Push Stricter to Decide Better: A Class-Conditional Feature Adaptive Framework for Improving Adversarial Robustness

Jia-Li Yin, *Member, IEEE*, Wanqing Zhu, Bin Chen, Bo-Hao Chen, *Member, IEEE* and Ximeng Liu

***Abstract*—In response to the threat of adversarial examples, adversarial training provides an attractive option for enhancing the model robustness by training models on online-augmented adversarial examples. However, most of the existing adversarial training methods focus on improving the robust accuracy by strengthening the adversarial examples but neglecting the increasing shift between natural data and adversarial examples, leading to a decrease in natural accuracy. To maintain the trade-off between natural and robust accuracy, we alleviate the shift from the perspective of feature adaption and propose a Feature Adaptive Adversarial Training (FAAT) optimizing the class-conditional feature adaption across natural data and adversarial examples. Specifically, we propose to incorporate a class-conditional discriminator to encourage the features become (1) class-discriminative and (2) invariant to the change of adversarial attacks. The novel FAAT framework enables the trade-off between natural and robust accuracy by generating features with similar distribution across natural and adversarial data within the same class, and achieves higher overall robustness benefited from the class-discriminative feature characteristics. Experiments on various datasets demonstrate that FAAT produces more discriminative features and performs favorably against state-of-the-art methods.**

***Index Terms*—Adversarial example, adversarial training, model robustness, feature adaption**

I. Introduction

Deep feed-forward networks have brought great advances to the state-of-the-art across various machine-learning tasks and applications. However, these advances in performance can be dramatically degraded when faced with maliciously crafted perturbations, i.e., *adversarial examples*. This phenomenon has raised growing concerns over the safety and reliability of deep feed-forward networks and significantly hinders the deployment of these state-of-the-art architectures in practical [1]–[6].

In response to the threat of adversarial attacks, learning to increase model robustness by augmenting training data with adversarial examples is known as *adversarial training*. A number of approaches to adversarial training has been suggested in a formulation of min-max optimization, e.g., generating online adversarial examples that maximize the model loss, and then the model is trained on these adversarial examples to minimize the loss, so that the model can prefer robust features when given adversarial data. The appeal of adversarial training approaches is to encourage the model to learn the distributions of adversarial data. This strategy can effectively improve the model robustness and has become the most popular way to defend adversarial examples. At the same time, however, these improvements of robustness against adversarial examples typically come at the cost of decreased natural data accuracy as there is a shift between adversarial and natural data distribution, as shown in Fig. 1. Especially when previous works [7], [8] focus on maximizing the attack strength of online generated adversarial examples to improve robustness, these methods work poorly on natural data (i.e., drop at least 5% on CIFAR-100 dataset), as reported in Table I.

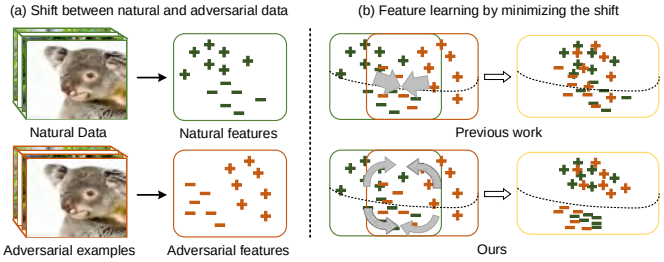

Fig. 1. Illustration for our method. (a) Shift between natural and adversarial data. Note that the adversarial examples are generated by adding imperceptible perturbations over the natural data. Human are not sensitive to the perturbations but CNN models can be easily affected as there is a clear shift between natural and adversarial feature distributions. (b) The comparison of previous methods and ours. Previous methods focus on minimizing the feature discrepancy between natural data and adversarial examples. Our method take class-level information into consideration for fine-grained feature adaption.

In this paper, we propose to alleviate the shift between natural and adversarial data distribution. There have been some pioneering works trying to address this problem. Kannan *et al*. [10] proposed adversarial logit pairing (ALP) method to encourage the logits from original inputs and adversarial examples in the CNN model to be similar. Zhang *et al*. [9] provided a regularizer, namely TRADES, to minimize the KL-divergence between the features of original inputs and adversarial examples. Cui *et al*. [11] proposed to parallelly train two models using original input and adversarial examples separately while the natural model is used to guide the training of robust model. Despite the improvements, these methods

This paper was supported in part by the National Natural Science Foundation of China under Grant Nos. 62072109 and U1804263; in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 111-2628-E-155-003-MY3; and in part by Youth Foundation of Fujian Province, P.R.China, under Grant No. 2021J05129. (*Corresponding authors: Ximeng Liu and Bo-Hao Chen*)

J.-L. Yin, W. Zhu, B. Chen, and X. Liu are with the College of Computer Science and Big Data, Fuzhou University, Fuzhou, 350108, China. E-mail: jlyin@fzu.edu.cn; snbnix@gmail.com

B.-H. Chen is with the Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan. E-mail: bhchen@saturn.yzu.edu.tw

TABLE I
COMPARISON OF PREVIOUS METHODS AND OUR METHOD UNDER PGD-20/CW-20 ATTACK FOR WIDERESNET-34-10 ON MNIST, CIFAR-10, AND CIFAR-100, RESPECTIVELY. WE HIGHLIGHT THE HIGHER ACCURACY IN BLUE AND THE LOWER IN RED COMPARED TO THE STANDARD ADVERSARIAL TRAINING.

Denfese Method	Trade-off term	MNIST			CIFAR-10			CIFAR-100		
		Natural	PGD-20	CW-20	Natural	PGD-20	CW-20	Natural	PGD-20	CW-20
Natural training	✗	99.50	0.07	0.00	95.80	0.00	0.00	78.76	0.00	0.00
Standard (PGD-10) [7]	✗	99.41	96.01	95.87	85.23	48.93	48.74	60.29	26.84	26.25
+ATTA [8]	✗	99.60	98.20	97.72	83.30	54.33	54.25	55.09	23.23	22.85
+TRADES ($\lambda = 3$) [9]	✓	99.48	96.50	95.79	85.98	54.86	54.02	62.37	25.31	24.53
+ALP [10]	✓	99.27	97.80	97.26	84.01	54.96	54.71	55.60	28.28	24.40
+Ours	✓	**99.70**	**98.53**	**97.89**	**86.74**	**55.18**	**54.79**	**62.58**	**30.22**	**30.58**

have two major issues. First, these methods generally employ arbitrary point-to-point metrics to evaluate the distribution discrepancy which does not guarantee the well distribution alignment. Second, there exist many feature points near the decision boundary, they may overfit the natural data but are less discrminative for the adversarial examples, as shown in Figure 1.

unlike the previsous methods, we proposed to alleviate the shift from the per Unlike the previous methods, we propose to alleviate the shift from the perspective of feature adaption. To address the above issues, we focus on learning features that are both class-level discriminative and invariant to the change of adversarial attacks, i.e., learning the same or very similar distributions across the natural and adversarial data with same class. In this way, the obtained network can maintain the accuracy on both natural and adversarial domains. It is conceptually similar to the problem of domain adaption [12]–[14], which aims to alleviate the domain shift problem by aligning the feature distributions of different domains. Recent success in domain adaption lies in the incorporation of a discriminator to model the feature distribution, where the discriminator aims to distinguish the features from different domains while the trained network tries to fool the discriminator, so the features become domain-invariant.

Motivated by this, we take advantage of such a discriminator in coming up with a novel adversarial training framework: *Feature Adaptive Adversarial Training* (FAAT), where a discriminator is incorporated to encourage the similar distribution of across natural data and adversarial examples. However, the simple discriminator only encourages distribution similarity across domains but neglects the underlying class structures. To make the features near the decision boundary more discriminative, the feature discrepancy among classes should be further ensured. Inspired by [12], we propose to directly encode class knowledge into the discriminator and encourage class-level fine-grained feature adaption of adversarial features. Specifically, the output of discriminator is transferred from binary domain label into class-conditional domain label. Such class-conditional discriminator can well assist the fine-grained feature adaption across natural data and adversarial examples according to their corresponding classes. In summary, the proposed FAAT has the following advantages:

- FAAT can effectively maintain the trade-off between natural and robust accuracy by encouraging feature adaption across natural data and adversarial examples.
- FAAT can significantly improve the overall robustness of adversarial training by incorporating class knowledge into feature adaption.
- FAAT provides an universal framework for adversarial training which can be easily combined with other methods.

We conducted extensive experiments on diverse datasets. The experimental results show that the proposed method achieves impressive improvements. For example, as can be seen in Table I, our proposed method improves the natural and PGD-20 robust accuracy of model trained with standard adversarial training from 60.29% to 62.58% and 26.84% to 30.22% on CIFAR-100 dataset. Our contributions in this work can be concluded as:

- We propose to alleviate the increasing shift between natural and adversarial examples in adversarial training from the perspective of feature adaption, which we believe could guide new thought-provoking directions for future work.
- We present a *Feature Adaptive Adversarial Training* (FAAT) framework which incorporates a class-conditional discriminator into the adversarial training for encouraging features to be both class-discriminative and invariant to the adversarial attacks.
- We analyze our proposed FAAT by conducting extensive experiments on CIFAR-10, CIFAR-100, and TinyImageNet datasets, and show that FAAT substantially improves the trade-off between the natural and robustness accuracy with relatively light computational burden.

The rest of this paper is organized as follows. Section II provides a survey of previous adversarial attack and defense methods; Section III describes the proposed class-conditional FAAT in details; Section IV conducts a sequence of experiments to validate both effectiveness and efficiency of our FAAT; and Section V draws conclusions of this work.

II. RELATED WORK

A. Adversarial attacks

An adversarial example refers to a maliciously crafted input that is imperceptible to humans but can fool the machine-learning models. It is typically generated by adding a perturbation δ subject on constraint $\mathcal{S}$ to the original input x, i.e.,

$x^{adv} = x + \delta$, $\delta \in \mathcal{S}$. Szegedy *et al*. [15] first observed the existence of adversarial examples and used a box-constrained L-BFGS method to generate δ. Goodfellow *et al*. [16] investigated the linear nature of networks and proposed the fast gradient sign method (FGSM) to generate more effective adversarial examples. Following this work, various attacks are proposed to generate stronger δ that can maximize the model training loss. Specifically, Madry *et al*. [7] divided one-step FGSM into multiple small steps to iteratively enhance the attack strength, known as Projected Gradient Descent (PGD) attack. Carlini *et al*. [17] proposed CW attack based on the margin loss to enhance the attack. Croce *et al*. [18] ensemble multiple attacks into a more powerful attack, namely Auto-Attack (AA), which is known as the strongest attack so far.

An intriguing property of adversarial examples is that they can be transferred to attack different network architectures [16], [19]. Liu *et al*. [20] first analyzed the transferability of adversarial examples and used ensemble-based methods to generate adversarial examples for attacking unknown models. Huang *et al*. [21] proposed to enhance the transferability of adversarial examples by increasing its perturbation on a pre-specified layer of the source model. Dong *et al*. [22], Xie *et al*. [23], Wu *et al*. [24] and Wang *et al*. [25] further boosted the transferability of adversarial examples via gradient momentum, diverse input patterns, attention scheme and invariance tuning, respectively, making the adversarial defense more challenging.

B. Adversarial defenses

Various defense methods against adversarial attacks have been proposed over the recent years, including detection-based methods [26]–[29], input transformation-based methods [30], and training-based methods [7], [8], [16], [31]–[36]. Here we focus on the most related and effective one: adversarial training methods. Adversarial training performs online defending which aims to minimize the loss of model on online generated adversarial examples that maximize the model loss at each training epoch. This process can be described as:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{X}}[\max_{\delta\in\mathcal{S}} \mathcal{L}(x+\delta, y; \theta)], \tag{1}$$

where (x, y) is a training pair in dataset $\mathcal{X}$, $\mathcal{S}$ denotes the region within the ϵ perturbation range under the ℓ_∞ threat model for each example, i.e., $\mathcal{S} = \{\delta : \|\delta\|_\infty \leq \epsilon\}$, where the adversary can change input coordinate x_i by at most ϵ. It formulates a game between adversarial examples and model training. The stronger adversarial examples are generated; more robust model can be achieved.

Thus existing adversarial training focus on maximizing the attack strength of adversarial training. Madry *et al*. [7] used PGD to approximate the adversarial attack for adversarial training. Cai *et al*. [35] and Zhang *et al*. [34] further incorporated curriculum attack generation and early stopping into the PGD adversarial training to make it more efficient and practical. On the other hand, Tramer *et al*. [33] proposed ensemble adversarial training where the adversarial examples are generated by multiple ensemble models to enhance the attack strength. Several methods are proposed to further promote the diversity in the ensemble models, including output logits diversification [37], gradient direction minimization [38], and vulnerability diversity maximization [39]. These methods can effectively improve the robustness of models but generally have the limitation of a decrease of natural accuracy due to the ignorance of the shift between natural and adversarial data distribution.

C. Trade-off between natural and robust accuracy

Besides simply pursuing the improvement over robustness, a recent line of work investigates the trade-off between natural and robust accuracy [9]–[11], [40]–[43]. Based on the theoretical insight [10], most works follow the path of shortening the distance of natural data and the corresponding adversarial examples in the CNN models. Progress has been achieved by adversarial logit pairing (ALP) in [10], KL-divergence based TRADES in [9]. Furthermore, Wang *et al*. [40] explicitly differentiates the misclassified and correctly classified examples during the training and minimize the KL-divergence for misclassified examples. Cui *et al*. [11] proposed to parallelly train two models using original input and adversarial examples separately while the natural model is used to guide the training of robust model. Although the ideas behind these methods are intuitive for encouraging the same representation of original inputs and adversarial examples in CNN models, these regularizations arbitrarily used point-to-point metrics and do not in practice align well with learning feature distributions. Wang *et al*. [42] took the trade-off parameter into the learning process to calibrate a trained model in-situ. However, directly considering trade-off as a learning process lacks explicit feature generation guidance, thus leads to unsatisfied results. A co-current work is [44], they directly incorporate domain adaption into adversarial training without considering the class-level alignment. A fine-grained feature adaption scheme is still lacking.

III. Methodology

Given a CNN model G trained on dataset $\mathcal{X}$ for K-class classification, We denote a natural data pair $(x, y) \in \mathcal{X}$ where $y \in \{0, ..., K-1\}$. Adversarial examples are generated by adding a perturbation over x, i.e., $x^{adv} = x + \delta$, which can lead to $G(x^{adv}) \neq y$. The goal of our work is to learn a model G which could achieve a low expected risk on both x and x^{adv}. We discuss the FAAT in the context of feature generation in CNN models. For more clear illustration, we divide the model G into a feature extractor F and a multi-class classifier C, where $G = F \circ C$. In the following, we first introduce the class-conditional feature adaption scheme and then discuss the objective and training procedure of our FAAT framework.

A. Class-conditional feature adaption

Recent success in domain adaption [12]–[14] reveal that feature distribution captured by a discriminator can be used to better measure the feature discrepancy between two domains. In these frameworks, a discriminator is usually equipped to

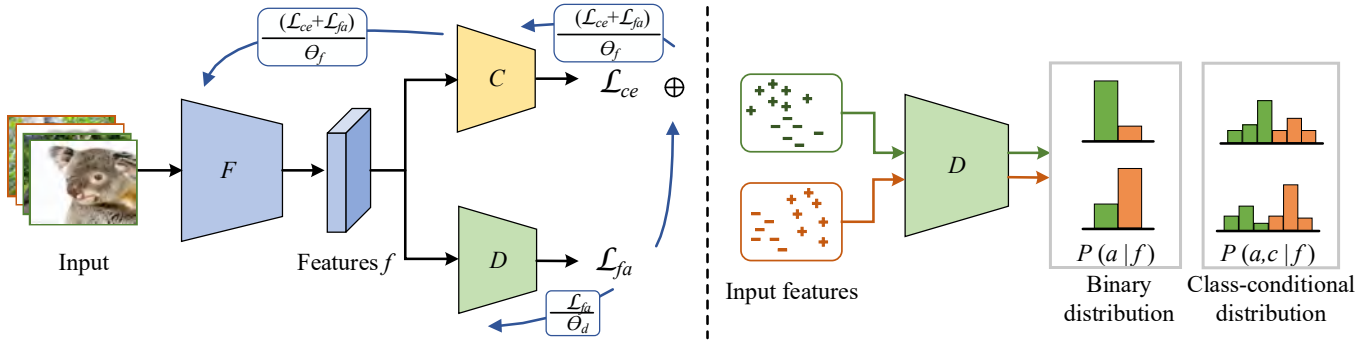

Fig. 2. Overview of the proposed FAAT. **Left**: The training process of FAAT. We adopt a discriminator D in the adversarial training framework to distinguish the feature distribution across natural data and adversarial examples, while the network G is trained to generate invariant features against adversarial attacks. Here we divide G into a feature extractor F and a classifier C for clear illustration, where $G = F \circ C$. **Right**: The class-conditional discriminator. We change the traditional discriminator which uses binary domain label into a class-conditional discriminator with multiple output to encourage the fine-grained class-level feature adaption.

model distribution $P(a \mid f) \in [0, 1]$ given feature f, where a denotes the data domain. By learning features that can confuse the discriminator, i.e., expecting $P(a = 0 \mid f) \approx P(a = 1 \mid f)$ where 0 and 1 stands for different domains, the features can be domain-invariant.

Inspired by this, Bashivan *et al*. [45] first proposed to adopt such a discriminator to encourage the invariant feature generation against the change of adversarial attacks. Specifically, the discriminator is trained to model distribution with the objective of $P(a = 0 \mid f) = 1$ and $P(a = 1 \mid f^{adv}) = 1$, where $a = 0$ and 1 denote attributes of natural data and adversarial examples, respectively, f and f^{adv} denotes the features of natural data and adversarial examples. Then the CNN model can be trained to fool the discriminator, i.e., expecting $P(a = 0 \mid f^{adv}) = 1$. The F in G is encouraged to generate the domain-invariant features by solving:

$$\begin{aligned} \min_F \mathcal{L}_{fa} &= -\sum_{i=1}^{n} \log D(F(x_i + \delta_i)) \\ &= -\sum_{i=1}^{n} \log P(a = 0 \mid F(x_i + \delta_i)), \end{aligned} \tag{2}$$

where n is the number of data pairs in dataset $\mathcal{X}$, and δ is the adversarial perturbation which can be optimized with PGD [7] subjected to $\|\delta_i\|_\infty \leq \epsilon$. By minimizing $\mathcal{L}_{fa}$, the features generated from adversarial examples are globally similar to the features from natural data, i.e., $f^{adv} \approx f$.

However, it is not precise to only focus on feature invariance but neglect the underlying class-level structure difference. To take class discrimination into account, we propose to expand the discriminator to model class conditional distribution according to [12]. Specifically, the output of discriminator is changed from binary channels into multiple channels as $P(a, c \mid f) \in [0, 1]$, where $c \in \{0, ..., K-1\}$ denotes the class label, and $P(a \mid f) = \sum_{c=0}^{K-1} P(a, c \mid f)$. With this design, the predicted probability for domains is represented as a probability over different classes in specific domain, which enables the class-discriminative but domain-invariant feature generation. Thus the equation in Eq. (2) becomes:

$$\begin{aligned} \min_F \mathcal{L}_{fa} &= -\sum_{i=1}^{n} \log D(F(x_i + \delta_i)) \\ &= -\sum_{i=1}^{n} \sum_{k=0}^{K-1} d_{i,k} \log P(a = 0, c = k \mid F(x_i + \delta_i)), \end{aligned} \tag{3}$$

where $d_{i,k}$ represents the class knowledge, $d_{i,k} = 1$ when x_i belongs to the k-th class, otherwise $d_{i,k} = 0$. The expectation is to push $P(a = 0, c = k \mid f^{adv}) = 1$, meaning that the feature distribution of adversarial examples belonging to k-th class are expected to be similar with natural data belonging to k-th class.

B. Adversarial training with feature adaption

We propose to incorporate the class-conditional feature adaptive objective in Eq. (3) into the adversarial training process to induce the trade-off between natural and robust accuracy. The overall training framework is illustrated in the left part of Fig. 2. Since our training method relies on a discriminator to align the features, we conduct the training process by alternatively optimizing the trained model and discriminator as follows:

Step 1: The discriminator D is first trained to model the distribution of features from different domains. This can be achieved by:

$$\begin{aligned} \min_D &\mathbb{E}_{(x) \sim p_{x,c}} \log D(F(x)) \\ &+ \mathbb{E}_{(x^{adv}) \sim p_{x^{adv},c}} [1 - \log D(F(x^{adv}))], \end{aligned} \tag{4}$$

where $D(F(x)) = \log P(a, c \mid F(x))$. Note that F is fixed during the optimization of D. The goal of discriminator D is to distinguish the features of natural data and adversarial examples in class-level.

Step 2: The model G is trained with the task loss on the adversarial examples and the feature adaption loss output from the class-conditional discriminator. The task loss is used to train discriminative features for classification. Here we use most popular classification loss, i.e., cross-entropy loss $\mathcal{L}_{ce}$

Algorithm 1: Feature Adaptive Adversarial Training (FAAT)

Input : Training data $\mathcal{D} = \{X, Y\}$, perturbation bound ϵ, training epoch N, batch size B, learning rate lr.
Output: Trained model G, D with parameter θ_g and θ_d.
Initialize model G and D randomly or with pre-trained configuration.
Initialize δ from uniform $(-\epsilon, \epsilon)$.
for *epoch = 1 ... N* **do**
  for *i = 1 ... B* **do**
    $\delta_i \leftarrow \alpha \cdot \text{sign}(\nabla_{\delta_i} \mathcal{L}(G(x_i + \delta_i), y_i))$; `// Generation of perturbation using PGD-10`
    $\delta_i \leftarrow \max(\min(\delta_i, \epsilon), -\epsilon))$;
    $x_i^{adv} \leftarrow x_i + \delta_i$; `// Generation of perturbation`
    Compute $\mathcal{L}_d$ based on Eq. (4);
    $\theta_d \leftarrow \theta_d - lr \nabla_{\theta_d} \frac{\partial \mathcal{L}_d}{\partial \theta_d}$; `// Update D`
    Compute $\mathcal{L}_{ce}$ and $\mathcal{L}_{fa}$ based on Eq. (3) and (5) ;
    $\theta_g \leftarrow \theta_g - lr \nabla_{\theta_g} \frac{\partial (\mathcal{L}_{ce} + \mathcal{L}_{fa})}{\partial \theta_g}$; `// Update G`
  end
end
return *model G.*

for classification. The feature adaption loss is used to push the model to generate class-conditional invariant features that can fool the discriminator. The overall objective can be represented as:

$$\min_G \mathbb{E}_{(x,y)\sim\mathcal{X}}[\mathcal{L}_{ce}(G(x+\delta), y) + \lambda \mathcal{L}_{fa}], \tag{5}$$

where λ is the weighting factor which is set as 0.0015 in our experiments. More adjustments to this factor are illustrated in Sec. IV-E.

The objective in Eq. (5) can be understood as adding a feature alignment constraint into adversarial training process. However, our method is fundamentally different from other trade-off methods. In our method, we employ a adversarial framework where a discriminator is equipped to model the feature distribution, which is more guaranteed than the MSE-based metric in ALP [10] or KL distance-based metric in TRADES [9] and MART [40]. Moreover, our method should also be distinguished from that of [44] and [46], which only focus on generating globally similar representations of natural data and adversarial examples, neglecting the class-level structure difference which is native in classification tasks. In our method, we propose to directly incorporate the class knowledge into the discriminator and ask for more fine-grained feature adaption. This procedure simultaneously minimizes the feature discrepancy between natural data and adversarial examples, and maximizes the discrimination among different classes, so that helps the model capture more discriminative features. The entire training process of FAAT is elaborated in Sec. III-D.

C. Discriminator architecture

In our FAAT framework, the discriminator is used to distinguish the feature distribution belongings in a class level. Since the traditional discriminator only consider binary classification, we need a special design for the class-conditional discriminator here. We consider the design of our discriminator from two aspects: (1) model architecture; and (2) multi-class label encoding. For the model architecture, since the input of the discriminator is the deep features generated by the target trained model, we can design the discriminator with a simple architecture. We adopt the discriminator architecture in [12], which comprises two 3×3 convolutional layers followed by LeakyReLU activation function, and then two fully connected layers are applied to predict the class-conditional possibility over natural and adversarial domain, respectively. An overview of the discriminator architecture is shown in Fig. 3.

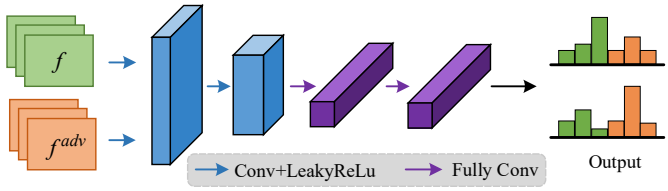

Fig. 3. Illustration of the multi-class discriminator architecture in our FAAT framework.

Another problem we are facing with the incorporation of discriminator is the encoding of multi-class labels. The labels for training a traditional discriminator are binary, namely [1,0] and [0,1], for the natural and adversarial data domain respectively. To incorporate the class knowledge into the discriminator, we first transform the labels into $[\mathbf{d}; \mathbf{0}]$ and $[\mathbf{0}; \mathbf{d}]$, where $\mathbf{d} \in \mathbb{R}^{K\times 1}$ denotes the class knowledge, and $\mathbf{0} \in \mathbb{R}^{K\times 1}$ is an all-zero vector. In our task, the labels of adversarial examples are known so we can directly use one-hot labels for generating $\mathbf{d}$. Specifically, to obtain better generalization of trained models, we use soft labels instead of hard labels as follows:

$$\mathbf{d}_k = \begin{cases} 1-\alpha & \text{if } k \text{ is the target label} \\ \frac{\alpha}{K-1} & \text{otherwise} \end{cases}, \tag{6}$$

where $\mathbf{d}_k$ denotes the k-th entry of soft label logits, and α is a temperature to encourage soft probability distribution over non-target classes, which can be empirically set as 0.1.

TABLE II
ROBUSTNESS OF MODELS TRAINED WITH DIFFERENT ADVERSARIAL TRAINING METHODS ON CIFAR-10 AND CIFAR-100 DATASETS. ALL STATISTICS ARE EVALUATED AGAINST PGD/CW ATTACKS WITH 20/100 ITERATIONS AND A RANDOM RESTART FOR $\epsilon = 8/255$.

Dataset	Defense method	Trade-off term	Natural	PGD-20	PGD-100	CW-20	CW-∞	AA
CIFAR-10	Standard [7]	✗	85.23	48.93	45.74	48.74	46.75	44.27
	+FREE (m=8) [31]	✗	85.75	45.76	45.52	44.95	44.45	43.22
	+ATTA-1 [8]	✗	83.36	50.05	49.90	49.02	48.75	45.21
	+ATTA-10 [8]	✗	84.43	54.65	53.54	54.25	52.15	50.09
	+CCG [47]	✗	**88.32**	54.71	52.79	54.34	52.98	51.00
	+ALP [10]	✓	86.45	53.12	52.59	52.85	52.74	44.96
	+TRADES($\lambda = 3$) [9]	✓	85.98	54.86	53.97	54.02	53.58	51.05
	+Ours	✓	86.74	**55.18**	**54.26**	**54.79**	**54.45**	**52.13**
CIFAR-100	Standard [7]	✗	60.29	26.84	25.21	26.44	25.43	22.48
	+FREE (m=8) [43]	✗	60.11	26.79	22.66	25.69	25.60	22.83
	+ATTA-1 [8]	✗	59.07	21.58	22.82	21.14	20.92	18.23
	+ATTA-10 [8]	✗	55.09	23.23	21.59	22.85	21.81	20.11
	+CCG [47]	✗	60.74	27.27	25.83	26.04	25.39	23.22
	+ALP [10]	✓	59.65	28.28	27.97	27.14	27.01	22.60
	+TRADES($\lambda = 3$) [9]	✓	62.37	25.31	24.89	24.53	24.19	22.24
	+Ours	✓	**62.58**	**30.22**	**30.12**	**30.58**	**30.09**	**27.87**
Tiny-ImageNet	Standard [7]	✗	43.27	19.19	19.11	18.21	17.05	11.21
	+CCG [47]	✗	**47.43**	22.34	22.19	20.03	18.95	16.66
	+TRADES($\lambda = 3$) [9]	✓	42.74	17.84	17.68	14.36	13.65	13.26
	+Ours	✓	45.59	**22.58**	**22.73**	**21.42**	**20.21**	**18.64**

D. Training routine

Algorithm 1 shows the pseudo-code of our proposed FAAT training routine. We first randomly initialize the weights of discriminator D and the target CNN model G. Then for each batch of training data during the training phase, a PGD optimization is first applied on training data to generate the corresponding adversarial examples. To distinguish the feature distribution of natural data and adversarial examples within class level, the discriminator is first updated based on Eq. (4). Next, we use the updated discriminator to compute the feature adaptive loss based on Eq. (3) to encourage the generation of class-conditional invariant features. The model G is then updated based on the feature adaptive loss combined with original task loss, as described in Eq. (5). This training routine can effectively minimize the discrepancy between natural data and adversarial examples while maximizing the class-level discrepancy. Consequently, the overall robustness and natural accuracy of the trained model can be improved. Note that during the inference phase, the equipped multi-class discriminator can be removed, and the trained model can be robust to adversarial examples.

IV. EXPERIMENTS

In this section, we present the experimental results of our proposed FAAT for improving the model robustness. We first describe the experimental settings including training details and compared baselines used in the experiments in Sec. IV-A. Then the comparison with state-of-the-art AT methods in terms of both clean and robust accuracy is presented in Sec. IV-B, with a visualization of feature distribution for a more intuitive comparison in Sec. IV-C. We also conduct the efficiency analysis to demonstrate the high potential for social impact of FAAT in Sec. IV-D. To identify the core components of our FAAT, Sec. IV-E gives two variants of FAAT and reports the quantitative results.

A. Experimental setup

1) Training details: We implement all our experiments using PyTorch on an Intel Core i9 with 48GB of memory and an NVIDIA RTX A6000 GPU. In our experiments, we alternatively update the discriminator and CNN model in each epoch. For the hyper-parameters setting, we generally follow the settings suggested by previous works [9], [11]. Specifically, we use PGD-10 algorithm to generate adversarial examples in each training epoch, where perturbation constraint $\epsilon = 8/255$, step size is set as $2\epsilon/10$. We train the discriminator and CNN model for 100 epochs using stochastic gradient descent (SGD) optimizer with an initial learning rate as 0.1 and it is decay by a factor of 0.1 at 50% and 75% of the total epoch.

2) Compared methods: We compare the performance of our method with various counterparts, including standard adversarial training method [7] which uses PGD-10 to generate online adversarial examples; methods focusing on maximizing the strength of adversarial examples for improving robust accuracy: ATTA [8] and CCG [47]; and methods focusing on achieving trade-off between natural accuracy and robust accuracy: ALP [10] and TRADES [9]. For TRADES, we set $\lambda = 4$ for comparison. Following previous works, we use WideResNet-34-10 as the trained CNN model architecture for CIFAR-10 and CIFAR-100 datasets, and PreActResNet-18 for TinyImageNet throughout all our experiments.

TABLE III
NATURAL AND ROBUST ACCURACY(%) OF WIDERESNET-34-10 TRAINED WITH l_∞ OF $\epsilon = 8/255$ BOUNDARY AGAINST UNSEEN ATTACKS ON CIFAR-10 DATASET. FOR UNSEEN ATTACKS, WE USE PGD-50 UNDER DIFFERENT SIZED l_∞ BALLS, AND OTHER TYPES OF NORM BALL, E.G., l_2, l_1.

Dataset	Method	l_∞		l_2		l_1	
		4/255	16/255	150/255	300/255	2000/255	4000/255
CIFAR-10	Standard	67.92	21.52	52.49	24.93	67.36	46.99
	+Ours	**73.26**	**22.68**	**58.25**	**25.61**	**70.59**	**50.16**
	ALP [40]	72.76	22.98	56.96	24.70	69.53	48.77
	+Ours	71.49	**26.44**	**59.96**	**30.10**	**69.56**	**52.71**
	TRADES ($\lambda = 4$) [9]	72.25	24.12	58.71	27.96	69.41	51.18
	+Ours	71.21	**24.81**	**58.73**	**28.99**	**69.51**	**52.01**
	CCG [47]	73.46	23.86	58.13	28.98	72.81	54.16
	+Ours	**73.55**	22.97	**58.83**	25.84	71.17	50.45
CIFAR-100	Standard	40.67	9.96	30.69	12.99	42.43	28.24
	+Ours	**46.01**	**11.70**	**35.35**	**15.55**	**46.64**	**31.87**
	ALP [10]	46.58	15.34	36.51	18.43	45.94	33.07
	+Ours	46.11	**17.25**	**38.33**	**21.43**	45.74	**34.88**
	TRADES ($\lambda = 6$) [9]	42.49	12.97	31.94	14.78	42.00	28.22
	+Ours	**43.01**	**16.99**	**35.42**	**19.84**	**42.54**	**31.80**
	CCG [47]	42.47	9.37	32.63	13.54	45.09	30.85
	+Ours	**46.51**	**11.96**	**35.75**	**14.93**	**46.50**	**31.60**

TABLE IV
NATURAL AND ROBUST ACCURACY (%) OF WIDERESNET34-10 MODELS TRAINED ON CIFAR-100 DATASET AGAINST BLACK-BOX TRANSFER ATTACK.WE CHOOSE l_∞ THREAT MODEL WITH $\epsilon = 8/255$ FOR PGD. SPECIALLY, FOR CW_2, ϵ IS FIXED TO $160/255$.

Method	Natural	PGD-50	CW_∞	FGSM-ILA [21]	CW_∞-ILA [21]
Standard	60.29	42.13	56.55	43.08	55.18
+ATTA [8]	55.09	39.79	50.74	40.86	49.86
+CCG [47]	60.74	**46.74**	58.92	**46.84**	56.55
+ALP [10]	59.75	45.79	56.94	46.12	55.85
+TRADES($\lambda = 6$) [9]	62.37	42.09	53.86	42.64	53.00
+Ours	**62.58**	44.65	**59.47**	45.50	**58.13**

TABLE V
NATURAL AND ADVERSARIAL ROBUST ACCURACY (%) OF THE WIDERESNET34-10 MODELS TRAINED BY COMBINATIONS OF ADVERSARIAL TRAINING METHODS.

Method	CIFAR-10		CIFAR-100	
	Natural	PGD-20	Natural	PGD-20
ATTA	84.43	54.65	55.09	23.23
ATTA+Ours	**87.33**	53.84	**59.01**	**25.32**
TRADES($\lambda = 6$)	85.05	51.20	62.37	25.31
TRADES+Ours	**87.29**	**55.77**	60.51	**28.18**
ALP	86.45	53.12	59.65	28.28
ALP+Ours	84.20	**56.33**	**61.35**	**29.58**
CCG	88.32	54.71	60.74	27.27
CCG+Ours	87.31	**55.14**	**63.69**	**30.95**

B. Main results

1) White-box attacks: We first evaluate the robustness of our proposed method under white-box attacks where the adversarial examples are generated by the known model. Here we use three attack methodologies: Common attacks including PGD-k and CW-k, and stronger attack AA. We also generate adversarial examples with more strong attack, Auto-Attack (AA), which is known as the strongest attack by far. As we can see in Table II, focusing on strengthening the adversarial attack in ATTA can significantly improve the robustness accuracy but also lead to a decrease in natural accuracy on CIFAR-10 dataset. CCG uses specific designed data augmentation to increase the data diversity so it achieves higher natural and robust accuracy in CIFAR-10 dataset. However, both the ATTA and CCG methods degrade in CIFAR-100 dataset. For the methods designed for trade-off terms, adding TRADES or ALP into the standard adversarial training can somewhat maintain the trade-off between natural accuracy and robust accuracy, but neither of them can achieve overall improvements over the standard training. Instead, our method focus on class-conditional feature adaption across natural data and adversarial examples, and can effectively enhance both natural and robustness accuracy, especially for CIFAR-100 and TinyImageNet datasets with more classes, as the natural and robust accuracy against AA on CIFAR-100 dataset achieves 62.58% and 27.87%, respectively, surpassing baselines with a large margin.

2) Unseen adversaries: We consider a wide range of unforeseen adversaries, e.g., robustness on different attack threat radii ϵ, or even on different norm constraints (e.g., l_2 and l_1), in order to further measure the robustness of trained models against multiple perturbation. The results are reported in Table III. As we can see, incorporating FAAT into adversarial training can remarkably improve the robustness against unseen attacks. CCG improves the model robustness

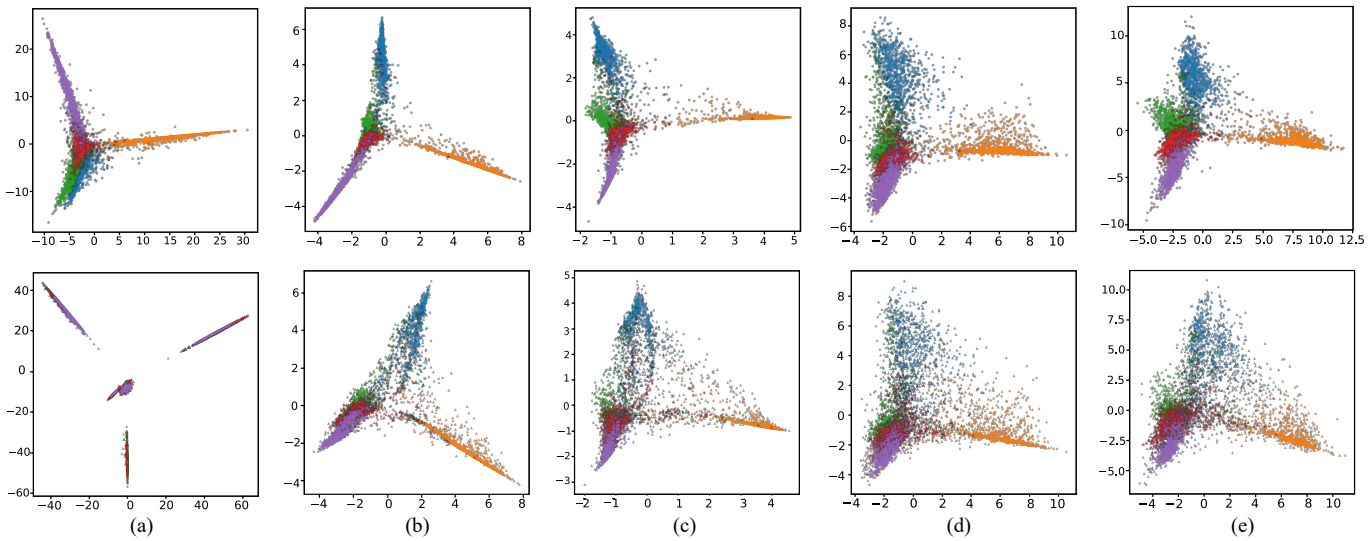

Fig. 4. Visualization of feature distribution in WideResNet-34-10 model trained with different methods on CIFAR-10 dataset. The upper row denotes the natural examples and the bottom row denotes the adversarial examples. Feature distribution in model trained with: (a) Natural training; (b) standard adversarial training; (c) ALP; (d) TRADES; and (e) our FAAT.

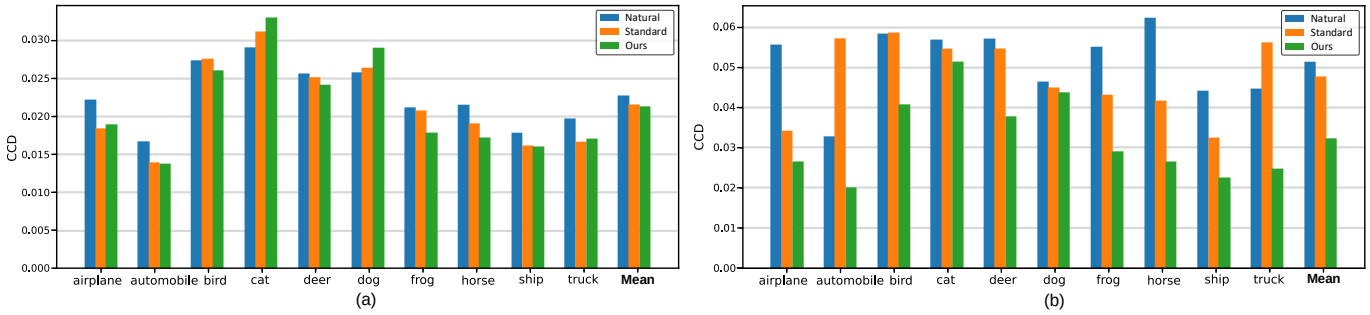

Fig. 5. CCD analysis of the feature distributions produced by models trained on CIFAR-10 dataset. We show the CCD value using Eq. (7) for each class and overall mean value on (a) natural data and (b) adversarial examples.

via augmenting additional data to increase the diversity of the dataset. As our FAAT encourages similar distribution across natural data and adversarial examples, it will weaken the augmentation strength, thus leading to a slight decrease under some attack conditions on CIFAR-10 dataset. Fortunately, we take the class knowledge into consideration in FAAT, which comes to play an important role when the the scale of dataset becomes larger. For CIFAR-100 dataset, with the class-conditional feature adaption scheme, our proposed FAAT can significantly help the existing methods improve the adversarial robustness against unseen attacks. Especially for CCG method, the CCG+Ours outperforms CCG under every attack scenario, which makes us believe the proposed FAAT has more potential for model robustness improvements on large-scale datasets which exist in more practical applications. Although ALP and TRADES methods share the same motivation with us that encourages the similar distribution of natural and adversarial data, incorporating FAAT still brings improvements under most attack scenarios which is benefited from the fine-grained feature adaption.

3) Black-box transfer attacks: To evaluate the effectiveness of our proposed FAAT in more practical defense scenarios, we test the model under black-box transfer attacks, where adversarial examples are generated from a source model and then transfer to the target model. In this experiment, adversarial examples are crafted from PreActResNet18 (source model) trained with standard adversarial training (PGD-10), and we use PGD-50 and CW_∞ as black-box adversaries. Furthermore, we also evaluate the defense performance against enhanced black-box attacks by applying [21] with intermediate level attack (ILA). The results are shown in Table IV. It is obvious that the black-box attacks are weaker than white-box attacks, as the robust accuracy denoted in Table IV are double times higher than that against white-box attacks in Table II on the same dataset. From the comparison with other methods, we can see that our method can achieve comparable results with CCG which applied data augmentation in their method on both common and enhanced black-box attacks. In addition, our method shows consistent improvement in natural accuracy by taking trade-off between natural and robust accuracy into consideration.

4) Combing with other methods: Our FAAT provides an universal framework for adversarial training. To verify the flexibility of our method, we combine various methods with our FAAT framework. The evaluation is under the white-box attack following the same setting in Sec. III-D. The results

TABLE VI
TRAINING TIME (S) IN EACH EPOCH FOR DIFFERENT ADVERSARIAL TRAINING METHODS.

Dataset	Standard	ATTA [8]	CCG [47]	ALP [10]	TRADES [9]	Ours
CIFAR-10	1235.10	1425.79	3556.23	2033.20	2199.24	1440.12
CIFAR-100	1234.11	1429.49	3567.37	2042.35	2205.41	1445.24

are summarized in Table V. On CIFAR-10 dataset, using our FAAT framework can effectively improve the existing adversarial methods on the trade-off between natural and robust accuracy. For ATTA which focuses on enhancing attack strength, equipped with our FAAT framework, the ATTA method can improve the natural accuracy by 3%. For TRADES and ALP, incorporating FAAT can improve the robustness accuracy while getting rid of a severe decrease in natural accuracy. On CIFAR-100 dataset, FAAT framework can help the existing methods improve the overall performance by incorporating the class-conditional feature adaption into the training process. Particularly, our FAAT framework improves CCG by 3% in both natural and robust accuracy, which creates new SOTA on CIFAR-100 dataset.

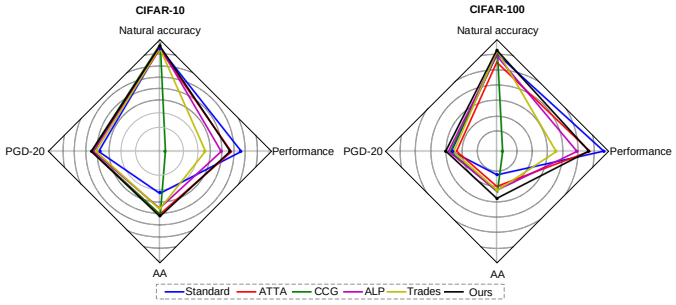

Fig. 6. Illustration of natural accuracy, robust accuracy against PGD-20, robust accuracy against AA, and performance trade-offs using different adversarial training methods.

C. Feature distribution

To verify whether incorporating such a discriminator in adversarial training can improve the feature learning of CNN models, we design an experiment to investigate the feature distribution before and after our FAAT. We intuitively show the feature distribution of natural data and adversarial examples in models trained with different approaches using t-SNE embedding [14] on CIFAR-10 dataset in Fig. 4. Here we generate the adversarial examples by PGD-20 attack, and the features are extracted from the last convolution layer before normalization. As we can observe, in natural training of Fig. 4 (a), the adversarial data reside in the same region which is hard to distinguish, thus leading to a very low robustness accuracy. By taking adversarial training, both natural and adversarial data distribution learned by ALP has much larger intra-class distance and small inter-class distance, which will lead to lower accuracy in natural and adversarial data, as shown in Fig. 4 (b). ALP and TRADES enlarge the inter-class distance and make the decision boundary easier to decide. As we incorporate the class knowledge into trade-off balance, our proposed FAAT demonstrates the highest intra-class compactness and largest inter-class margin, as can be seen in Fig. 4 (e). Moreover, the features from natural data and adversarial examples have similar distributions. This suggests that our FAAT can generate more class-discriminative but invariant features against adversarial attacks, which is consistent with our motivation and the results in Table II.

Moreover, we use the Class Center Distance (CCD) suggested in [12] as a metric to evaluate the feature distribution by computing intra-class compactness over inter-class distance. The CCD for class k can be represented as:

$$CCD(k) = \frac{1}{K-1} \sum_{i=1, i\neq k}^{K-1} \frac{\frac{1}{|S_k|} \sum_{\mathbf{f} \in S_k} \|\mathbf{f} - \mu_i\|^2}{\|\mu_k - \mu_i\|^2}, \quad (7)$$

where μ_k is the class center of class k, and S_k denotes the set of all features belonging to class k. Note that a lower CCD value suggests a more densely clustering. The results are provided in Fig. 5. As expected, our FAAT achieves a much lower CCD on most classes and the lowest mean CCD value on both natural data and adversarial examples. It indicates that FAAT can achieve better class-level feature adaption across natural data and adversarial examples.

D. Efficiency analysis

A bottleneck of adversarial training is the expensive training time due to the online data augmentation process. In this part, we analyze the efficiency of our proposed FAAT. Compared to standard adversarial training, our FAAT additionally equips a discriminator. Since we use the feature extractor in the target model to extract features thus the discriminator architecture is simple, and training such a discriminator only takes trivial additional consumption cost. We report the training time for each epoch of different methods in Table VI. As we can see, the training of FAAT is faster than TRADES or ALP, and much faster than CCG which needs to deliberately augment additional training data.

To better understand the trade-offs among natural accuracy, robust accuracy, and efficiency, Fig. 6 demonstrates a four-dimensional radar plot with natural accuracy, robust accuracy against PGD-20, robust accuracy against AA, and performance on four axes. Note that we plot the negative of the running time for each training epoch on the performance axis. Thus, the ideal adversarial training method exhibits highly rhombus-shape quadrangle. We find that our FAAT (denoted by the black solid quadrangle) gives a better trade-off on efficiency and effectiveness on adversarial training. Hence, we believe the FAAT can provide an universal and efficient framework for adversarial training while free of burden.

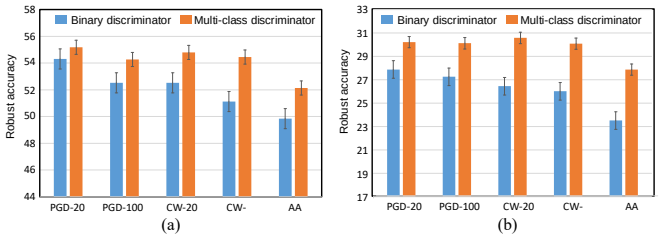

Fig. 7. Comparison of using binary discriminator and multi-class discriminator in FAAT. We use WideResNet-34-10 as the task model trained on (a) CIFAR-10 dataset and (b) CIFAR-100 dataset.

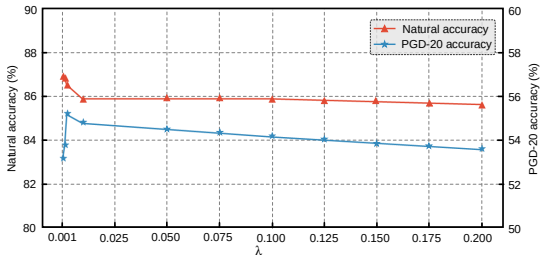

Fig. 8. The curve of natural accuracy and robust accuracy of WideResNet-34-10 trained on CIFAR-10 dataset using FAAT with different values of λ.

E. Ablation studies

1) Effect of class knowledge encoding: We compare FAAT using the class-conditional discriminator with that uses the traditional binary discriminator, to verify the merits of introducing class-level knowledge in adversarial training. The robustness performance of model trained on CIFAR-10 and CIFAR-100 datasets are shown in Fig. 7. It is obvious that incorporating the class-level knowledge into feature adaption can effectively improve the adversarial training performance, especially for CIFAR-100 dataset with more classes. We believe the reason is that the incorporation of class-level clustering helps improve the overall performance in model robustness.

2) Hyper-parameter sensitivity: We study the sensitivity of FAAT to the balance weight λ on CIFAR-10 dataset. Generally, we follow the setting in [12] to initialize $\lambda = 0.001$ and test the values floating up or down. The results are provided in Fig. 8. As λ gets larger, the overall accuracy steadily increases before decreasing. We set $\lambda = 0.0015$ throughout the experiments as the setting denotes the best performance.

V. Conclusion

In this paper, we focus on maintaining the trade-off between natural accuracy and robust accuracy in adversarial training by proposing a class-conditional feature adaptive adversarial training framework. A class-conditional discriminator is incorporated in adversarial training to guide the feature adaption across natural data and adversarial examples. Comprehensive experiments and analysis validate the effectiveness of our FAAT, where our method achieves the best robust accuracy in AA while maintaining the natural accuracy.

References

[1] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1369–1378.

[2] A. Sayles, A. Hooda, M. Gupta, R. Chatterjee, and E. Fernandes, "Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2021, pp. 14 666–14 675.

[3] W. Ding, X. Wei, R. Ji, X. Hong, Q. Tian, and Y. Gong, "Beyond universal person re-identification attack," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 3442–3455, 2021.

[4] T. Wu, Z. Liu, Q. Huang, Y. Wang, and D. Lin, "Adversarial robustness under long-tailed distribution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2021, pp. 8659–8668.

[5] Y. Zhong and W. Deng, "Towards transferable adversarial attack against deep face recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1452–1466, 2021.

[6] X.-C. Li, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Decision-based adversarial attack with frequency mixup," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1038–1052, 2022.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int'l Conf. Learn. Repres.*, 2018.

[8] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash, "Efficient adversarial training with transferable adversarial examples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1178–1187.

[9] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. Int'l Conf. Machine Learn.*, 2019, pp. 12 907–12 929.

[10] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," *arXiv preprint arXiv: 1803.06373*, 2018.

[11] J. Cui, S. Liu, L. Wang, and J. Jia, "Learnable boundary guided adversarial training," in *Proc. IEEE Int'l Conf. Comput. Vis.*, October 2021, pp. 15 721–15 730.

[12] H. Wang, T. Shen, W. Zhang, L. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *Proc. Euro. Conf. Comput. Vis.*, August 2020.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int'l Conf. Machine Learn.*, 2015, p. 1180–1189.

[14] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4893–4902.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int'l Conf. Learn. Repres.*, 2014.

[16] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int'l Conf. Learn. Repres.*, 2015.

[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. & Priv.*, May 2017, pp. 39–57.

[18] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. Int'l Conf. Machine Learn.*, 2020.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int'l Conf. Learn. Repres.*, 2014.

[20] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. Int'l Conf. Learn. Repres.*, 2017.

[21] Q. Huang, I. Katsman, Z. Gu, H. He, S. Belongie, and S.-N. Lim, "Enhancing adversarial example transferability with an intermediate level attack," in *Proc. IEEE Int'l Conf. Comput. Vis.*, 2019, pp. 4732–4741.

[22] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.

[23] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2725–2734.

[24] W. Wu, Y. Su, X. Chen, S. Zhao, I. King, M. R. Lyu, and Y.-W. Tai, "Boosting the transferability of adversarial samples via attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2020, pp. 1158–1167.

[25] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2021, pp. 1924–1933.

[26] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," in *Proc. Int'l Conf. Learn. Repres.*, 2017.
[27] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. Int'l Conf. Learn. Repres.*, 2018.
[28] B. Huang, Y. Wang, and W. Wang, "Model-agnostic adversarial detection by random perturbations," in *Proc. Int'l Joint Conf. Artif. Intell.*, Macao, China, 2019, pp. 4689–4696.
[29] S. Wang, S. Nepal, A. Abuadbba, C. Rudolph, and M. Grobler, "Adversarial detection by latent style transformations," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1099–1114, 2022.
[30] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1778–1787.
[31] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *Proc. Adv. Neural Inform. Process. Syst.*, 2019, pp. 3358–3369.
[32] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *Proc. Int'l Conf. Learn. Repres.*, Addis Ababa, Ethiopia, 2020.
[33] F. Tramer, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int'l Conf. Learn. Repres.*, 2018.
[34] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *Proc. Int'l Conf. Machine Learn.*, 2020.
[35] Q.-Z. Cai, C. Liu, and D. Song, "Curriculum adversarial training," in *Proc. Int'l Joint Conf. Artif. Intell.*, Stockholm, Sweden, 2018, pp. 3740–3747.
[36] J. Liu, C. P. Lau, H. Souri, S. Feizi, and R. Chellappa, "Mutual adversarial training: Learning together is better than going alone," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2364–2377, 2022.
[37] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *Proc. Int'l Conf. Machine Learn.*, 2019, pp. 8759–8771.
[38] S. Kariyappa and M. K. Qureshi, "Improving adversarial robustness of ensembles with diversity training," *arXiv preprint arXiv: 1901.09981*, 2019.
[39] H. Yang, J. Zhang, H. Dong, N. Inkawhich, A. Gardner, A. Touchet, W. Wilkes, H. Berry, and H. Li, "Dverge: Diversifying vulnerabilities for enhanced robust generation of ensembles," in *Proc. Adv. Neural Inform. Process. Syst.*, 2020.
[40] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *Proc. Int'l Conf. Learn. Repres.*, 2020.
[41] T. Wang, R. Zhang, X. Chen, K. Zhao, X. Huang, Y. Huang, S. Li, J. Li, and F. Huang, "Adaptive feature alignment for adversarial training," *arXiv preprint arXiv: 2105.15157*, 2021.
[42] H. Wang, T. Chen, S. Gui, T.-K. Hu, J. Liu, and Z. Wang, "Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free," in *Proc. Adv. Neural Inform. Process. Syst.*, 10 2020.
[43] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2020.
[44] M. Levi, I. Attias, and A. Kontorovich, "Domain invariant adversarial learning," *arXiv preprint arXiv: 2104.00322*, 2021.
[45] P. Bashivan, R. Bayat, A. Ibrahim, K. Ahuja, M. Faramarzi, T. Laleh, B. A. Richards, and I. Rish, "Adversarial feature desensitization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2021.
[46] Z. Qian, S. Zhang, K. Huang, Q. Wang, R. Zhang, and X. Yi, "Improving model robustness with latent distribution locally and globally," *arXiv preprint arXiv: 2107.04401*, 2021.
[47] J. Tack, S. Yu, J. Jeong, M. Kim, S. J. Hwang, and J. Shin, "Consistency regularization for adversarial robustness," in *Proc. Int'l Conf. Machine Learn.*, 2021.