (e.g., equipped with LBGAT) on average. Since the target model keeps secret from potential attackers and it is difficult for attackers to design unique adversarial perturbation, compared with the white-box results, all defense methods achieve much better robustness against black-box attacks, even close to the natural accuracy.

C. Feature Visualization Analysis

Furthermore, to qualitatively analyze the effectiveness of our proposed VFDAT, we randomly select five classes in CIFAR-10 and utilize t-SNE [42] to visualize features of 500 natural examples per class and corresponding adversarial examples generated by PGD (step = 20, $\ell_\infty$, $\varepsilon = 8/255$), they come from the layer before the classifier. As shown in Fig. 4, the first and third rows present the t-SNE results of natural examples and the second and fourth rows present the results of corresponding adversarial examples.

For the vanilla model, we can observe that there are 5 clear separable clusters in the case of natural examples but 10 clusters in the case of adversarial examples. The phenomenon indicates that the vanilla model cannot extract right discriminative features from these adversarial examples hence classify them into wrong classes. For all existing defense methods (e.g., TRADE), features of adversarial examples are separated into 5 clusters to some degree, but we can observe that there would appear unexpected obscure boundaries in natural examples, thus leading
| Model      | Dataset | CIFAR10 | CIFAR100 |
|------------|---------|---------|----------|
| Vanilla    | 0.0%    | 22.0%   | 0.0%     | 94.4% | 0.0% | 7.3% | 0.0% | 0.0% | 75.2% |
| ALP        | 50.2%   | 57.1%   | 30.2%    | 45.9% | 84.0% | 27.1% | 31.7% | 23.2% | 24.8% | 51.4% |
| ALP + KD   | 51.1%   | 58.4%   | 33.2%    | 46.3% | 84.5% | 29.4% | 29.8% | 20.1% | 23.4% | 56.0% |
| ALP + BGAT | 51.9%   | 57.5%   | 31.7%    | 46.4% | 85.5% | 30.4% | 33.5% | 21.1% | 24.7% | 56.3% |
| ALP + LBGAT| 52.2%   | 57.1%   | 32.6%    | 45.9% | 85.1% | 31.5% | 31.1% | 22.0% | 23.4% | 59.4% |
| ALP + Our(l1) | 50.1%  | 56.9%   | 30.0%    | 45.4% | 88.8% | 29.3% | 31.9% | 20.6% | 23.2% | 67.4% |
| ALP + Our(l2) | 51.7%  | 60.2%   | 33.2%    | 46.6% | 88.3% | 32.0% | 33.6% | 23.2% | 25.1% | 65.2% |
| TRADES     | 53.1%   | 56.7%   | 53.7%    | 47.9% | 81.1% | 27.4% | 28.0% | 24.5% | 20.7% | 56.0% |
| TRADES + KD| 52.8%   | 57.1%   | 53.2%    | 48.0% | 82.1% | 29.3% | 31.7% | 27.3% | 22.9% | 57.4% |
| TRADES + BGAT | 53.7%    | 57.4%   | 52.3%    | 48.7% | 82.9% | 28.9% | 31.8% | 26.3% | 24.4% | 60.2% |
| TRADES + LBGAT | 54.1%     | 57.6%   | 53.3%    | 48.0% | 83.1% | 30.8% | 32.6% | 25.7% | 23.3% | 61.6% |
| TRADES + Our(l1) | 53.4%     | 59.8%   | 54.9%    | 48.7% | 87.7% | 30.1% | 32.6% | 26.3% | 23.6% | 68.9% |
| TRADES + Our(l2) | 54.7%     | 60.8%   | 55.4%    | 48.6% | 87.5% | 30.5% | 33.1% | 28.4% | 24.4% | 67.3% |
| MART       | 54.1%   | 59.5%   | 46.3%    | 47.1% | 82.3% | 30.2% | 34.0% | 27.2% | 23.5% | 54.9% |
| MART + KD  | 54.9%   | 58.4%   | 45.7%    | 47.6% | 81.5% | 28.3% | 31.3% | 29.4% | 24.7% | 56.6% |
| MART + BGAT | 53.4%    | 58.7%   | 44.7%    | 46.6% | 82.1% | 29.8% | 32.7% | 27.9% | 23.0% | 58.6% |
| MART + LBGAT | 54.1%     | 59.2%   | 46.5%    | 47.3% | 82.4% | 30.4% | 33.5% | 29.3% | 24.1% | 57.2% |
| MART + Our(l1) | 53.8%     | 61.8%   | 47.8%    | 48.0% | 89.7% | 28.7% | 35.1% | 27.9% | 23.9% | 68.7% |
| MART + Our(l2) | 54.6%     | 62.0%   | 49.1%    | 47.5% | 88.6% | 30.5% | 35.4% | 28.9% | 25.4% | 65.8% |
| ADT        | 53.6%   | 57.7%   | 44.1%    | 44.9% | 83.6% | 29.8% | 33.4% | 28.3% | 24.9% | 59.6% |
| ADT + KD   | 52.3%   | 59.2%   | 47.5%    | 45.4% | 83.7% | 30.9% | 34.5% | 30.6% | 25.1% | 59.0% |
| ADT + BGAT | 53.9%   | 58.6%   | 45.9%    | 45.0% | 83.1% | 30.5% | 34.9% | 27.5% | 24.1% | 59.3% |
| ADT + LBGAT | 53.6%     | 59.8%   | 46.4%    | 44.2% | 82.9% | 31.3% | 33.1% | 29.4% | 25.4% | 59.9% |
| ADT + Our(l1) | 54.2%     | 61.4%   | 49.9%    | 46.0% | 88.6% | 32.8% | 34.8% | 30.1% | 27.8% | 65.2% |
| ADT + Our(l2) | 53.7%     | 62.0%   | 47.5%    | 45.3% | 87.6% | 31.8% | 35.2% | 29.7% | 26.9% | 66.7% |

We apply the PGD (step=20, l∞=8/255), FGSM (l∞=8/255), C&W (l2, step=1000, it=0.01), and Auto-attack(l∞=8/255) to generate adversarial examples respectively. The best results are highlighted in bold.
## TABLE II

**Robust and Natural Accuracy of ResNet-18 and VGG-16 Under Several Black-Box (WideResNet is the Surrogate Model) Attacks on CIFAR-10 and CIFAR-100**

| Model     | Dataset | CIFAR10 Accuracy | CIFAR100 Accuracy |
|-----------|---------|------------------|-------------------|
| Vanilla   |         |                  |                   |
| ALP       | 83.1%   | 82.7%            | 72.9%             |
| ALP + KD  | 82.9%   | 82.5%            | 72.6%             |
| ALP + BGAT| 83.5%   | 83.1%            | 73.1%             |
| ALP + LGAT| 82.9%   | 82.4%            | 72.4%             |
| ALP + Ours | 86.2%  | 84.4%            | 87.9%             |
| ALP + Ours | 86.8%  | 85.8%            | 87.5%             |
| TRADES    | 80.0%   | 80.5%            | 78.1%             |
| TRADES + KD | 80.7% | 81.6%            | 81.1%             |
| TRADES + BGAT | 81.1% | 80.6%            | 81.2%             |
| TRADES + LGAT | 81.2% | 81.7%            | 81.8%             |
| TRADES + Ours | 86.0% | 85.2%            | 86.3%             |
| TRADES + Ours | 86.3% | 85.1%            | 86.4%             |
| MART      | 81.2%   | 80.8%            | 81.2%             |
| MART + KD | 79.7%   | 79.3%            | 80.2%             |
| MART + BGAT | 81.1% | 80.5%            | 81.0%             |
| MART + LGAT | 81.0% | 81.4%            | 81.5%             |
| MART + Ours | 88.1% | 87.8%            | 89.0%             |
| MART + Ours | 87.5% | 86.6%            | 87.8%             |
| ADT       | 82.3%   | 81.7%            | 82.3%             |
| ADT + KD  | 81.8%   | 80.2%            | 80.8%             |
| ADT + BGAT | 82.1% | 81.2%            | 81.1%             |
| ADT + LGAT | 81.8% | 81.4%            | 81.8%             |
| ADT + Ours | 86.4% | 87.1%            | 87.9%             |
| ADT + Ours | 86.3% | 86.3%            | 87.1%             |
| Vanilla   |         |                  |                   |
| ALP       | 78.1%   | 79.5%            | 79.2%             |
| ALP + KD  | 79.9%   | 80.5%            | 81.1%             |
| ALP + BGAT | 77.2% | 78.8%            | 78.9%             |
| ALP + LGAT | 78.1% | 79.3%            | 79.2%             |
| ALP + Ours | 84.1% | 83.4%            | 85.9%             |
| ALP + Ours | 83.9% | 83.5%            | 84.8%             |
| TRADES    | 77.4%   | 77.0%            | 77.6%             |
| TRADES + KD | 78.4% | 78.0%            | 78.7%             |
| TRADES + BGAT | 77.0% | 77.3%            | 77.9%             |
| TRADES + LGAT | 78.2% | 77.9%            | 78.2%             |
| TRADES + Ours | 82.8% | 81.4%            | 83.2%             |
| TRADES + Ours | 83.3% | 82.1%            | 84.8%             |
| MART      | 77.2%   | 76.9%            | 77.5%             |
| MART + KD | 76.3%   | 77.5%            | 79.0%             |
| MART + BGAT | 76.9% | 77.4%            | 78.7%             |
| MART + LGAT | 77.2% | 78.9%            | 78.2%             |
| MART + Ours | 83.6% | 84.2%            | 85.7%             |
| MART + Ours | 84.7% | 83.6%            | 85.8%             |
| ADT       | 77.0%   | 76.6%            | 77.9%             |
| ADT + KD  | 79.3%   | 78.4%            | 80.2%             |
| ADT + BGAT | 78.5% | 78.0%            | 79.6%             |
| ADT + LGAT | 82.5% | 82.0%            | 84.0%             |
| ADT + Ours | 83.6% | 81.8%            | 84.3%             |
| ADT + Ours | 84.5% | 82.6%            | 85.4%             |

We apply the PGD (\(\text{step}=20, l_{\infty} = 8/255\)), FGSM (\(l_{\infty} = 8/255\)), C&W (\(l_{\infty}=1000\), \(\epsilon=0.01\)), and Auto-attack (\(l_{\infty} = 8/255\)) to generate adversarial examples respectively on WideResNet. The best results are highlighted in bold.
Fig. 5. (a) Robust and natural accuracy of VFDAT with distilling the outputs of different layers from the vanilla model. VFDAT achieves the best performance in both robust and natural accuracy when using the layer of the last residual block. (b) Robust and natural accuracy of VFDAT with different distillation parameters $\lambda$. When using the layer of the last residual block, VFDAT achieves the best performance in both robust and natural accuracy with $\lambda = 0.01$.

Fig. 6. Comparing the robust and natural accuracy of the model with different adversarial training methods. Our approach works better than feature alignment mechanisms, no matter what kind of adversarial training methods.

D. Ablation Study

1) Effect of Parameters in VFDAT: There are two parameters, i.e., the choice of vanilla feature layer $k$ and the distillation parameter $\lambda$, which affect the performance of the proposed VFDAT. Here, we adopt ResNet18 trained on CIFAR-10 to explore the effects of the two parameters and use $l_2$ norm to constraint features. We use the PGD ($\text{step} = 20, \ell_\infty, \varepsilon = 8/255$) white-box attack to test robust accuracy. Since ResNet18 has four residual blocks, we choose the output of the last Conv layer of each block (e.g., Layer_1, etc.) and fully connected layer respectively, and modify $\lambda$ from 0 to 0.04 with a step of 0.005. As shown in Fig. 5(a), we can observe that the proposed method would achieve the best performance in both robust and natural accuracy when using the layer of the last residual block (i.e., Layer_4). Compared to early layers, the Layer_4 contains more high-level and task-specific information while keeping more spatial information than the logits, hence the Layer_4 comes as a better choice.

In addition, the distillation parameter $\lambda$ also influences robust and natural accuracy. The larger distillation parameter $\lambda$ is expected to guide the robust model to learn more predictive features like the vanilla model. But over-relying on such the vanilla model that has little robustness makes the training model less robust. As shown in Fig. 5(b), a larger $\lambda$ tends to yield higher natural accuracy but diminishes the robustness, while a smaller $\lambda$ would achieve better robustness with compromising the natural accuracy. When $\lambda = 0.01$, VFDAT achieves the best performance in both robust and natural accuracy in the ResNet-18 on CIFAR-10.

2) Effect of Teacher Model in VFDAT: To highlight the contribution of vanilla feature distillation, we conduct the ablation study to compare the performance of existing adversarial training (e.g., ALP, TRADE, and MART), Feature-Aligned Adversarial Training (F-AT) [43] with Vanilla Feature Distillation Adversarial Training (VFDAT). The F-AT only aligns the features of natural examples and adversarial examples for improving the model robustness based on existing adversarial training methods. In contrast, the VFDAT not only aligns the features of natural examples and adversarial examples, but also constrains these features with vanilla features. The main difference between
the F-AT and VFDA T is that vanilla features participate in model adversarial training and guide model adversarial training in the VFDA T. The loss for feature alignment \( \mathcal{L}_F \) can be expressed as:

\[
\mathcal{L}_F = \frac{1}{N} \sum_{i=1}^{N} \left( \Gamma \left( \mathcal{F}_{\text{adv}}^k (x_i), \mathcal{F}_{\text{adv}}^k (x_i^{\text{adv}}) \right) \right),
\]

(7)

where \( \Gamma (\cdot, \cdot) \) is a \( l_2 \) norm distance metric in this part. The loss for Feature-align Adversarial Training \( \mathcal{L}_{FAT} \) and the loss for our method can be expressed as:

\[
\mathcal{L}_{FAT} = \mathcal{L}_{AT} + \mathcal{L}_F,
\]

(8)

\[
\mathcal{L}_{VFDA T} = \mathcal{L}_{AT} + \mathcal{L}_{KD},
\]

(9)

where \( \mathcal{L}_{AT} \) is the existing adversarial training method, e.g., ALP, TRADE, and MART. We use the PGD (step = 20, \( \varepsilon = 8/255 \)) white-box attack to test robust accuracy of model (e.g., ResNet-18) on CIFAR-10 datasets as shown in Fig. 6. Compared with the original TRADES, the TRADES accompanied by FAT can improve the robust accuracy by 1% and natural accuracy by 2%. Compared with the original TRADES, the TRADES accompanied by VFDA T can further improve the robust accuracy by 2%, the accuracy has increased by 8% in particular. These results are persuasive with a validity of the knowledge distillation of the teacher model for achieving the better trade-off between accuracy and robustness.

VI. CONCLUSION

In this work, we proposed Vanilla Feature Distillation Adversarial Training (VFDA T) to achieve a better trade-off between accuracy and robustness. VFDA T conducts knowledge distillation from a high-accuracy pre-trained model to guide the adversarial training model to learn more predictive/well-separable features, which allows the model to preserve the accuracy and improve robustness as well. Specifically, the proposed method can be flexibly adapted to and boost existing adversarial training methods. Extensive experiments conducted across different datasets, network architectures, and adversarial training algorithms demonstrate the state-of-the-art performance of our method. We hope that our method can serve as the adversarial robustness benchmark and inspire the community to further ameliorate the accuracy-robustness trade-off.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 630–645.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 84–90.
[3] T. B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” Comput. Vis. Image Understanding, vol. 81, no. 3, pp. 231–268, 2001.
[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
[5] J. Levinson et al., “Towards fully autonomous driving: Systems and algorithms,” in Proc. IEEE Intell. Veh. Symp., 2011, pp. 163–168.
[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, arXiv:1706.06083.
[7] J. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.
[8] A. Kurakin et al., “Adversarial attacks and defenses competition,” in Proc., Conf. Buildling Intell. Syst., 2018, pp. 195–231.
[9] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent advances in adversarial training for adversarial robustness,” 2021, arXiv:2102.01356.
[10] D. Chen et al., “CARTL: Cooperative adversarially-robust transfer learning,” in Proc. Int. Conf. Mach. Learn., PMLR, 2019, pp. 7472–7482.
[11] Y. Ganin et al., “Domain-adversarial training of neural networks,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 2096–2030, 2016.
[12] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, “Theoretically principled trade-off between robustness and accuracy,” in Proc. Int. Conf. Mach. Learn., PMLR, 2019, pp. 4731–4739.
[13] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation methods,” in Proc. Workshop Adversarial Mach. Learn., 2021, pp. 640–647.
[14] R. Rade and S.-M. Moosavi-Dezfooli, “Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy versus robustness trade-off,” in Proc. Workshop Adversarial Mach. Learn., 2021, pp. 1–10.
[15] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, and K. Ren, “Feature importance-aware transferable adversarial attacks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 7639–7648.
[16] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, “Adversarial examples are not bugs, they are features,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 125–136.
[17] T. Chen, Z. Zhang, S. Liu, S. Chang, and Z. Wang, “Robust overfitting may be mitigated by properly learned smoothing,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–19.
[18] C. Cui, S. Liu, L. Wang, and J. Jia, “Learnable boundary guided adversarial training,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 1572–15730.
[19] Y. Dong et al., “Boosting adversarial attacks with momentum,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9185–9193.
[20] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” 2017, arXiv:1705.07204.
[21] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” 2016, arXiv:1611.02770.
[22] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2017, pp. 39–57.
[23] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, “Generative adversarial perturbations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4422–4431.
[24] A. N. Bhagoji, W. He, B. Li, and D. Song, “Practical black-box attacks on deep neural networks using efficient query mechanisms,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 154–169.
[25] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in Proc. ACM Conf. Commun. Secur., 2017, pp. 506–519.
[26] Y. Wang, D. Zou, Y. Ji, J. Bailey, X. Ma, and Q. Gu, “Improving adversarial robustness requires revisiting misclassified examples,” in Proc. Int. Conf. Learn. Representations, 2019, pp. 1–14.
[27] Z. Jiang, T. Chen, T. Chen, and Z. Wang, “Robust pre-training by adversarial contrastive learning,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 16199–16210.
[28] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su, “Adversarial distributional training for robust deep learning,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 8270–8283.
[29] H. Xiong et al., “GroD: Deep learning with gradients orthogonal decomposition for knowledge transfer, distillation, and adversarial training,” ACM Trans. Knowl. Discov. Data, vol. 16, no. 6, pp. 1–25, 2022.
[30] Y. Qian, Y. Zhang, Q. Wen, Y. Ye, and C. Zhang, “Rep2vec: Repository embedding via heterogeneous graph adversarial contrastive learning,” in Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining, 2022, pp. 1390–1400.
[31] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 9, pp. 2805–2824, Sep. 2019.
[32] J. R. Hershey and P. A. Olsen, “Approximating the kullback leibler divergence between gaussian mixture models,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2007, pp. IV–317.
Guodong Cao is currently working toward the PhD degree with the School of Cyber Science and Engineering, Wuhan University. His research interest focuses on privacy protection, AI security and edge intelligence.

Zhibo Wang (Senior Member, IEEE) received the BE degree in automation from Zhejiang University, China, in 2007, and the PhD degree in electrical engineering and computer science from the University of Tennessee, Knoxville, in 2014. He is currently a professor with the School of Cyber Science and Technology, Zhejiang University, China. His current research interests include AI security, data security and privacy, and mobile crowdsensing systems. He is member of ACM.

Xiaowei Dong is currently working toward the BE degree in information security with the School of Cyber Science and Engineering, Wuhan University. His research interest focuses on AI security.

Zhifei Zhang received the BS and MS degrees from the Northeastern University and Zhejiang University, China in 2010 and 2013, respectively, and the PhD degree in computer engineering from the University of Tennessee, Knoxville in 2018 supervised by Prof. Hairong Qi. He is currently a research engineer in Adobe Research, San Jose. His interest lies in deep learning based image synthesis, computer vision, text detection and recognition, etc.

Hengchang Guo is currently working toward the BE degree in information security with the School of Cyber Science and Engineering, Wuhan University. His research interest focuses on AI security.

Zhan Qing (Senior Member, IEEE) received the PhD degree from the Computer Science and Engineering department, State University of New York at Buffalo in 2017. He is currently a ZJU100 young professor, with both the College of Computer Science and Technology and the Institute of Cyberspace Research (ICSR), Zhejiang University, China. He was an assistant professor with the Department of Electrical and Computer Engineering, the University of Texas at San Antonio. His current research interests include data security and privacy, secure computation outsourcing, artificial intelligence security, and cyber-physical security in the context of the Internet of Things. His works explore and develop novel security sensitive algorithms and protocols for computation and communication on the general context of Cloud and Internet devices.

Kui Ren (Fellow, IEEE) received the PhD degree from the Worcester Polytechnic Institute. He is currently a professor and the associate dean with the College of Computer Science and Technology, Zhejiang University, where he also directs the School of Cyber Science and Technology. Before that, he was the SUNY Empire Innovation professor of The State University of New York at Buffalo. His current research interests include Data Security, IoT Security, AI Security, and Privacy. He received Guohua Distinguished Scholar Award from ZJU in 2020, IEEE CISTC Technical Recognition Award in 2017, SUNY Chancellor’s Research Excellence Award in 2017, Sigma Xi Research Excellence Award in 2012 and NSF CAREER Award in 2011. He has published extensively in peer-reviewed journals and conferences and received the Test-of-time Paper Award from IEEE INFOCOM and many Best Paper Awards from IEEE and ACM including Mobisys’20, ICDCS’20, Globecom’19, ASIACCS’18, ICDCS’17, etc. His h-index is 74, and his total publication citation exceeds 32000 according to Google Scholar. He is a fellow of ACM and a Clarivate Highly-Cited researcher. He is a frequent reviewer for funding agencies internationally and serves on the editorial boards of many IEEE and ACM journals. He currently serves as chair of SIGSAC of ACM China.