Mitigating Feature Gap for Adversarial Robustness by Feature Disentanglement

Nuoyan Zhou[1*], **Dawei Zhou**[1*], **Decheng Liu**[1], **Nannan Wang**[1†], **Xinbo Gao**[2]

[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China
[2]Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing, China
nuoyanzhou@stu.xidian.edu.cn, dwzhou.xidian@gmail.com, {dchliu, nnwang}@xidian.edu.cn, gaoxb@cqupt.edu.cn

Abstract

Adversarial fine-tuning methods enhance adversarial robustness via fine-tuning the pre-trained model in an adversarial training manner. However, we identify that some specific latent features of adversarial samples are confused by adversarial perturbation and lead to an unexpectedly increasing gap between features in the last hidden layer of natural and adversarial samples. To address this issue, we propose a disentanglement-based approach to explicitly model and further remove the specific latent features. We introduce a feature disentangler to separate out the specific latent features from the features of the adversarial samples, thereby boosting robustness by eliminating the specific latent features. Besides, we align clean features in the pre-trained model with features of adversarial samples in the fine-tuned model, to benefit from the intrinsic features of natural samples. Empirical evaluations on three benchmark datasets demonstrate that our approach surpasses existing adversarial fine-tuning methods and adversarial training baselines.

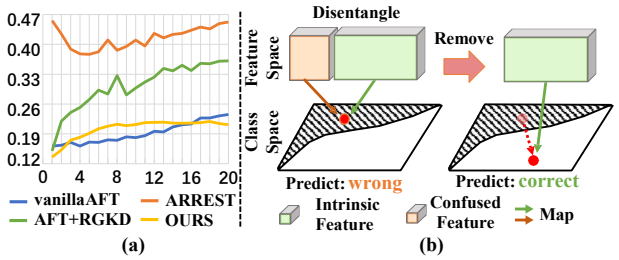


Figure 1: Illustration of our motivation. (a) L_∞ distances of features between natural and adversarial samples during fine-tuning. Previous AFT methods exhibit an increasing trend of the feature gap. (b) Toy illustration of Disentanglement. We model features of adversarial samples and then remove the specific latent features confused by adversarial perturbation to correct the prediction.

Introduction

Deep neural networks (DNNs) have shown impressive performances in various domains of machine learning. However, it has been demonstrated that DNNs are susceptible to adversarial samples and the prediction can be easily manipulated (Goodfellow, Shlens, and Szegedy 2014). Adversarial samples deceive DNNs by introducing imperceptible noise to clean samples. The presence of adversarial samples poses a growing potential threat (Madry et al. 2017; Croce and Hein 2020; Yu, Gao, and Xu 2021; Wei et al. 2022, 2023; Liu et al. 2024; Xia et al. 2024), making it crucial to enhance the robustness of networks.

Many defensive techniques are proposed including adversarial training (AT) (Madry et al. 2017; Zhang et al. 2019; Wang et al. 2020; Wu, Xia, and Wang 2020; Huang, Zhang, and Zhang 2020; Dong et al. 2021; Jia et al. 2022; Jin et al. 2022; Yu et al. 2022; Zhou et al. 2022), adversarial fine-tuning (AFT) (Suzuki et al. 2023; Maini, Wong, and Kolter 2020; Madaan, Shin, and Hwang 2021; Croce and Hein 2022; Jeddi, Shafiee, and Wong 2020; Zhu et al. 2023; Si et al. 2020; Agarwal et al. 2023; Liu et al. 2023), adversarial purification (Yoon, Hwang, and Lee 2021; Sun et al. 2023; Lee and Kim 2023; Allen-Zhu and Li 2022; Nie et al. 2022), adversarial detection (Hickling, Aouf, and Spencer 2023; Zheng and Hong 2018; Pang et al. 2018), etc. AT is regarded as the most effective defensive technique among them, but it suffers from several times the training cost as standard (natural) training. To save the cost of training time, AFT employs the same loss function as AT to fine-tune the pre-trained model within a few epochs, which has aroused wide attention.

We identify an issue of the increasing feature gap by analyzing three models fine-tuned by three AFT methods: vanilla AFT (Jeddi, Shafiee, and Wong 2020), AFT+RGKD, ARREST (Suzuki et al. 2023). As illustrated in Figure 1 (a), the feature distance between natural and adversarial samples abnormally maintains an increasing direction during the fine-tuning process. Though ARREST initially exhibits notable fluctuations with a decreasing trend, the overall magnitudes of feature distance in three AFT methods show an increasing trend. We suppose that some specific latent features are easily confused by adversarial attacks and lead to the growing gap between the features of natural and adversarial samples. Considering that a robust model should treat natural and adversarial samples equally and extract similar

*These authors contributed equally.
†Corresponding author.

features from them, we expect to remove the specific latent features causing the feature gap to achieve better robust performance, as shown in Figure 1 (b).

In this paper, we analyze the specific latent features causing the feature gap and introduce an approach based on disentanglement to explicitly model and remove these features. We propose a new technique named **A**dversarial **F**ine-tuning via **D**isentanglement (**AFD**). First, we disentangle the features of adversarial samples as two components, in which the specific latent features causing the feature gap are identified as the features confused by adversarial perturbation. Next, we propose a feature disentangler to model and separate them from the features of adversarial samples. We maximize the predicted probabilities of the wrong predicted classes to acquire the specific latent features. Then we impose a constraint to distance the features of adversarial samples from them, to eliminate the specific latent features and mitigate the feature gap. Besides, we align features of adversarial samples in the fine-tuned model with features of natural samples in the naturally pre-trained model, further eliminating the specific latent features. Experiments demonstrate that our AFD alleviates the feature gap and improves the robustness. Contributions are summarized as follows:

- We observe the gap in features between natural and adversarial samples anomalously increases in adversarial fine-tuning methods, resulting from the specific latent features confused by adversarial perturbation.
- We provide the theoretical analysis of feature disentanglement. Then we introduce a disentanglement-based AFT approach, which explicitly models and further eliminates the specific latent features causing the feature gap by disentanglement and alignment.
- Empirical evaluations show our approach mitigates the gap in features between natural and adversarial samples and surpasses existing methods. We also provide extended analyses for a holistic understanding.

Related Work

Adversarial Attack Adversarial Attack is first introduced in Goodfellow, Shlens, and Szegedy (2014), which fools DNNs by imperceptible perturbations. Madry et al. (2017) propose Projected Gradient Descent (PGD) attack based on the gradient projection and random start. Croce and Hein (2020) introduce a powerful adaptive attack (AutoAttack), including three white-box attacks and a black-box attack. Yu, Gao, and Xu (2021) demonstrate that exploiting latent features is highly effective against many defense techniques. Besides, adversarial attacks can also work in the physical world. Wei et al. (2022) present a comprehensive overview of physical adversarial attacks. Wei et al. (2023) propose a novel physical adversarial attack that applies the Warming Paste and Cooling Paste to hide persons from being detected. Liu et al. (2024) learn adversarial semantic perturbations in the latent space for high attack capabilities and low perceptibility. We take PGD and AutoAttack to evaluate robustness.

Adversarial Training Adversarial training (AT) is the most effective technique to defend against attacks. Madry et al. (2017) propose PGD-based adversarial training (PGD-AT), compelling the model to correctly classify adversarial samples within the epsilon sphere during training. Zhang et al. (2019) reduce the divergence of probability distributions of natural and adversarial samples to mitigate the difference between robust and natural accuracy. Xie et al. (2019) apply non-local means or other filters as the denoising block. Wang et al. (2020) find that misclassified samples harm adversarial robustness significantly, and propose to improve the model's attention to misclassification by adaptive weights. Yan et al. (2021) manipulates channels' activations to align the channel with the relevance to predictions. Yang et al. (2021) introduce a disentanglement-based architecture to generate class-specific and class-irrelevant representations. Zhou et al. (2022) embed a label transition matrix into models to infer natural labels from adversarial noise. Kim et al. (2023) propose the FSR module to separate non-robust activations and recalibrate the features. Zhou et al. (2023) define two attributes of robust features to guide robust training. Though AT has good robust performances, it suffers from a large quantity of computing expenses. Some work has been developed to accelerate adversarial training, commonly known as fast adversarial training (FAT) (Wong, Rice, and Kolter 2020; Kim, Lee, and Lee 2020; Huang et al. 2023). Since these FAT methods only require a little training time like AFT, we compare them with our method.

Adversarial Fine-tuning Adversarial fine-tuning (AFT) has been employed to enhance the pre-trained model within a few epochs for various targets, particularly for achieving adversarial robustness. Moosavi-Dezfooli et al. (2018) analyze the effect of AFT by comparing the decision boundary of a DNN before and after applying AFT. Kumari et al. (2019); Yu, Gao, and Xu (2021) discover the vulnerability of latent features in adversarial trained models and propose adversarial approaches to mitigate this. Jeddi, Shafiee, and Wong (2020) employ vanilla AFT to expedite the training of robust models with a few training epochs and a warm-up learning rate schedule. Some researchers (Tramer and Boneh 2019; Maini, Wong, and Kolter 2020; Madaan, Shin, and Hwang 2021; Croce and Hein 2022) aim to achieve multiple l_p adversarial robustness through specific strategies. Based on representation learning, Suzuki et al. (2023) constrain the representation to mitigate the accuracy-robustness trade-off. **Although they aim to learn suitable features, the gap between natural and adversarial features can't be eliminated by simple feature constraint, which has been indicated in Figure 1 (a)**. Zhu et al. (2023) propose a metric to measure the robustness of modules and fine-tune them for better out-of-distribution performance. Liu et al. (2023) introduce a two-pipeline structure to improve the relationship between the weight norm and its gradient norm in batch normalization layers. Despite the extensive work, few methods (vanilla AFT (Jeddi, Shafiee, and Wong 2020), ARREST and AFT+RGKD (Suzuki et al. 2023)) are based on naturally pre-trained models, which are our important baselines.

Preliminaries

We denote a DNN as $h(x) = \omega \cdot g(x; \theta_g)$, where x is the input image, and θ_g is the parameters of the feature ex-

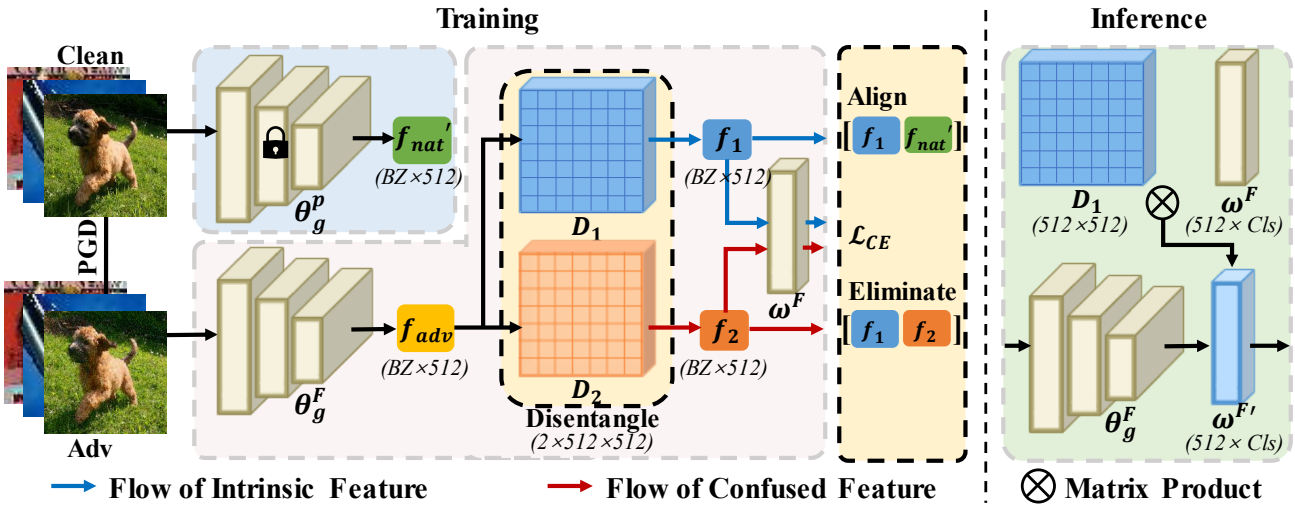


Figure 2: Overview of our AFD. Lock denotes the frozen parameters, BZ denotes the batch size, Cls denotes the class number, and numbers in brackets denote the sizes of features or parameters. During fine-tuning, Adversarial features are disentangled into f_1 and f_2. $\mathcal{L}_{CE}$ ensures f_2 to approximate confused features. We keep f_1 away from f_2 to eliminate confused features in f_1. Besides, we align f_1 with pre-trained intrinsic features f'_{nat} to further correct the prediction. During inference, we multiply D_1 and ω^F to obtain the robust model without additional modules.

tractor g, i.e., the model before the last linear classifier. ω denotes the parameters of the last linear classifier. The parameters of the naturally pre-trained and fine-tuned model are denoted as θ^P and θ^F, respectively. The model parameters θ^F are initialized by θ^P. x denotes the clean sample, and x' denotes the adversarial sample. $\mathbb{N}(x, \epsilon)$ represents $\{x' : \|x' - x\|_\infty \leq \epsilon\}$, ϵ is the perturbation budget, $\|\cdot\|_p$ denotes the l_p norm. Adversarial and natural features denote the features in the last hidden layer of adversarial and natural samples, i.e., $g(x'; \theta_g)$ and $g(x; \theta_g)$ respectively.

Methodology

In this section, we first reveal the issue of the increasing gap of features between natural and adversarial samples in AFT. Then we introduce a disentanglement-based method, to model and further remove the specific latent features leading to the feature gap. Our method is expected to mitigate the feature gap to enhance adversarial robustness.

Specific Latent Feature Causing Feature Gap

As shown in Figure 1 (a), there is a growing gap in features between natural and adversarial samples in existing AFT methods. Intuitively, the ideal robust model should have a consistent understanding and analysis of natural and adversarial samples, thus extracting similar (the same) features from them. Thus, narrowing the feature gap is expected to achieve better robust performance.

To realize the target, we can disentangle two types of features from the features of natural and adversarial samples. We define natural features f_{nat} as the intrinsic features f_i. We can define adversarial features f_{adv} as the combination of the intrinsic features f_i and the confused features f_c, the latter of which leads to the gap in features between natural and adversarial samples. In this modeling, the intrinsic features represent the features that contribute to accurate classification, which are easily extracted from natural samples. Besides, the confused features are identified as the specific latent features causing the gap between features of natural and adversarial samples, which are confused by the adversarial perturbation. As illustrated in Figure 1 (b), if the confused features are removed in adversarial features, the features of adversarial samples can align with those of natural samples, resulting in a zero feature gap and accurate predictions for adversarial samples.

We suppose the phenomenon in Figure 1 (a) is attributable to the presence of the confused features f_c. Because the confused features persist as latent features and don't diminish during fine-tuning, the gap between features of natural and adversarial samples cannot be bridged. The confused features induce the gap between natural and adversarial features, leading to limited robust performance. Based on our observations and analyses, we introduce a new approach named **A**dversarial **F**ine-tuning via **D**isentanglement (**AFD**), as shown in Figure 2. It applies a feature disentangler to explicitly extract the confused features, and remove them by the elimination and alignment of features.

Adversarial Fine-tuning Based on Disentanglement

The primary strategy is to explicitly disentangle the intrinsic and confused features from the features of adversarial samples. We propose a feature disentangler to model confused features for further elimination. The features in the last hidden layer are input into the disentangler, yielding two disentangled feature vectors. We define these feature vectors as intrinsic features f_1 and confused features f_2. To implement the disentanglement, we introduce a disentangler with two linear blocks D_1, D_2 between the feature extractor g and

the linear classifier ω. These linear blocks are expected to disentangle adversarial features into intrinsic and confused features. The disentangled feature vectors f_1 and f_2 are constrained to approximate the intrinsic and confused features. Given by:

$$\begin{aligned} f_1 &= D_1(g(x';\theta_g^F)), \\ f_2 &= D_2(g(x';\theta_g^F)), \\ \text{where } &\begin{cases} f_1 \approx f_i, \\ f_2 \approx f_c, \end{cases} \end{aligned} \tag{1}$$

where θ_g^F denotes the parameters of the feature extractor in the fine-tuned model. x' denotes the adversarial samples created by PGD-10 (Madry et al. 2017), formulated as:

$$x' = \underset{\mathbb{N}(x,\epsilon)}{\mathbf{\Pi}} \left(x + \epsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}_{CE}\left(\omega^F \cdot D_1(g(x;\theta_g^F)), y \right) \right) \right), \tag{2}$$

where $\mathbf{\Pi}$ denotes a clamp function, sign denotes a sign function, $\mathcal{L}_{CE}$ denotes the cross-entropy loss. We take f_1 and f_2 to match the intrinsic and confused features. The input and output dimensions of D_1, D_2 are set to match the channel dimensions of the feature vectors before disentanglement. This ensures the parameters of the linear block θ_{D_1} can be seamlessly integrated with the parameter of the last linear classifier ω^F, expressed as $\theta_{D_1} \cdot \omega^F = \omega^{F\prime}$. Therefore, we only need to load the linear classifier $\omega^{F\prime}$ without additional disentangling modules in the inference (test) phase.

To achieve good performance on both accuracy and robustness, we should design suitable loss functions to realize Formula 1. We suggest fitting f_2 to incorrectly predicted labels y' to model the confused features. The confused features result in obvious shifts in both features and predicted probabilities, leading to wrong predicted class labels in most cases. Thus, it can be identified as the specific latent features causing the gap between natural and adversarial features. To decrease the dependence between the confused features and ground-truth labels, we force the disentangled features f_2 to match incorrect class labels, formulated as:

$$\begin{aligned} \mathcal{L}_1 &= \mathcal{L}_{CE}(\omega^F \cdot f_2, y'), \\ \text{where} \quad y' &= \max_{\mathrm{y}' \neq \mathrm{y}} (\omega^F \cdot f_1), \end{aligned} \tag{3}$$

where y' denotes wrong predicted labels with the maximum predicted probability. We only optimize the disentangling block D_2 instead of the entire model. This selective optimization prevents potential harm to the feature extractor that could result from learning the confused features.

In addition, we introduce a constraint on f_1 to keep away from the confused features f_2, removing the confused features from the features of adversarial samples. The exclusive force between f_1 and f_2 encourages the elimination of the confused features from the features of adversarial samples. This constraint aims to protect adversarial features from being perturbed by the confused features, and reduces the association between confused and intrinsic features. Given by:

$$\mathcal{L}_2 = -\mathcal{D}(f_1, f_2^-), \tag{4}$$

where $\mathcal{D}$ denotes a distance function, f_2^- means the gradient back-propagation of f_2 is frozen.

Algorithm 1: Adversarial Fine-tuning via Disentanglement

Require: Training set $\mathcal{D}$, Hyperparameters α, β, Number of epochs T, Learning rate η, Naturally pre-trained model parameters θ^P, Model parameters except linear block D_2 parameters θ_1, Linear block D_2 parameters θ_2.
Ensure: Fine-tuned model parameters θ_1.

```
Initialize θ1 by θ^P;
for t = 1, ··· , T do
  for x, y ∈ D do
    Calculate x' by by Formula 2;
    Calculate features f1, f2 by Formula 1;
    Calculate loss L1 Formula 3;
    θ2 = θ2 − η · d(α·L1)/dθ2;
    Calculate loss L2 Formula 4;
    θ1 = θ1 − η · d(β·L2)/dθ1;
    Calculate loss L3 by Formula 5;
    θ1 = θ1 − η · dL3/dθ1;
  end for
end for
```

Adversarial Fine-tuning Based on Alignment

The second strategy is to align the adversarial features in the fine-tuned model with the natural features in the pre-trained model. Since the pre-trained model has a high natural accuracy and doesn't suffer from adversarial attacks, the natural features in the pre-trained model can be identified as ideal intrinsic features. Thus, adversarial features in the fine-tuned model can be greatly improved by being aligned with the natural features in the pre-trained model. Besides, lots of work (Zhang et al. 2019; Wang et al. 2020; Suzuki et al. 2023) conducts an alignment between natural and adversarial features, which is regarded to mitigate the trade-off between accuracy and robustness. We take the disentangled features f_1 to match the natural features in the pre-trained model, further removing the confused features from f_1. The AFT loss function with the alignment can be formulated as:

$$\mathcal{L}_3 = \mathcal{L}_{CE}(\omega^F \cdot f_1, y) + \gamma \cdot \mathcal{D}(f_1, g(x;\theta_g^P)^-), \tag{5}$$

where θ_g^P denotes parameters of the feature extractor in the pre-trained model, γ denotes the weight of the alignment, and $g(x;\theta_g^P)^-$ denotes the gradient back-propagation of $g(x;\theta_g^P)$ is frozen. Therefore, the objective function of our AFD is defined as follows:

$$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_1 + \beta \cdot \mathcal{L}_2 + \mathcal{L}_3, \tag{6}$$

where α and β denote the weights of $\mathcal{L}_1$ and $\mathcal{L}_2$. The algorithm of AFD in Algorithm 1.

Experiments

This section presents experiments with AFD. We first introduce our experiment setting. We further make comparisons on multiple architectures and datasets to show the effectiveness. Then we conduct an ablation study to show the functions of the disentangling module and alignment. Finally, we conduct an empirical analysis to support our hypothesis.

Setting

Datasets We conduct experiments on three benchmark datasets including CIFAR-10 and CIFAR-100 (Krizhevsky 2009), Tiny-ImageNet (Deng et al. 2009). CIFAR-10 dataset contains 60,000 color images having a size of 32×32 in 10 classes, with 50,000 training and 10,000 test images. CIFAR-100 dataset contains 50,000 training and 10,000 test images in 100 classes. Tiny-ImageNet dataset contains 100000 images of 200 classes (500 for each class) downsized to 64×64 colored images. Each class has 500 training images, 50 validation images and 50 test images. We only use training and validation images of Tiny-ImageNet.

Baselines To make a comprehensive comparison, we have employed various techniques to demonstrate the effectiveness of our method AFD. Our primary baselines are AFT methods including vanilla AFT (vAFT) (Jeddi, Shafiee, and Wong 2020), AFT+RGKD (AFKD) (Suzuki et al. 2023), and ARREST (Suzuki et al. 2023). All of them are built upon naturally pre-trained models. Additionally, we compare our AFD with a state-of-the-art AFT method RiFT (Zhu et al. 2023), which relies on adversarially pre-trained and has powerful out-of-distribution performances. Besides, we select several advanced fast adversarial training methods, including FreeAT (Shafahi et al. 2019), FGSM-GA (Andriushchenko and Flammarion 2020), ATAS (Huang et al. 2023), since they are also low-cost techniques. Moreover, we compare with baseline methods of adversarial training including PGD-AT (Madry et al. 2017), TRADES (Zhang et al. 2019) to show our great advance in robustness.

Optimization Details The optimizer is Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9. We fine-tune the pre-trained models with a batch size of 128, a learning rate of 0.0025, a weight decay of 5.0×10^{-4}, and training epochs of 20. The schedule of learning rate is the same as Suzuki et al. (2023). We utilize Exponential Moving Average (EMA) (Hunter 1986) to gain better parameters.

Implementation Details We use ResNet18 (He et al. 2016) and WideResNet-34-10 (Zagoruyko and Komodakis 2016) as the main DNN architectures, following previous studies (Goldblum et al. 2020; Suzuki et al. 2023; Wang et al. 2020). The distance function chosen for $\mathcal{L}_2$ and $\mathcal{L}_3$ is the angular distance: $\mathcal{D}(u, v) = 1 - \frac{u \cdot v}{\|u\|_2 \cdot \|v\|_2}$. For hyperparameters $\{\alpha, \beta, \gamma\}$, we suggest $\{0.05, 0.25, 25\}$ in most cases. We report the natural accuracy (Clean), the robust accuracy against PGD-20 (PGD) (Madry et al. 2017), the robust accuracy against AutoAttack (AA) (Croce and Hein 2020), and the average value (Avg) for evaluation. Since D_1 is seamlessly integrated with ω^F at inference time, we don't design adaptive attacks about f_1. The maximum perturbation is set to $8/255$. L_∞-norm PGD with a random start, a step size of 0.003, and attack iterations of 20 is utilized in the evaluation. In the following tables, † denotes the results excerpted in the paper (Huang et al. 2023), which are evaluated by PGD-10 in the manuscript. **Bold** and underline indicate the highest and second-highest values for each metric. We only report **the results of the last epoch**, since AFT methods always gain the best performance in the last epoch.

Main Results

ResNet18 First, AFD effectively mitigates the trade-off between generalization and robustness. As shown in Table 1, AFD takes the lead in robust performance and exhibits the second-best clean accuracy **among AFT methods**. It gains less clean accuracies compared to ARREST by 1.70%, but its robust accuracies against PGD-20 and AA outperform ARREST by 2.50% and 2.68%. The improvement in robustness is evidently larger than the drop in clean accuracy. This result indicates the positive effect of feature disentanglement on the accuracy-robustness trade-off.

Moreover, AFD exceeds almost all the AT and FAT methods. Although the powerful AT baseline TRADES outperforms AFD in the robust accuracies on CIFAR-10 and CIFAR-100 by 0.56%, it surpasses TRADES in clean and average accuracy by 6.365% and 1.84%. Besides, it surpasses TRADES on Tiny-ImageNet in all metrics. The result implies that efficient AFD can achieve comparable performances with AT.

WideResNet-34-10 Second, AFD takes further advantages on large architectures. As shown in Table 2, AFD maintains the best robust and average accuracies among all the approaches. It exceeds the second-best baselines by approximately 1.62% and 1.34%. The enhancement of robustness further enlarges compared to that on ResNet18. In addition, the natural generalization of AFD always ranks second and is less than the best clean accuracy by 0.88% on average. The discrepancy of clean accuracies between AFD and the best one is smaller than that on ResNet18 (1.69%). Furthermore, AFD outperforms AT baselines in all metrics. These demonstrate that AFD achieves a better trade-off between generalization and robustness on larger architectures.

It's worth noting that AFD doesn't exhibit optimal natural generalization among AFT methods. It is attritued to merely adversarial features purified by disentangling modules. Intuitively, natural features can also be boosted by disentanglement mechanism. Nevertheless, it will result in additional computing cost. Besides, current AFD has remained the second-best clean accuracy among various methods.

Summary These experiments indicate that AFD always outperforms all the baselines for different attacks, datasets, and architectures in the aspect of overall performance. It proves the effectiveness of our AFD.

Ablation Study

Disentangler and Alignment As shown in Table 3, vanilla AFT with the disentangler (vAFT+D) exhibits superior performances compared to vanilla AFT by 0.70%, 1.28%, and 1.40%, respectively. It shows that the disentangler effectively eliminates the confused features that lead to wrong predictions. Besides, vanilla AFT with the alignment of features (vAFT+A) outperforms vanilla AFT by 0.68%, 0.31%, and 1.33%. It indicates that the alignment successfully contributes to removing the confused features. Moreover, the combination of disentangler and alignment (AFD) has the best performance on both natural and robust accuracy, which shows two strategies are compatible and the combination can boost the positive effects of individual strategies.

Methods	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	Clean	PGD	AA	Avg	Clean	PGD	AA	Avg	Clean	PGD	AA	Avg
PGD-AT	84.43	48.82	43.68	58.98	59.09	24.05	20.67	34.60	52.12	19.38	16.79	29.43
TRADES	82.37	**53.27**	**48.52**	61.39	55.47	28.12	**23.49**	35.69	49.28	22.59	17.06	29.64
Free-AT†	78.37	40.90	36.00	51.76	50.56	19.57	15.09	28.41	-	-	-	-
FGSM-GA†	80.10	49.14	43.44	57.56	50.61	24.48	19.42	31.50	-	-	-	-
ATAS†	81.22	50.03	45.38	58.88	55.49	27.68	22.62	35.26	-	-	-	-
vAFT	83.93	51.40	45.48	60.27	60.83	26.55	21.96	36.45	52.75	21.29	16.76	30.27
RiFT	84.49	49.24	43.66	59.13	59.51	24.01	20.48	34.67	52.24	20.89	17.05	30.06
AFKD	84.71	51.32	46.09	60.71	65.54	28.09	22.92	38.85	57.17	22.29	16.68	32.05
ARREST	**85.71**	51.23	46.23	61.05	**67.00**	26.89	21.34	38.41	**60.36**	19.28	12.58	30.74
AFD	84.88	52.70	47.40	**61.66**	65.69	**28.56**	23.05	**39.10**	57.41	**23.63**	**17.73**	**32.92**

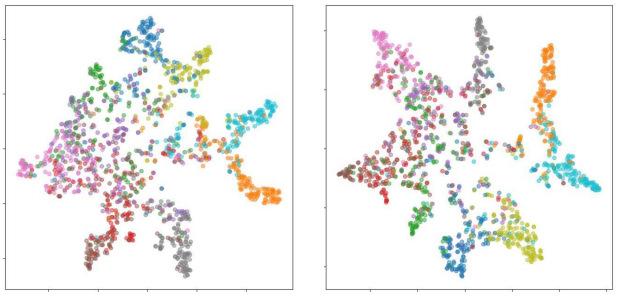
Table 1: Quantitative evaluations of all methods on ResNet18.

Methods	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	Clean	PGD	AA	Avg	Clean	PGD	AA	Avg	Clean	PGD	AA	Avg
PGD-AT	86.35	49.57	45.81	60.58	60.30	25.14	22.65	36.03	54.24	19.03	16.51	29.93
TRADES	85.68	52.53	49.22	62.48	57.67	28.26	25.25	37.06	49.22	23.33	18.51	30.35
vAFT	87.42	53.57	48.49	63.16	64.92	27.19	23.59	38.57	57.17	22.24	18.02	32.48
RiFT	84.49	49.24	45.83	59.85	60.59	25.57	22.67	36.28	55.09	19.51	16.79	30.46
AFKD	**88.54**	54.86	49.57	64.31	68.50	30.12	25.28	41.30	60.92	24.27	19.01	34.73
ARREST	88.42	54.16	49.45	64.01	**69.52**	29.47	24.55	41.18	**65.07**	22.73	16.67	34.82
AFD	88.08	**56.12**	**51.62**	**65.27**	68.51	**32.63**	**27.62**	**42.92**	63.89	**25.73**	**19.13**	**36.25**

Table 2: Quantitative evaluations of all methods on WideResNet-34-10.

Methods	vAFT	vAFT+D	vAFT+A	AFD
Clean	83.93	84.63	84.61	84.88
PGD	51.40	52.68	51.71	52.70
AA	45.48	46.88	46.81	47.40

Table 3: The performances of vAFT and other methods on CIFAR-10 on ResNet18. '+D' denotes the method with the disentangler, and '+A' denotes the method with the alignment of features.



(a) Features before disentanglement (b) Features after disentanglement

Figure 3: t-SNE visualization on CIFAR-10 on ResNet18.

Hyperparameter There are three hyperparameters in our AFD: α, β and γ. As illustrated in Figure 4, there is a remarkable trade-off between accuracy and robustness when β and γ are growing. On the one hand, a larger value of β indicates a larger exclusive force between the intrinsic and confused features, leading to less non-robust features utilized in classification and better robustness. On the other hand, as γ increases, the natural accuracy increases but robust accuracy drops. This occurs because more feature information of natural samples is transferred to the fine-tuned model through the feature alignment. Besides, the variations of both natural and robust accuracies are less than 0.60%, demonstrating that our **AFD is insensitive to variations of hyperparameters**. This conclusion is also true among different datasets.

Empirical Analysis of Disentangled Features

we analyze the feature distance between the intrinsic features f_1 and natural features f_{nat}, as shown in Figure 1 (a). In contrast to the growing trend of feature distance in existing AFT methods, the optimization trajectory of the feature distance in AFD exhibits a fast convergence trend. It implies that AFD has mitigated the gap between natural and adversarial features. Additionally, the magnitudes of the feature distance in AFD are smaller than other AFT methods, e.g., the L_∞ distance of AFD (0.212) is notably less than those of ARREST and vanilla AFT (0.453, 0.236). These results demonstrate our AFD can enhance the adversarial robustness by alleviating the feature gap.

Moreover, we evaluate the performances of two types of disentangled features f_1, f_2 on various datasets and models. As shown in Table 4, the natural accuracies of f_2 are lower than those of f_1 by $5.67\% \sim 51.04\%$. It indicates the disentangler has successfully extracted and removed the con-

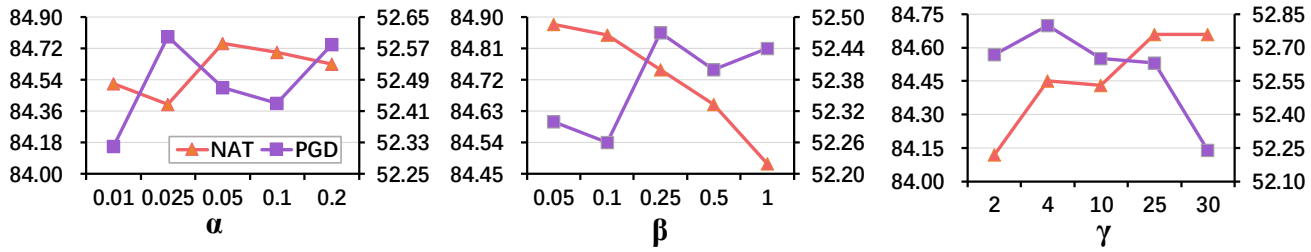


Figure 4: The clean accuracy (NAT) and robust accuracy against PGD-20 (PGD) of AFD with different hyperparameters.

Model	Data	f_1	f_2	Δ
RN18	CIFAR-10	84.88	39.05	45.83
RN18	CIFAR-100	65.69	58.17	7.52
RN18	Tiny-ImageNet	57.41	48.78	8.63
RN50	CIFAR-10	87.01	35.97	51.04
RN50	CIFAR-100	67.45	56.17	11.28
WRN-34-10	CIFAR-10	88.08	46.89	41.19
WRN-34-10	CIFAR-100	68.51	62.84	5.67
WRN-34-10	Tiny-ImageNet	63.89	55.15	8.74

Table 4: The natural accuracies of the intrinsic features f_1 and confused features f_2. RN and WRN are abbreviations of ResNet and WideResNet, respectively. Δ denotes the accuracy disparity.

fused features as f_2. High accuracies of f_1 indicate models benefit from eliminating the confused features. Interestingly, the disparity of accuracies on CIFAR-100 is smaller than those on CIFAR-10 and Tiny-ImageNet. We suppose that it is challenging to learn clear feature space from a small-resolution dataset with numerous classes, which made the linear disentangler difficult to extract confused features.

We make further feature visualization by t-SNE (Van der Maaten and Hinton 2008). We take the fine-tuned model by AFD. The we acquire features of adversarial samples crafted by PGD-20. We visualizes two types of features: the features before disentanglement and the intrinsic features f_1. As illustrated in Figure 3, features after disentanglement has a distincter class boundary than features before disentanglement, indicating the effectiveness of disentanglement.

Training Time

We also measure the training time as an evaluation indicator. Notice that the training time of PGD-AT with † is almost the same as ours. Thus it is fair to compare our training time with theirs. The vanilla ResNet18 and WideResNet-34-10 have parameters of 11.2M and 46.2M. As shown in Table 5, our AFD costs several times less time than AT methods, and its total training time surpasses other AFT approaches merely by less than 200 seconds. Considering its advanced performance, the time cost is worthy.

The vanilla ResNet18 and WideResNet-34-10 have parameters of 11.2M and 46.2M. In the training process, the additional module has a weight matrix that doesn't depend on the input size. It is $2*512*512 = 0.5$M (4.46% increase) on ResNet18 and $2*1024*1024 = 2$M (4.33% increase) on WideResNet-34-10, which is acceptable for better training.

Method	Epoch	Time_{one}	$\text{Time}_{\text{total}}$
Standard	100	6.0	590
PGD-AT	120	37	4440
PGD-AT†	-	-	4428
TARDES	120	53	6360
Free-AT†	10	119	1188
FGSM-GA†	30	68	2052
ATAS†	30	36	1080
vanilla AFT	20	37	1330
RiFT	10	25	4690
AFKD	20	40	1390
ARREST	20	36	1310
AFD	20	46	1510

Table 5: Training time (second) of various methods on ResNet18 on CIFAR10. We show the time for one epoch (Time_{one}) and the expected values of the total time ($\text{Time}_{\text{total}}$). 'Standard' denotes the natural pre-training.

Limitation and Futural Work

Subject to computation resources, we haven't conducted experiments on ImageNet-1k. We will explore effective adversarial fine-tuning approaches on large-resolution datasets in the future. Besides, we observe that the increasing gap of features also appears in adversarial training. It remains a lot for further studies.

Conclusion

This paper uncovers a surprising increasing trend in the gap of features between natural and adversarial samples in AFT methods, and further investigates it from the perspective of features. We model features as the intrinsic features and confused features, in which the latter is defined as the specific latent features leading to the feature gap. Then we propose Adversarial Fine-tuning via Disentanglement (AFD) to bridge the feature gap to enhance robustness. We design the feature disentangler to explicitly separate out the confused features from features of adversarial samples. Besides, the disentangled features are aligned with the natural features in the pre-trained model. Experiments demonstrate that AFD effectively mitigates the feature gap and achieves a good trade-off between generalization and robustness.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants U22A2096, 62441601, and 62306227, in part by the Shaanxi Province Core Technology Research and Development Project under grant 2024QY2-GJHX-11, in part by the Fundamental Research Funds for the Central Universities under Grants QTZX23042 and ZYTS24142.

References

Agarwal, A.; Ratha, N.; Singh, R.; and Vatsa, M. 2023. Robustness Against Gradient Based Attacks Through Cost Effective Network Fine-Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 28–37.

Allen-Zhu, Z.; and Li, Y. 2022. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 977–988. IEEE.

Andriushchenko, M.; and Flammarion, N. 2020. Understanding and Improving Fast Adversarial Training. *ArXiv*, abs/2007.02617.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*.

Croce, F.; and Hein, M. 2022. Adversarial Robustness against Multiple and Single l_p-Threat Models via Quick Fine-Tuning of Robust Classifiers. In *International Conference on Machine Learning*, 4436–4454. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dong, Y.; Xu, K.; Yang, X.; Pang, T.; Deng, Z.; Su, H.; and Zhu, J. 2021. Exploring memorization in adversarial training. *arXiv preprint arXiv:2106.01606*.

Goldblum, M.; Fowl, L.; Feizi, S.; and Goldstein, T. 2020. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3996–4003.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. *CoRR*, abs/1412.6572.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hickling, T.; Aouf, N.; and Spencer, P. 2023. Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning. *IEEE Transactions on Intelligent Vehicles*.

Huang, L.; Zhang, C.; and Zhang, H. 2020. Self-adaptive training: beyond empirical risk minimization. *Advances in neural information processing systems*, 33: 19365–19376.

Huang, Z.; Fan, Y.; Liu, C.; Zhang, W.; Zhang, Y.; Salzmann, M.; Süsstrunk, S.; and Wang, J. 2023. Fast adversarial training with adaptive step size. *IEEE Transactions on Image Processing*.

Hunter, J. S. 1986. The exponentially weighted moving average. *Journal of Quality Technology*, 18: 203–210.

Jeddi, A.; Shafiee, M. J.; and Wong, A. 2020. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *arXiv preprint arXiv:2012.13628*.

Jia, X.; Zhang, Y.; Wu, B.; Ma, K.; Wang, J.; and Cao, X. 2022. LAS-AT: adversarial training with learnable attack strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13398–13408.

Jin, G.; Yi, X.; Huang, W.; Schewe, S.; and Huang, X. 2022. Enhancing adversarial training with second-order statistics of weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15273–15283.

Kim, H.; Lee, W.; and Lee, J. 2020. Understanding Catastrophic Overfitting in Single-step Adversarial Training. In *AAAI Conference on Artificial Intelligence*.

Kim, W. J.; Cho, Y.; Jung, J.; and Yoon, S.-E. 2023. Feature separation and recalibration for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8183–8192.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Kumari, N.; Singh, M.; Sinha, A.; Machiraju, H.; Krishnamurthy, B.; and Balasubramanian, V. N. 2019. Harnessing the vulnerability of latent layers in adversarially trained models. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2779–2785.

Lee, M.; and Kim, D. 2023. Robust evaluation of diffusion-based adversarial purification. *arXiv preprint arXiv:2303.09051*.

Liu, D.; Wang, X.; Peng, C.; Wang, N.; Hu, R.; and Gao, X. 2024. Adv-diffusion: imperceptible adversarial face identity attack via latent diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3585–3593.

Liu, Z.; Xu, Y.; Ji, X.; and Chan, A. B. 2023. TWINS: A Fine-Tuning Framework for Improved Transferability of Adversarial Robustness and Generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Madaan, D.; Shin, J.; and Hwang, S. J. 2021. Learning to generate noise for multi-attack robustness. In *International Conference on Machine Learning*, 7279–7289. PMLR.

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

Maini, P.; Wong, E.; and Kolter, Z. 2020. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, 6640–6650. PMLR.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Uesato, J.; and Frossard, P. 2018. Robustness via Curvature Regularization,

and Vice Versa. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9070–9078.

Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*.

Pang, T.; Du, C.; Dong, Y.; and Zhu, J. 2018. Towards robust detection of adversarial examples. *Advances in neural information processing systems*, 31.

Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J. P.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial Training for Free! In *Neural Information Processing Systems*.

Si, C.; Zhang, Z.; Qi, F.; Liu, Z.; Wang, Y.; Liu, Q.; and Sun, M. 2020. Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning. *arXiv preprint arXiv:2012.15699*.

Sun, J.; Wang, J.; Nie, W.; Yu, Z.; Mao, Z.; and Xiao, C. 2023. A critical revisit of adversarial robustness in 3D point cloud recognition with diffusion-driven purification. In *International Conference on Machine Learning*, 33100–33114. PMLR.

Suzuki, S.; Yamaguchi, S.; Takeda, S.; Kanai, S.; Makishima, N.; Ando, A.; and Masumura, R. 2023. Adversarial Finetuning with Latent Representation Constraint to Mitigate Accuracy-Robustness Tradeoff. *arXiv preprint arXiv:2308.16454*.

Tramer, F.; and Boneh, D. 2019. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*.

Wei, H.; Tang, H.; Jia, X.; Wang, Z.; Yu, H.; Li, Z.; Satoh, S.; Van Gool, L.; and Wang, Z. 2022. Physical adversarial attack meets computer vision: A decade survey. *arXiv preprint arXiv:2209.15179*.

Wei, H.; Wang, Z.; Jia, X.; Zheng, Y.; Tang, H.; Satoh, S.; and Wang, Z. 2023. Hotcold block: Fooling thermal infrared detectors with a novel wearable design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15233–15241.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. *ArXiv*, abs/2001.03994.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33: 2958–2969.

Xia, R.; Zhou, D.; Liu, D.; Li, J.; Yuan, L.; Wang, N.; and Gao, X. 2024. Inspector for Face Forgery Detection: Defending against Adversarial Attacks from Coarse to Fine. *IEEE Transactions on Image Processing*.

Xie, C.; Wu, Y.; Maaten, L. v. d.; Yuille, A. L.; and He, K. 2019. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 501–509.

Yan, H.; Zhang, J.; Niu, G.; Feng, J.; Tan, V.; and Sugiyama, M. 2021. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *International Conference on Machine Learning*, 11693–11703. PMLR.

Yang, S.; Guo, T.; Wang, Y.; and Xu, C. 2021. Adversarial robustness through disentangled representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3145–3153.

Yoon, J.; Hwang, S. J.; and Lee, J. 2021. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, 12062–12072. PMLR.

Yu, C.; Han, B.; Shen, L.; Yu, J.; Gong, C.; Gong, M.; and Liu, T. 2022. Understanding robust overfitting of adversarial training and beyond. In *International Conference on Machine Learning*, 25595–25610. PMLR.

Yu, Y.; Gao, X.; and Xu, C.-Z. 2021. Lafeat: Piercing through adversarial defenses with latent features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5735–5745.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. *ArXiv*, abs/1605.07146.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.

Zheng, Z.; and Hong, P. 2018. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. *Advances in Neural Information Processing Systems*, 31.

Zhou, D.; Wang, N.; Han, B.; and Liu, T. 2022. Modeling Adversarial Noise for Adversarial Training. In *International Conference on Machine Learning*, 27353–27366. PMLR.

Zhou, N.; Wang, N.; Liu, D.; Zhou, D.; and Gao, X. 2023. Enhancing Robust Representation in Adversarial Training: Alignment and Exclusion Criteria. *arXiv e-prints*, arXiv–2310.

Zhu, K.; Hu, X.; Wang, J.; Xie, X.; and Yang, G. 2023. Improving Generalization of Adversarial Training via Robust Critical Fine-Tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4424–4434.