Advancing Example Exploitation Can Alleviate Critical Challenges in Adversarial Training

Yao Ge[1] Yun Li[1 *] Keji Han[1] Junyi Zhu[2] Xianzhong Long[1]
[1] School of Computer Science, Nanjing University of Posts and Telecommunications[†]
[2] College of Arts & Sciences, Boston University

Abstract

Deep neural networks have achieved remarkable results across various tasks. However, they are susceptible to adversarial examples, which are generated by adding adversarial perturbations to original data. Adversarial training (AT) is the most effective defense mechanism against adversarial examples and has received significant attention. Recent studies highlight the importance of example exploitation, where the model's learning intensity is altered for specific examples to extend classic AT approaches. However, the analysis methodologies employed by these studies are varied and contradictory, which may lead to confusion in future research. To address this issue, we provide a comprehensive summary of representative strategies focusing on exploiting examples within a unified framework. Furthermore, we investigate the role of examples in AT and find that examples which contribute primarily to accuracy or robustness are distinct. Based on this finding, we propose a novel example-exploitation idea that can further improve the performance of advanced AT methods. This new idea suggests that critical challenges in AT, such as the accuracy-robustness trade-off, robust overfitting, and catastrophic overfitting, can be alleviated simultaneously from an example-exploitation perspective. The code can be found in https://github.com/geyao1995/advancing-example-exploitation-in-adversarial-training.

1. Introduction

Adversarial examples, generated by adding adversarial perturbations to original data, can easily deceive today's deep learning models [1]. Among numerous methods to mitigate this vulnerability, adversarial training (AT) is the most effective which takes adversarial examples into the training process [2, 3, 4, 5]. However, AT is not without limitations and confronts critical challenges, including: (1) the trade-off between accuracy (classification success rate for original samples) and robustness (classification success rate for examples after adding adversarial perturbations), where improving one metric comes at the expense of the other [3]; (2) the phenomenon of robust overfitting (RO), which is characterized by a gradual decline in robustness during the later stage of training [6]; and (3) the occurrence of catastrophic overfitting (CO), which leads to a sudden drop in robustness after a particular epoch of training [7]. To alleviate these challenges, numerous research efforts have explored diverse perspectives, such as modifying model components [8, 9], refining weight optimization policies [10, 11], and generating supplementary training data [12, 13]. In addition, example exploitation has emerged as a promising perspective that has gained sustained attention. It emphasizes the unequal contribution of examples to the model during AT and is less computationally demanding compared to other perspectives [14, 15, 16, 17, 18]. Moreover, it holds great potential to uncover the essence of adversarial examples.

To enhance AT, example-exploitation methods typically aim to promote or discourage the learning of specific example features by the model. To prevent the model from overfitting erroneous features, SAT dynamically adjusts the one-hot label of each example in each training epoch [14]. MART prioritizes the impact of misclassified examples on model robustness by incorporating a misclassification-aware term into its objective function [15]. FAT assigns each example a different attack iteration to search for friendly adversarial examples that can improve model accuracy [16]. GAIRAT reweights the loss function for each example based on its geometry value, which approximates the distance from the example to the class boundary [17]. The recent work TEAT integrates the temporal ensembling approach to prevent excessive memorization of noisy adversarial examples [18]. Although these works offer various strategies for exploiting examples, their underlying insights are different and sometimes conflicting. For instance, MART focuses on examples that FAT aims to avoid. To

*Corresponding author (liyun@njupt.edu.cn).
[†]This work was partially supported by National Natural Science Foundation of China (No.61772284) and Graduate Research and Innovation Projects of Jiangsu Province (No.KYCX21_0795).

eliminate confusion caused by these discrepancies in future studies, a systematic summary of example-exploitation methods is necessary to advance this field.

In this paper, we propose an unified framework to summarize the exploiting strategies used in representative works by dividing examples into two crucial parts: accuracy-crucial (A-C) and robustness-crucial (R-C). Our investigation shows that A-C and R-C examples significantly contribute to accuracy and robustness, respectively, and the insights of existing example-exploitation research can be interpreted as treating A-C and R-C examples differently. We also demonstrate that there is further potential for advancement in the topic of example-exploitation AT by investigating the roles of A-C and R-C examples. To improve the efficacy of A-C and R-C examples, we propose a novel example treatment that emphasizes the importance of both A-C examples for accuracy and R-C examples for robustness. By applying this treatment in AT, we achieve simultaneous alleviation of the previously mentioned critical challenges from the perspective of example exploitation, which has not been achieved by any prior work. Specifically, our contributions are summarized as follows:

- We perform a systematic analysis of example-exploitation methods in adversarial training and identify the examples that have a greater impact on improving either accuracy or robustness.
- We propose a novel treatment idea for exploiting examples in adversarial training, which fully leverages the potential of accuracy-crucial examples to improve accuracy and robustness-crucial examples to enhance robustness.
- Through simply applying our treatment to adversarial training, we demonstrate, for the first time, the possibility of simultaneously alleviating three critical challenges: the accuracy-robustness trade-off, robust overfitting, and catastrophic overfitting, solely from the example-exploitation perspective.

2. Related work

To confer adversarial robustness on the model, Adversarial Training (AT) calculates the perturbation for each original example to generate the adversarial counterpart and considers them as training examples. Over samples $(\boldsymbol{x}, y) \in D : (\mathcal{X}, \mathcal{Y})$, let $f(\boldsymbol{x}; \boldsymbol{\theta})$ denotes the classification function of the model f with parameters $\boldsymbol{\theta}$, which maps input example $\boldsymbol{x}$ to the output logits for classes in $\mathcal{Y}$. The adversarial perturbation $\boldsymbol{\delta}$ satisfies $\|\boldsymbol{\delta}\|_p \leq \varepsilon$ for a small $\varepsilon > 0$ to keep it imperceptible (we focus on the $p = \infty$ in this paper). According to [2], training a robust classification model can be formalized as the following min-max optimization problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim D} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \varepsilon} L(f(\boldsymbol{x}+\boldsymbol{\delta}; \boldsymbol{\theta}), y) \right], \quad (1)$$

where L can be set as Cross-Entropy (CE) loss, which is commonly used in classification tasks. The inner maximization is approximated by generating adversarial perturbation $\boldsymbol{\delta}$ through the attack in the training process. One popular method to generate $\boldsymbol{\delta}$ is Projected Gradient Descent (PGD) [2], which performs a fixed number of gradient ascent iterations using a small step-size a:

$$\boldsymbol{\delta}_{k+1} = \mathrm{Clip}(a \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}+\boldsymbol{\delta}_k} L(f(\boldsymbol{x}+\boldsymbol{\delta}_k; \boldsymbol{\theta}), y)), \quad (2)$$

where Clip keeps $\boldsymbol{x} + \boldsymbol{\delta}$ stay in the ε-ball centered at $\boldsymbol{x}$. AT exhibits the trade-off phenomenon between accuracy and robustness. For balancing the them, TRADES [3] uses Kullback-Leibler divergence (KL) loss to implement L in Equation (2). The overall loss of TRADES is

$$\mathrm{CE}\left(f(\boldsymbol{x}, \boldsymbol{\theta}), y\right) + \lambda \cdot \mathrm{KL}\left(f(\boldsymbol{x}, \boldsymbol{\theta}) \| f(\boldsymbol{x}', \boldsymbol{\theta})\right), \quad (3)$$

where $\boldsymbol{x}'$ is the adversarial example $\boldsymbol{x} + \boldsymbol{\delta}$ and λ is the regularization weight. Generally, a larger λ may increase robustness but decrease accuracy [3].

In contrast to the multi-step AT method presented above, single-step adversarial training remains an area of interest as it can reduce the computational costs associated with multi-step iterations during the training phase. One iteration of gradient ascent with respect to the original examples $\boldsymbol{x}$ is performed to generate $\boldsymbol{\delta}$ [19]:

$$\boldsymbol{\delta} = \varepsilon \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} L(f(\boldsymbol{x}; \boldsymbol{\theta}), y)). \quad (4)$$

While many methods, such as SAT [14], MART [15], FAT [16], GAIRAT [17], and TEAT [18], have demonstrated the effectiveness of example-exploitation in multi-step AT, there has been no work done to extend this perspective to single-step AT. In this paper, we show that single-step AT can also benefit from example-exploitation, as we improve two single-step methods: FastAT [7] and GradAlign [20]. The appendix provides additional details on both representative example-exploitation AT methods and advanced single-step AT methods.

3. Roles of different examples

In this section, we introduce a new metric called "robustness confidence" that can identify examples crucial to accuracy or robustness. We then use this metric to analyze representative adversarial training (AT) methods. The demonstrative experiments in this section are based on CIFAR10 [21] dataset and PreAct ResNet-18 [22] model.

3.1. Robustness confidence of each example

To identify the role of example $\boldsymbol{x}_i$ in training, we define the *robustness confidence* c_i. At t_{th} training epoch, c_i can be denoted as c_i^t:

$$c_i^t = \alpha \cdot c_i^{t-1} + (1-\alpha) \cdot \boldsymbol{p}_y(\boldsymbol{x}_i'), \quad (5)$$

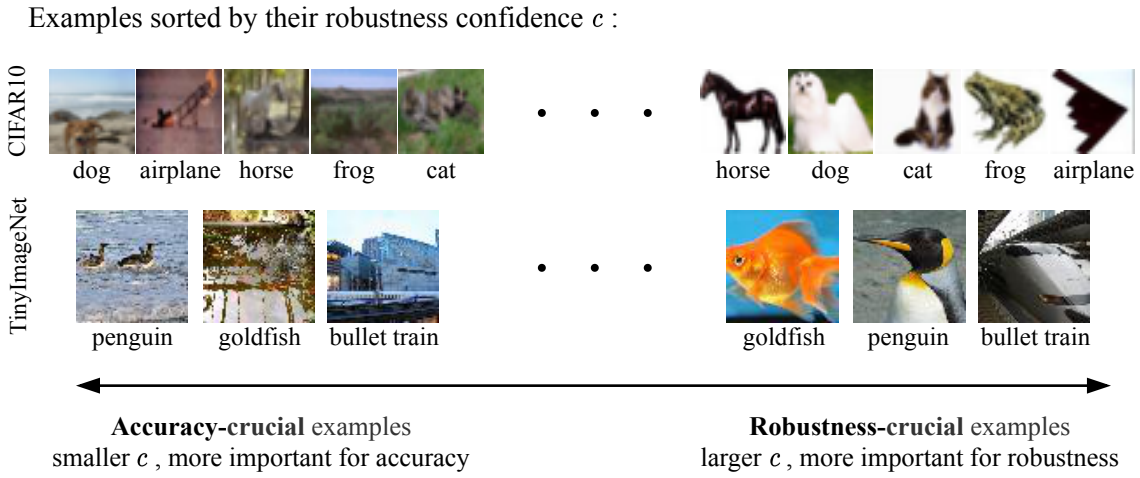
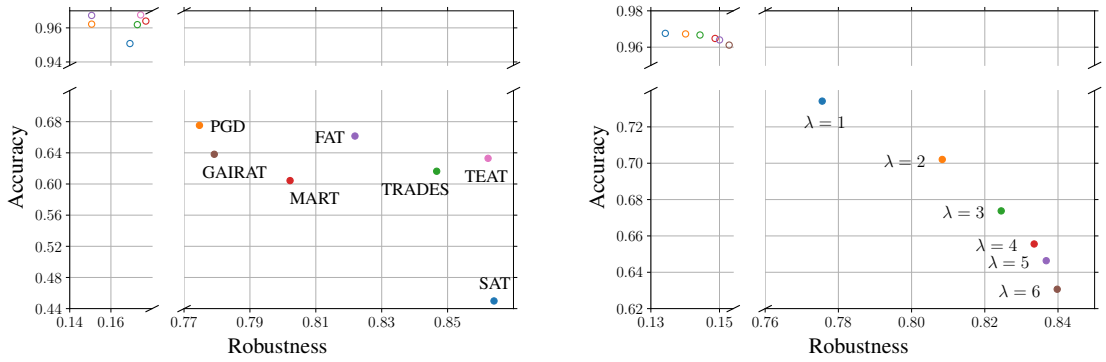


Figure 1: Identification for A-C and R-C examples in two datasets. More examples are shown in the appendix.



(a) Test results of different AT methods

(b) Test results of TRADES with different λ

Figure 2: In two panels, the solid points correspond to the accuracy on the A-C examples subset and the robustness on the R-C examples subset. And the hollow points correspond to the accuracy on the R-C examples subset and the robustness on the A-C examples subset. The axes are broken for zooming in on details.

where $c_i^t \in [0, 1]$, α is the momentum factor and $\boldsymbol{p}_y(\boldsymbol{x}_i')$ is the predicted classification probability for true class y. At first epoch, c_i^1 is initialized as $\boldsymbol{p}_y(\boldsymbol{x}_i')$. The accumulation of $\boldsymbol{p}_y(\boldsymbol{x}_i')$ in Equation (5) is achieved through temporal ensembling operation technique, which is also adopted in other works [23, 14, 18]. However, robustness confidence c_i serves a distinct purpose: it quantifies the model's ability to correctly classify adversarial examples $\boldsymbol{x}_i'$ generated from $\boldsymbol{x}_i$ throughout the training process. A value of c_i closer to 1 implies that the model can more easily fit the adversarial features around $\boldsymbol{x}_i$, whereas a value closer to 0 indicates that the model struggles to learn valid features from $\boldsymbol{x}i'$, as $\boldsymbol{p}_y(\boldsymbol{x}_i')$ remains small in every epoch.

3.2. Accuracy/Robustness-crucial examples

Based on proposed robustness confidence c, we define two types of examples: accuracy-crucial (A-C) and robustness-crucial (R-C). An A-C example is characterized by having a small robustness confidence, while an R-C example has a large robustness confidence. To illustrate the concept of A-C and R-C examples, we present a diagram in Figure 1. Additionally, we evaluate the performance of different adversarial training (AT) methods on subsets of A-C and R-C examples (i.e., 30% of the test examples with the smallest and largest c values, respectively) in Figure 2a. Since TRADES method can disentangle the accuracy and robustness of the model, we also vary λ in Equation (3) to investigate the changes of model performance on A-C and R-C examples, as shown in Figure 2b.

In Figure 2, we can observe the following: 1) The hollow points predominantly appear in the upper left corner, indicating that AT methods typically yield low robustness on A-C examples and high accuracy on R-C examples. 2) Compared to the hollow points, the solid points display noticeable variability in the lower right region, suggesting that the performance differences among these AT methods mainly

Table 1: Different treatments of some adversatial training methods and corresponding test results. The symbol +/− means the method tends to enhance/reduce the accuracy (Acc) or robustness (Rob) learning on A-C or R-C examples. The symbol ↑/↓ means the performance increases/decreases on A-C or R-C examples in test set (compared with TRADES). The symbol ✓/✗ means the method relieves/worsens the robust overfitting (RO). The treatments of each AT method are analyzed in detail in the appendix.

	Treatments on training set				Results on test set						
	A-C examples		R-C examples		A-C examples		R-C examples		All examples		RO
	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	
SAT [14]	−				↓	↓		↑	↓	↑	✓
MART [15]		+			↓	↑		↓	↓	↑	✗
FAT [16]		−		−	↑	↑		↓	↑	↓	✓
GAIRAT [17]	+	+			↑	↑	↓	↓	↓	↑	✗
TEAT [18]		−			↑			↑	↑	↑	✓
Ours		−		+	↑			↑	↑	↑	✓

stem from their accuracy on A-C examples and robustness on R-C examples. 3) The results of the TRADES method provide a more intuitive rule: improving robustness on R-C examples can lead to a decrease in accuracy on A-C examples, and vice versa. These findings offer a novel insight into AT: **To strengthen the effectiveness of adversarial training, it is important to emphasize accuracy on A-C examples and robustness on R-C examples**.

The above conclusion motivate us to investigate how the insights derived from example-exploitation AT methods interact with the A-C and R-C examples. To ensure a viable comparison, we explore the correlation of robustness confidence (c) with other metrics studied in AT [17, 24, 25, 26] and find positive/negative correlations with them (see appendix for details). Thus, we can use c to uniformly analyze the treatments of examples in various AT methods. Table 1 summarizes different treatments and corresponding test results for some multi-step AT methods from four viewpoints: accuracy learning on A-C examples, accuracy learning on R-C examples, robustness learning on A-C examples and robustness learning on R-C examples. The table shows that some treatments have opposing strategies to others, such as MART and GAIRAT enhancing robustness learning on A-C examples while FAT and TEAT reducing this learning. Hence, it is imperative to conduct additional research on A-C/R-C examples to identify the optimal treatment that confers maximal benefits to AT.

4. Further exploitation on examples

In this section, we investigate the effect of A-C and R-C examples on the accuracy and robustness in AT by designing comparative experiments. We propose a new treatment of examples that addresses how to handle A-C/R-C parts, and provide a simple implementation for its application.

4.1. Effect of A-C and R-C examples

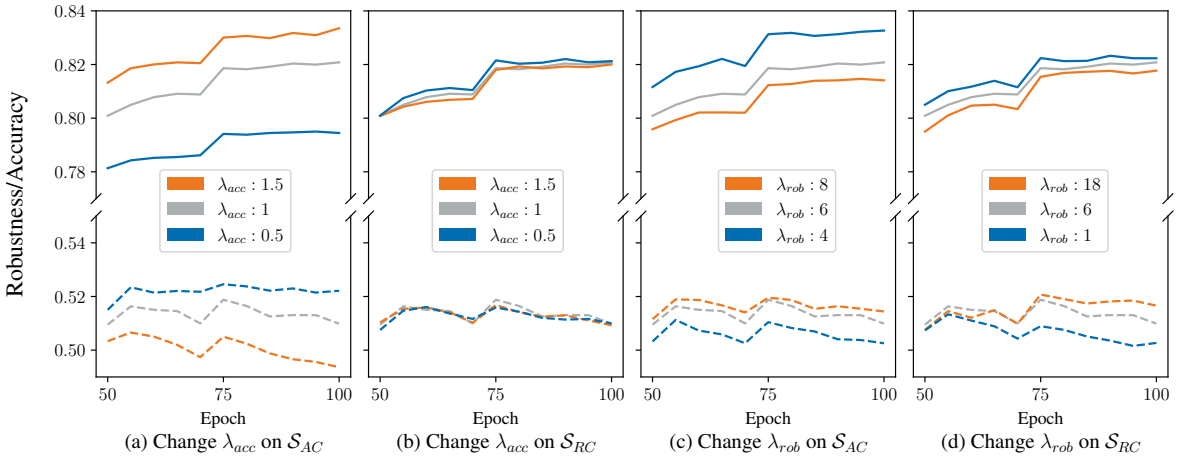
As example-exploitation AT methods treat accuracy-crucial (A-C) and robustness-crucial (R-C) examples differently, it is important to investigate how these two types of examples impact model performance. For this purpose, we utilize TRADES to disentangle the accuracy and robustness. The loss function of TRADES can be rewritten as

$$\lambda_{acc}\cdot \mathrm{CE}\left(f(\boldsymbol{x},\boldsymbol{\theta}),y\right)+\lambda_{rob}\cdot \mathrm{KL}\left(f(\boldsymbol{x},\boldsymbol{\theta})\|f(\boldsymbol{x}',\boldsymbol{\theta})\right), \quad (6)$$

where the weights for CE and KL in Equation (6) are denoted by λ_{acc} and λ_{rob}, respectively. Since CE and KL are responsible for accuracy and robustness learning, increasing (decreasing) λ_{acc}/λ_{rob} means we want the model to learn more (less) features to achieve a higher (lower) accuracy/robustness [27, 3]. Therefore, we can adjust them to investigate the influence on model performance when treating A-C/R-C examples differently.

We use $\mathcal{S}_{AC}$ and $\mathcal{S}_{RC}$ to denote two subsets consisting of 30% training examples with the smallest and largest c_i^{49} (just before the 50_{th} epoch, where the first learning rate decay occurs), respectively. For λ_{acc}/λ_{rob}, we only change them for $\mathcal{S}_{AC}$/$\mathcal{S}_{RC}$ and keep them at default values for the remaining 70% examples (default λ_{acc}/λ_{rob} is 1/6 in TRADES). The main observations from the experiments, shown in Figure 3, are as follows:

1. As shown in Figure 3a, enhancing the accuracy learning on A-C examples (λ_{acc}: 1→1.5) significantly improves the test accuracy but causes severe robust overfitting



(a) Change λ_{acc} on $\mathcal{S}_{AC}$ (b) Change λ_{acc} on $\mathcal{S}_{RC}$ (c) Change λ_{rob} on $\mathcal{S}_{AC}$ (d) Change λ_{rob} on $\mathcal{S}_{RC}$

Figure 3: Test accuracy (solid line) and robustness (dashed line) of models when change λ_{acc} / λ_{rob} on $\mathcal{S}_{AC}$ / $\mathcal{S}_{RC}$ in training after the first learning rate decay occurs (50_{th} epoch). At 75_{th} epoch, the second learning rate decay occurs. The λ_{acc} is 1 and the λ_{rob} is 6 if not specified. We vary λ_{rob} by a larger degree in panel (d) to obtain the change between curves comparable to that in panel (c).

(the dashed orange line shows a noticeable downward tendency). Conversely, reducing the accuracy learning (λ_{acc}: $1 \rightarrow 0.5$) relieves the robust overfitting but brings a large drop in accuracy. This observation is linked to the discovery that in order to improve accuracy, the model needs to memorize A-C examples [28, 29, 30], which will come at the cost of reduced robustness [18]. Therefore, modifying the accuracy learning on A-C examples to enhance the model's accuracy or robustness may not be appropriate for AT.

2. As shown in Figure 3b, changing λ_{acc} for R-C examples has less impact on test performance. This is due to the fact that the model has already generalized R-C examples well in the early stage of training [31]. Therefore, changing the accuracy learning on R-C examples to improve the model accuracy or robustness may have a negligible effect on AT.

3. By comparing the results presented in Figure 3c and 3d, it becomes evident that the trade-off between accuracy and robustness is influenced differently depending on whether we decrease or increase the robustness learning on A-C or R-C examples. Specifically, when the robustness learning is reduced (represented by the blue lines), decreasing λ_{rob} on A-C examples (as shown in Figure 3c) results in more accuracy improvements with fewer sacrifices in robustness compared to decreasing λ_{rob} on R-C examples (as shown in Figure 3d). On the other hand, when enhancing the robustness learning (represented by the orange lines), increasing λ_{rob} on R-C examples (as shown in Figure 3d) leads to more improvements in robustness with less loss in accuracy compared to increasing λ_{rob} on A-C examples (as shown in Figure 3c). These results highlight the distinct impact of A-C and R-C examples on adversarial training, and indicate that the most effective approach for enhancing the accuracy-robustness trade-off is to reduce (or enhance) the robustness learning on A-C (or R-C) examples.

4.2. Reasonableness of new treatment

Based on our observations, we propose an appropriate treatment for examples in AT: **reduce the robustness learning on A-C examples and enhance the robustness learning on R-C examples**. We found that R-C examples have salient features while A-C examples have misleading features, as illustrated in Figure 1. The R-C examples on the right are easily identifiable as true classes, while the A-C examples on the left are challenging even for humans to recognize the inner objects. Moreover, when comparing A-C and R-C examples within a single class (see appendix), we observed that the R-C examples share strong visual similarities, while the A-C examples exhibit diverse visual characteristics. Several works have demonstrated that to defend against adversarial examples, AT should learn more salient features aligned with human perception [27, 32, 33, 34], which supports our proposed treatment of emphasizing robustness learning on R-C examples.

4.3. Application of new treatment

Applying our treatment is easy and flexible. Here we introduce a straightforward way to integrate the treatment into AT. Specifically, we can replace a fixed hyper-parameter of existing methods with an adaptive one to adjust the degree of robustness learning for each example.

Taking multi-step TRADES method for instance, the hyper-parameter λ in its loss (Equation (3)) is fixed for all examples. Using a larger fixed λ will lead to higher robustness but lower accuracy of the model [3]. Guided by our

Algorithm 1: Implementation for the proposed treatment in AT

Require: Training dataset $\mathcal{D}$, model f with parameters $\boldsymbol{\theta}$, attacker A (from TRADES or FastAT), batch size m, number of epochs T, initial warm-up epochs $T_s (T_s > 2)$, $(\lambda_{\min}, \lambda_{\max})$ for multi-step AT, $(a_{\min}, a_{\max})$ for single-step AT

```
// ****** For multi-step AT ******
for t = 1 to T do
    if t ≥ T_s then Calculate c_i^t in Equation (5);
    for mini-batch {(x_1, y_1), ..., (x_m, y_m)} ⊂ D do
        for i = 1 to m (in parallel) do
            if t < T_s then λ_i = λ_min else Calculate λ_i
              in Equation (7);
            δ_i = A(x_i); x_i' = x_i + δ_i;
        end
        Optimize θ in Equation (8) using λ_i;
    end
end
```

```
// ****** For single-step AT ******
for t = 1 to T do
    if t ≥ T_s then Calculate c_i^t in Equation (5);
    for mini-batch {(x_1, y_1), ..., (x_m, y_m)} ⊂ D do
        for i = 1 to m (in parallel) do
            if t < T_s then a_i = a_min else Calculate a_i
              in Equation (9);
            δ_i = A(x_i, a_i); x_i' = x_i + δ_i;
        end
        Optimize θ in Equation (11);
    end
end
```

treatment of examples, replacing the fixed λ, we assign a different λ_i to each training example $\boldsymbol{x}_i$ by linear interpolation according to the position of $\boldsymbol{x}_i$ in the ranking result of all examples using robustness confidence c. At the t_{th} epoch, λ_i can be expressed as

$$\lambda_i = \lambda_{\min} + \frac{\lambda_{\max} - \lambda_{\min}}{M-1} \cdot \left(\text{argsort}_{\boldsymbol{x}_i}(c^t) - 1\right) \quad (7)$$

where M is the number of training examples, $\lambda_{\min}$ and $\lambda_{\max}$ define the range of different λ_i, and argsort determines the rank index of c_i^t in all c^t in ascending order. $\lambda_{\min}$ is the value for the A-C example with the smallest c at every epoch, and $\lambda_{\max}$ is the value for the R-C example with the largest c. The loss function of such updated TRADES is

$$\text{CE}\left(f(\boldsymbol{x}_i, \boldsymbol{\theta}), \boldsymbol{y}_i\right) + \lambda_i \cdot \text{KL}\left(f(\boldsymbol{x}_i, \boldsymbol{\theta}) \| f(\boldsymbol{x}_i', \boldsymbol{\theta})\right). \quad (8)$$

For single-step AT, considering FastAT method as a case, we use adaptive step-size a_i instead the fixed one for each example $\boldsymbol{x}_i$ to generate adversarial perturbation $\boldsymbol{\delta}_i$ in training process:

$$a_i = a_{\min} + \frac{a_{\max} - a_{\min}}{M-1} \cdot \left(\text{argsort}_{\boldsymbol{x}_i}(c^t) - 1\right); \quad (9)$$

$$\boldsymbol{\delta}_i = a_i \cdot \text{sign}(\nabla_{\boldsymbol{x}_i} \text{CE}(f(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i)). \quad (10)$$

Comparing Equation (9) with Equation (7), we can find a_i is calculated in the same way as λ_i and the range of a_i is also determined by two hyper-parameters $a_{\min}$ and $a_{\max}$. The loss function of such updated FastAT is

$$\text{CE}\left(f(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{\theta}), \boldsymbol{y}_i\right). \quad (11)$$

As presented above, we can implement our treatment in AT by simply assigning large/small λ_i and a_i to R-C/A-C examples. The extra computational costs (memory and time) are negligible. Algorithm 1 outlines the training procedures for both implementations.

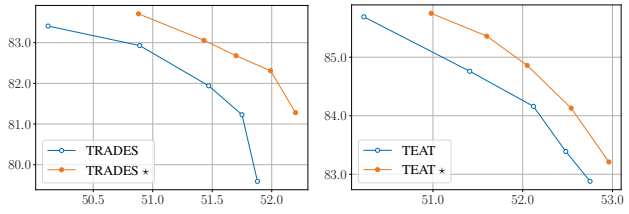
5. Advantages of new treatment

In this section, we apply our treatment to more AT methods and evaluate their performance. Our experimental results corroborate the effectiveness of the proposed treatment in alleviating three critical challenges in AT: the trade-off between accuracy and robustness, robustness overfitting, and catastrophic overfitting.

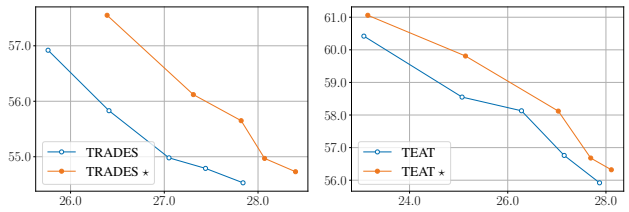
Settings of experiments We conduct experiments on three datasets: CIFAR10, CIFAR100 [21], and TinyImageNet [35]. The models used include PreAct ResNet [22] (for CIFAR10 and CIFAR100) and Wide ResNet [36] (for TinyImageNet). All models are trained for 100 epochs on a single NVIDIA 3090 GPU. To evaluate the robustness, we adopt ℓ_∞-norm PGD [2] and Auto [37] methods to generate adversarial examples. For PGD attack, we use PGD-n to represent PGD attack with n iterations. We perform Auto attack using open-source toolboxes with default settings [37]. We conduct three independent trials with different random seeds and report the means. As the standard deviations are small and have no impact on the results, we omit them from the table to save space. The appendix includes more detailed hyper-parameters as well as additional experimental results and ablation studies.

5.1. Improvement on accuracy-robustness trade-off

As mentioned in Section 4.1, our treatment can improve the trade-off between accuracy and robustness. To confirm this benefit, we update two examples-exploitation AT methods, the classic TRADES [3] and state-of-the-art TEAT [18], based on the approach proposed in Section 4.3, and compare them with their original versions. The trade-off curves for these methods are shown in Figure 4, where we observe that, in comparison with original method, the updated method achieves improved robustness without sacrificing accuracy, and vice versa. These results validate our

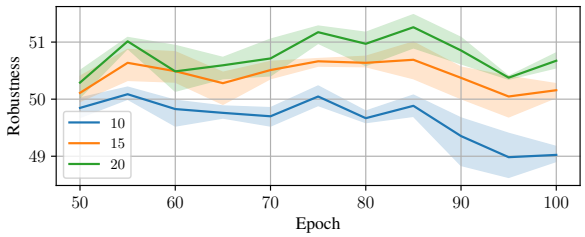


(a) On CIFAR10 testset

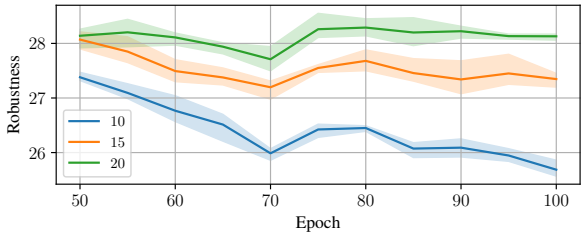


(b) On CIFAR100 testset

Figure 4: The trade-off comparison between original and updated (denoted with $\star$) methods. The robustness is evaluated by PGD-20 attack. We vary the λ/$\lambda_{\min}$, $\lambda_{\max}$ in these methods for better accuracy or robustness. The x-/y-axis indicates the robustness/accuracy.



(a) On CIFAR10 testset



(b) On CIFAR100 testset

Figure 5: Comparison of robust overfitting after the first learning decay happens (50_{th} epoch) with different parameters. The test curves are evaluated by PGD-20 attack. The $\lambda_{\min}$ is fixed to 1 and $\lambda_{\max}$ is chosen from 10, 15, 20.

conclusion that the proposed treatment effectively exploits A-C examples with outlying features to improve accuracy and R-C examples with salient features to improve robustness.

5.2. Inspiration for relieving robustness overfitting

To relieve the issue of robustness overfitting (RO), which is characterized by a slow decline in robustness that typically occurs in the later stage (after the first learning decay happens) of multi-step AT, TEAT hinders the model from excessively memorizing difficult examples. However, we discover that enhancing robust learning for easy R-C examples, can also relieve RO. As depicted in Figure 5, increasing $\lambda_{\max}$ to enhance the robustness learning of R-C examples further relieves robustness degradation in the late training stages. By combining these ideas, we can more intuitively interpret the RO phenomenon: The model expects to learn robustness from R-C examples with salient features, but the memorization effect [28, 18] may lead the model to focus on A-C examples in the later training stage, whose features are inconsistent with those of R-C examples, causing a decline in robustness. As a result, maintaining robustness learning on R-C examples should be prioritized to relieve RO.

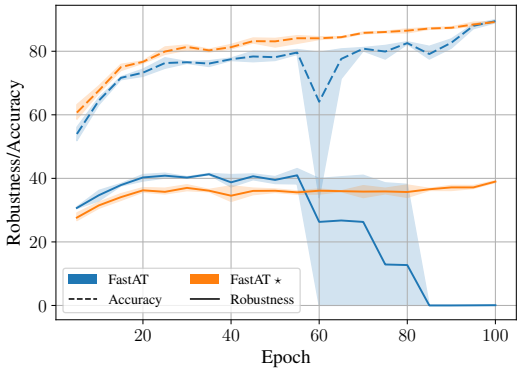
5.3. Effective for preventing catastrophic overfitting

Catastrophic overfitting (CO) is a major challenge in single-step adversarial training, where robustness rapidly degrades after a certain training epoch. Current solutions

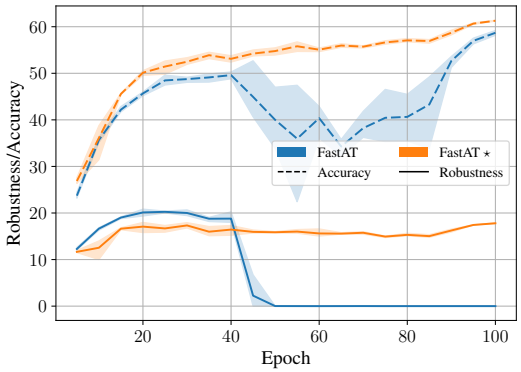
Table 2: Test performance (%) of original and updated (denoted with $\star$) single-step AT methods. The test checkpoint of FastAT is selected before CO happens with the best robustness. We set a to 10 for original methods following their default settings [7, 20]. For the updated methods, we adjust $a_{\min}$, $a_{\max}$ to make one of accuracy and robustness similar to the original method to provide a more obvious comparison.

Dataset	Method	Accuracy	Robustness	
			PGD-50	Auto
CIFAR10	FastAT	76.13	**41.30**	**38.94**
	FastAT ⋆	**89.22**	38.98	37.27
	GradAlign	75.96	42.28	38.41
	GradAlign ⋆	**76.50**	**44.29**	**40.58**
CIFAR100	FastAT	48.47	**20.25**	**16.93**
	FastAT ⋆	**61.29**	17.79	16.75
	GradAlign	48.65	20.26	17.03
	GradAlign ⋆	**49.78**	**20.69**	**17.38**
TinyImageNet	FastAT	34.72	**13.53**	9.64
	FastAT ⋆	**61.03**	13.18	**11.92**
	GradAlign	35.54	14.74	11.28
	GradAlign ⋆	**53.36**	**15.83**	**12.61**

to this problem often require non-trivial modifications to the training process, which come with additional computational burden [20, 25, 11]. However, from the perspective of example exploitation, we find that CO can be prevented



(a) On CIFAR10 testset



(b) On CIFAR100 testset

Figure 6: Comparison of test curves of FastAT method before and after (denoted with $\star$) the update. The robustness curves are evaluated by PGD-50 attack.

almost cost-free. To this end, we update two single-step AT methods, FastAT [7] and GradAlign [20], according to the approach outlined in Section 4.3.

FastAT is a classic method whose proposed random initialization technique has been adopted in subsequent single-step works. However, it still cannot prevent CO definitively. Our treatment can effectively prevent catastrophic overfitting (CO) in FastAT method with 100 training epochs. The results are presented in Figure 6, where the test curves of FastAT before and after the update are compared. It can be observed that FastAT suffers from CO, causing a rapid degradation of robustness to 0% on both CIFAR10 and CIFAR100 datasets. However, after applying our treatment based on example exploitation, FastAT exhibits stable robustness and a significant improvement in accuracy. Specifically, considering the detailed results in Table 2, we observe that the updated FastAT method sacrifices up to 2.46% of its robustness in exchange for the ability to prevent CO. Furthermore, our treatment yields a significant improvement in accuracy, of at least 12.82%, owing to the enhanced contribution of A-C examples. The GradAlign method aims to prevent CO by maximizing the gradient alignment inside the perturbation set. The results in Table 2 demonstrate that our treatment can further enhance its performance in terms of both accuracy and robustness. Our successful application of the treatment to FastAT and GradAlign methods highlights the potential of example-exploitation as a promising approach for preventing CO and enhancing the performance of single-step AT.

6. Discussion of proposed application

When considering the application of existing treatments in different example-exploitation methods, we acknowledge that the treatment we designed in Section 4.3 is the simplest. There is potential to apply our treatment in a more advanced way. For instance, the weight assignment function could be changed from a linear to a more advanced non-linear function, such as $\mathrm{sigmoid}$ or $\tanh$ [17]. Additionally, the adjustment of robustness learning on different examples, which is implemented by controlling the λ parameter for the KL loss in our application, can also be accomplished by varying the attack strength of generated adversarial examples during training [16]. Furthermore, it is feasible to emphasize the contribution of A-C and R-C examples to AT by adding a new regularization term to the loss [15].

The primary contribution in this paper is a treatment for examples that is better suited for adversarial training. Our proposed application approach serves as a validation of the effectiveness of this treatment. Although more advanced application approaches may potentially yield better performance, we did not extensively study them. This is because, for future works seeking to benefit from the exploitation of examples, the application approach may vary depending on the specific use case, but the guiding insight we provide will remain constant.

7. Conclusions

This paper focuses on example-exploitation adversarial training and investigates the unequal contribution of examples to the accuracy and robustness of the model. We identify two types of examples, namely accuracy-crucial and robustness-crucial, which have different roles in adversarial training. We conduct a systematic analysis of the different treatments in various methods and propose a novel treatment that guides both multi-step and single-step AT. Specifically, our treatment reduces robustness learning on accuracy-crucial examples and enhances it on robustness-crucial examples. The experimental results demonstrate that using this treatment in adversarial training can effectively alleviate critical challenges in AT, including the accuracy-robustness trade-off, robust overfitting, and catastrophic overfitting.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. 1

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 6

[3] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019. 1, 2, 4, 5, 6

[4] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020. 1

[5] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 1

[6] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning (ICML)*, 2020. 1

[7] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 2, 7, 8

[8] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[9] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[10] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[11] Tao Li, Yingwen Wu, Sizhe Chen, Kun Fang, and Xiaolin Huang. Subspace adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 7

[12] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. Unlabeled data improves adversarial robustness. In *Advances in neural information processing systems (NeurIPS)*, 2019. 1

[13] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A. Mann. Improving robustness using generated data. In *Advances in neural information processing systems (NeurIPS)*, 2021. 1

[14] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 3, 4

[15] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 2, 4, 8

[16] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning (ICML)*, 2020. 1, 2, 4, 8

[17] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 4, 8

[18] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. Exploring memorization in adversarial training. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 3, 4, 5, 6, 7

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[20] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 7, 8

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images (https://www.cs.toronto.edu/ kriz/cifar.html). 2009. 2, 6

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 6

[23] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017. 3

[24] Chengyu Dong, Liyuan Liu, and Jingbo Shang. Data quality matters for adversarial training: An empirical study. *arXiv preprint*, arXiv:2102.07437, 2021. 4

[25] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *Conference on Artificial Intelligence (AAAI)*, 2021. 4, 7

[26] Zhichao Huang, Yanbo Fan, Chen Liu, Weizhong Zhang, Yong Zhang, Mathieu Salzmann, Sabine Süsstrunk, and Jue Wang. Fast adversarial training with adaptive step size. *arXiv preprint*, arXiv:2206.02417, 2022. 4

[27] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in*

Neural Information Processing Systems (NeurIPS), 2019. 4, 5

[28] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017. 5, 7

[29] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 5

[30] Yangdi Lu, Yang Bo, and Wenbo He. Confidence adaptive regularization for deep learning with noisy labels. *arXiv preprint*, arXiv:2108.08212, 2021. 5

[31] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[32] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 5

[33] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint*, arXiv:1906.00945, 2019. 5

[34] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[35] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. 6

[37] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020. 6