The following publication C. Lin et al., "Hard Adversarial Example Mining for Improving Robust Fairness," in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 350-363, 2025 is available at https://doi.org/10.1109/TIFS.2024.3516554.

Hard Adversarial Example Mining for Improving Robust Fairness

Chenhao Lin, *Member, IEEE,* Xiang Ji, Yulong Yang, Qian Li, *Member, IEEE,* Zhengyu Zhao, *Member, IEEE,* Zhe Peng, *Member, IEEE,* Run Wang, *Member, IEEE,* Liming Fang, *Member, IEEE,* Chao Shen, *Senior Member, IEEE*

***Abstract*—Adversarial training (AT) is widely considered the state-of-the-art technique for improving the robustness of deep neural networks (DNNs) against adversarial examples (AEs). Nevertheless, recent studies have revealed that adversarially trained models are prone to unfairness problems. Recent works in this field usually apply class-wise regularization methods to enhance the fairness of AT. However, this paper discovers that these paradigms can be sub-optimal in improving robust fairness. Specifically, we empirically observe that the AEs that are already robust (referred to as "easy AEs" in this paper) are useless and even harmful in improving robust fairness. To this end, we propose the hard adversarial example mining (HAM) technique which concentrates on mining hard AEs while discarding the easy AEs in AT. Specifically, HAM identifies the easy AEs and hard AEs with a fast adversarial attack method. By discarding the easy AEs and reweighting the hard AEs, the robust fairness of the model can be efficiently and effectively improved. Extensive experimental results on four image classification datasets demonstrate the improvement of HAM in robust fairness and training efficiency compared to several state-of-the-art fair adversarial training methods. Our code is available at https://github.com/yyl-github-1896/HAM.**

***Index Terms*—Adversarial training, robust fairness, hard adversarial example mining, convolutional neural network.**

I. Introduction

Deep Neural Networks (DNNs) have rapidly advanced, reaching a level beyond human intelligence in many areas. However, several studies [1], [2] have discovered that when exposed to specifically designed imperceptible perturbations added to the original inputs, known as adversarial examples (AEs), the accuracy of DNNs can drop dramatically.

This work was supported in part by the National Key Research and Development Program of China (2023YFE0209800), the National Natural Science Foundation of China (T2341003, 62376210, 62161160337, 62132011, U21B2018, U20A20177, 62206217), the Shaanxi Province Key Industry Innovation Program (2023-ZDLGY-38), the Hong Kong RGC GRF Projects (12202922, 15238724), and Shenzhen Science and Technology Program (JCYJ20230807140412025). Corresponding author: Chao Shen (chaoshen@mail.xjtu.edu.cn).

Chenhao Lin, Yulong Yang, Qian Li, Zhengyu Zhao, and Chao Shen are with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China.

Xiang Ji is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an, China.

Zhe Peng is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, and The Hong Kong Polytechnic University Shenzhen Research Institute

Run Wang is with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China.

Liming Fang is with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Various approaches have been proposed to enhance the defense capabilities of DNNs against AEs. Among these, adversarial training (AT) has been demonstrated as one of the most effective strategies [3]. Nevertheless, recent research [4], [5] have identified that the adversarially trained models usually suffer from a serious unfairness problem, i.e., adversarially trained models exhibit a noticeable disparity in both clean accuracy and robust accuracy across different classes, even when the training data distribution is class-wise balanced. Additionally, AT methods incur substantial computational overhead, typically requiring $10\times$ more resources than standard training methods. The above intriguing phenomenon and challenges restrict the applicability of AT methods in real-world applications.

Many recent works address the issue of robust fairness by regularizing the class-wise fairness disparity during adversarial training [4]–[7]. However, this paper reveals that these recent class-wise regularization techniques are not optimal. Instead and interestingly, we find that the root of the robust fairness issue in DNN models lies at the sample level. By reweighting the training examples, we can mitigate this issue, even though our method does not explicitly regularize class-wise disparity as previous works do.

To enhance the fairness of AT, we propose the following motivations: (1) The computational cost of AT arises because it requires the model to prioritize the optimization of the already robust AEs (referred to as "easy AEs" in this paper), rather than non-robust ones (referred to as "hard AEs"). Easy AEs can interfere with the optimization of hard AEs, which finally leads to fairness issues. (2) By dropping the easy AEs, both the robust fairness and training efficiency of the model can be improved.

Specifically, we distinguish the easy adversarial examples (easy AEs) as the AEs that can still be correctly classified after the first M steps of an adversarial attack. The hard adversarial examples (hard AEs) are the rest of the AEs. Through formal analysis and experimental investigation, we first observe a strong positive correlation between the proportion of easy AEs of a certain class and the severity of the unfairness problem for that class. In other words, classes with a higher portion of easy AEs have a higher robust accuracy. This observation motivates us to drop the easy AEs to solve the fairness issue. Second, we also observe that the robust overfitting issue of adversarial training is caused by the conflict of the training gradients between easy AEs and hard AEs. By dropping the easy AEs, the training conflicts will be mitigated and the

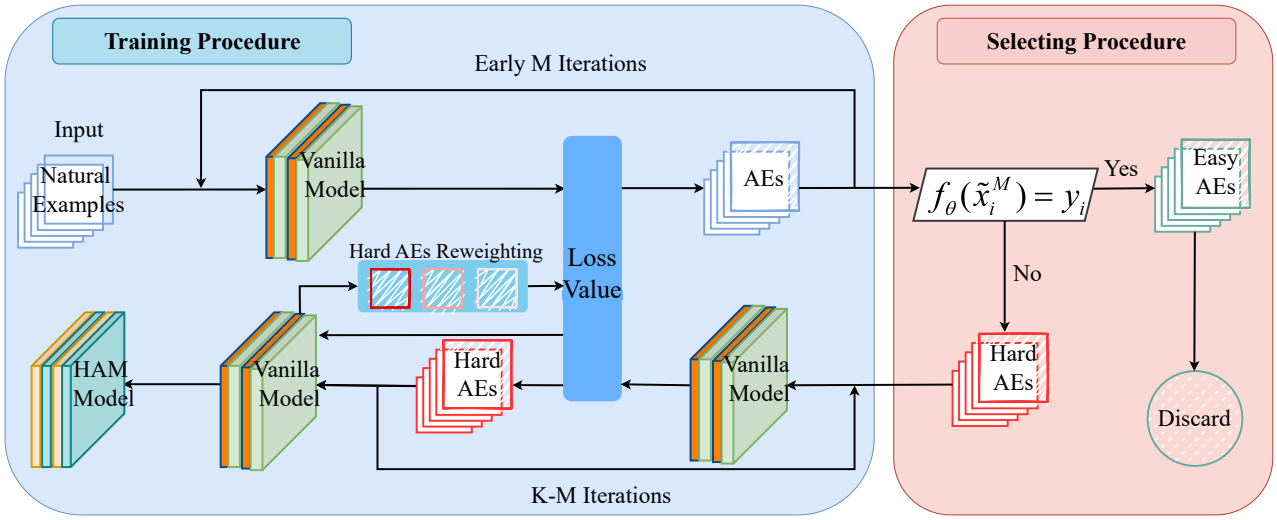


Fig. 1. Pipeline of HAM. The easy and hard AEs are distinguished in terms of the PGD attack with M iterations. After that, the easy AEs are discarded and the hard AEs are reweighted with the step size-based reweighting module.

overall robust accuracy of the adversarially trained models can be preserved. Besides, dropping the easy AEs can also speed up the training convergence. The detailed analysis and investigation can be found in *Section III*. In sum, dropping the easy AEs is beneficial in addressing both robust overfitting and robust fairness issues, which motivates us to propose the Hard Adversarial example Mining (HAM) approach inspired by the sample mining techniques [8]–[10].

HAM is an efficient adversarial training technique that improves robust fairness without increasing the training overhead. As illustrated in Figure 1, HAM firstly distinguishes easy AEs and hard AEs with a fast adversarial attack method, namely the M-step PGD attack, in which examples unable to cross the decision boundary would be identified as easy AEs. These identified easy AEs would be discarded from the subsequent training process. The preserved hard AEs will be re-weighted in the loss function to place different priorities on the hard AE training examples. The re-weighting factors are computed in terms of the attack strength (measured by the logits value difference) such that the HAM training process would place even higher priority on the most challenging hard AEs. Besides, dropping easy AEs makes HAM an efficient training algorithm that reduces the training time by 45% compared to conventional PGD adversarial training [3].

To comprehensively evaluate the effectiveness of HAM, we conduct experiments on four popular datasets, including CIFAR-10, CIFAR-100, SVHN, and Imagenette. The experimental results show that, compared to state-of-the-art class-wise fair adversarial training methods, HAM can achieve better clean fairness as well as robust fairness with less computational overhead. For example, compared to the conventional PGD-AT on CIFAR-10, the HAM reduces the maximum class Discrepancy (a fairness metric, lower is better) by more than 20% while reducing the overall training time cost by 45%. Besides, HAM is a plug-n-play module that can be combined with other adversarial training frameworks like MART [11], and AWP [12] to fix their fairness issues.

In sum, our contributions are as follows:

- We reveal that the class-wise regularization techniques adopted in recent fair adversarial training methods are sub-optimal and propose the hard adversarial example mining (HAM) method to improve the robust fairness of the model from the sample level.
- The proposed HAM method first distinguishes easy AEs and hard AEs by applying a fast adversarial attack. The easy AEs are dropped and the hard AEs are reweighted. The HAM can enhance robust fairness while saving the computational cost.
- We conduct experimental evaluations on four datasets, and the results show that our HAM significantly outperforms state-of-the-art fair adversarial training methods in terms of robust fairness and computational cost without sacrificing the overall adversarial robustness.

II. Related Work

A. Adversarial training

Adversarial training is widely recognized as the most effective defense against adversarial attacks [13]. The mainstream AT methods typically use PGD attack [3] to generate AEs and various methods have been proposed to improve its performance from different aspects. For instance, TRADES [14] divides the original loss function into two parts, representing the accuracy for a clean sample and the robustness for malicious disturbance. According to the entropy of its predicted distribution, Entropy entropy-weighted AT scheme [15] reweighs each AE to increase the robustness accuracy. HAT [16] proposes to generate an additional sample with a larger perturbation and give it an "error" label as a helper for achieving better robustness. DAT [17] explores the use of discrete adversarial training methods to improve visual feature robustness. Wang et al. [18] demonstrate how diffusion models could enhance robustness in adversarial settings by generating more complex adversarial examples. Jin et al. [19] use the small Gaussian noise and Taylor expansion method to generate

random weights, which improves the robustness of the neural network and has a good effect in balancing robustness and correctness. UIAT [20] proposes using inverse adversarial examples to improve robustness by leveraging adversarial features that aid learning in adversarial training. Additionally, HFAT [21] introduces an iterative evolutionary optimization strategy to streamline the optimization process. It incorporates an auxiliary model to effectively expose hidden adversarial examples, integrating the optimization pathways of traditional adversarial training with a strategy to mitigate the effect of concealed attacks.

Despite their successes, these approaches generally fail to address the critical issues of fairness and computational overhead in AT. High computational cost and fairness disparity, where the accuracy and robustness vary significantly across different classes, remain significant challenges for the widespread applicability of adversarial training.

B. Fairness issue of AT methods

Recent studies have observed a serious fairness problem in AT, where there is a noticeable disparity in accuracy and robustness across different classes of adversarially trained models. To address this issue, recent works have typically employed class-wise regularization techniques to enhance the fairness of AT models. For instance, Sun et al. [22] demonstrate that the trade-off between fairness, robustness, and model accuracy can introduce a great challenge for robust deep learning. They propose a fair and robust classification method by modifying the input data and models. Xu et al. [4] empirically find the serious deficiency of AT in fairness and attempt to mitigate this problem using the proposed fair robust learning (FRL) framework. More recently, Sun et al. [6] propose balance adversarial training (BAT) to improve robust fairness by balancing the source-class and target-class fairness. Ma et al. [7] theoretically study the trade-offs between adversarial robustness and class-wise fairness, and a fairly adversarial training (FAT) method is proposed to mitigate the unfairness problem. WAT [23] focuses on the worst class during the adversarial training, ultimately improving overall robustness across all classes. FAAL [24] introduces a new fairness-aware adversarial learning paradigm to address robust fairness through distributional robust optimization. We provide a detailed description and analysis of the above fair AT methods in *Appendix A*.

Although several approaches have been proposed to study the robust unfairness problem in AT, the fairness of the adversarially trained models is still not satisfying and needs further improvement. Our contribution lies in demonstrating that a sample-level reweighting paradigm can be more effective and efficient than recent class-wise regularization methods in enhancing the robust fairness of models. By focusing on reweighting individual samples rather than applying broad class-wise adjustments, our approach provides a finer-grained and more efficient solution to the fairness issue in AT.

C. Efficient adversarial training

To reduce the huge amount of computation consumed by multiple iterations of PGD during AT, several methods have been proposed [25]–[27]. For instance, Free AT [28] recycles the gradient information when updating model parameters during AT to improve the AT efficiency. Fast AT [29] finds that the specifically designed FGSM AT can achieve comparable performance with PGD AT while reducing the training overhead. YOPO [25] significantly reduces the computational burden by restricting most forward and backward propagations to the first layer during adversarial updates. ATTA [30] enhances the efficiency and effectiveness of adversarial training by leveraging the high transferability of adversarial examples between neighboring epochs. While these methods significantly reduce the adversarial training time, they often do so at the expense of sacrificing the fairness and robustness of the AT models. Instead, our proposed method focuses on improving the robust fairness and computational efficiency of AT without compromising its overall robustness.

D. Hard negative mining

Hard negative mining techniques [31], [32] have been extensively utilized in object detection to address the issue of imbalance in the proportion of positive and negative samples. The online hard example mining (OHEM) [8] method extends hard negative mining by using an online selection process with both hard negatives and hard positives, which reduces the training time considerably and improves the performance.

Recent research highlights advancements in example mining and balancing methods that could further inform adversarial training strategies. For instance, Cui et al. [33] propose a continual representation learning approach to address compatibility across Lifelong Person Re-identification tasks, which could inspire methods for adaptive example mining in dynamic environments. Additionally, Xia et al. [34] introduce a high-discrepancy sample selection method for noisy labels. Furthermore, Waseda et al. [35] analyze the different kinds of prediction mistakes caused by transfer attacks, revealing that the difference can be attributed to non-robust features exploited uniquely by each model.

The success of the hard example mining mechanism [36], [37] across various tasks has motivated us to investigate adversarial training from the perspective of example mining. This paper applies the hard example mining techniques in adversarial training and focuses on the hard adversarial examples during AT training to address the fairness issue and reduce the training complexity.

III. Methodology

A. Preliminaries

This fairness issue denotes the phenomenon that the average performance (prediction accuracy, latency, robustness, etc.) of a machine-learning model on a certain group of samples is largely different from that of the other groups, while the property of that group should not be regarded as a reasonable judgment for making a decision (for instance, age, gender, race, etc.). The robust fairness issue denotes that the robustness of a certain group of samples is largely diverged from other samples. This paper focuses on the robust fairness issue of adversarially trained image recognition models, which have

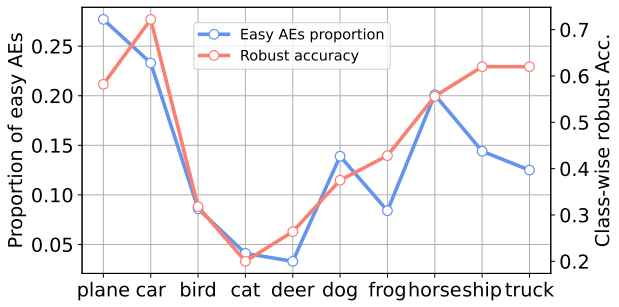


Fig. 2. For an AT model, classes with a higher proportion of easy AEs have higher robustness in most cases.

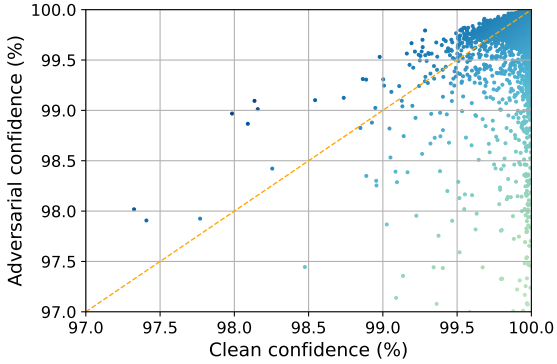


Fig. 3. Illustrattion of the robust overfitting issue on an AT model on CIFAR-10. Confidence scores of adversarial examples and corresponding clean examples in adversarially trained WideResNet-34-10. Darker points mean that it has much larger adversarial confidence than clean confidence. Brighter points represent examples with larger clean confidence than adversarial confidence.

also been widely adopted by recent fairness studies. For a DNN model, the fairness requirement can be formalized as that the average accuracy of a certain class c should not be largely diverged from the whole average accuracy:

$$|\frac{\sum_i^{n_c} \mathcal{I}(f_\theta(x_{i,c}) = c)}{n_c} - \frac{\sum_i^{n} \mathcal{I}(f_\theta(x_i) = y_i)}{n}| \leq \epsilon_1, \quad (1)$$

where n_c is the number of test samples of the c-th class, $\mathcal{I}$ is the counting function, n is size of the whole test set, and the ϵ_1 and ϵ_2 are pre-defined thresholds. Similarly, the robust fairness requirement can be formalized as the average robustness disparity of a certain class c from the average robustness:

$$|\frac{\sum_i^{n_c} \mathcal{I}(f_\theta(x_{i,c} + \delta_{i,c}) = c)}{n_c} - \frac{\sum_i^{n} \mathcal{I}(f_\theta(x_i + \delta_i) = y_i)}{n}| \leq \epsilon_2, \quad (2)$$

where δ denotes the generated adversarial perturbation.

B. Our motivation

Our motivation is to distinguish between easy AEs and hard AEs during the AT process and explore how mining these AEs can address the fairness and robust overfitting issues in AT. This subsection first defines easy/hard AEs and then provides analysis and numerical results to support the correlation between easy/hard AEs and the above two issues.

Defination of easy/hard AEs. We define the **easy adversarial examples (easy AEs)** as the AEs that can still be correctly classified after the first few steps of an adversarial attack. The **hard adversarial examples (hard AEs)** are the rest of the AEs. Intuitively, to enhance the robust fairness of a model, one direct way is to set higher priorities on the non-robust examples in the low-robustness classes (in other words, hard examples) and turn them into easy examples. In the following, we will show why this sample-level reweighting strategy is more effective than the previously proposed class-wise reweighting strategies in improving robust fairness.

Correlation with the fairness issue. The objective towards a fair and robust model can be formalized as [4]:

$$\begin{aligned} \min_{\theta,\phi} \mathcal{L}(f_\theta, \phi) &= \mathcal{R}_{nat}(f_\theta) + \mathcal{R}_{bndy}(f_\theta) \\ &+ \sum_{i=1}^{Y} \phi^i_{nat}(\mathcal{R}_{nat}(f_\theta, i) - \mathcal{R}_{nat}(f_\theta) - \tau_1) \\ &+ \sum_{i=1}^{Y} \phi^i_{bndy}(\mathcal{R}_{bndy}(f_\theta, i) - \mathcal{R}_{bndy}(f_\theta) - \tau_1), \end{aligned} \quad (3)$$

where the $\mathcal{R}_{nat}$ and the $\mathcal{R}_{bndy}$ denote the clean error rate and the boundary error rate, respectively (Please refer to [4] for a detailed definition of them). The ϕ denotes the Lagrangian multiplier, the Y denotes the total number of classes, and the τ denotes the pre-defined threshold.

Please recall that the definition of the easy example, then we can discover that easy examples contribute zero to the $\mathcal{R}_{nat}(f_\theta, i)$ and $\mathcal{R}_{bndy}(f_\theta, i)$, in other words, training on easy examples will not improve the robust fairness. On the contrary, the gradient of the easy examples may contradict that of the hard examples, leading to an inferior robust fairness. The shortage of class-wise reweighting methods like FRL is that they do not consider the contradiction between easy samples and hard examples within the same class, which can be fixed by dropping easy AEs. We verify the above analysis with the following two experimental observations.

(1) The proportion of easy AEs of a certain class shows a strong positive correlation with the severity of the unfairness problem for that class. Figure 2 plots the distribution of easy AEs and robust accuracy across different classes in a PGD-AT model on CIFAR-10. We can observe that the class-wise distribution of fairness and robust overfitting are highly consistent. For instance, the plane and car classes exhibit the highest proportion of easy AEs, along with the highest robust accuracy, whereas the cat and deer classes show the lowest proportion of easy AEs and the lowest robust accuracy. Similar trends were observed in more advanced AT models (detailed in *Appendix C*), where the robust overfitting issue was even more pronounced. We suppose that the class-wise AE confidence disparity causes AT to prioritize classes with a higher ratio

TABLE I
THE ERROR RATE OF WORST-CLASS STANDARD AND ROBUST RESULTS OF TRAINING WITH CONFIDENCE-WISE AES DROP.

Drop rate (%)	0 (full-AT)	10	20	30
Worst Std. (%)	33.0	31.6	33.0	30.4
Worst Rob. (%)	85.8	82.8	85.0	85.2

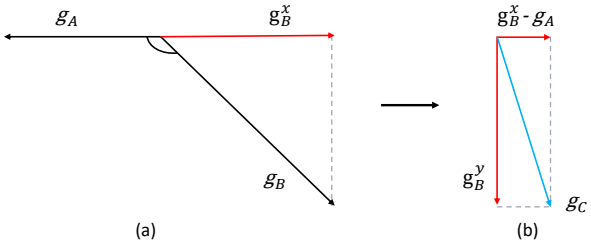


Fig. 4. (a) The angle between g_B and g_A is obtuse, g_B will produce a component in the horizontal. (b) The horizontal component generated by g_B cancels out g_A, leading to the final gradient direction tending towards g_B.

of easy AE while neglecting others, resulting in the class-wise robust unfairness. Numerical evidence will be provided to support this hypothesis.

(2) We verify the above hypothesis by showing that the fairness issue in AT models can be mitigated by simply dropping easy AEs. We conducted adversarial training and reported both the worst class standard error rate and the worst class robust error rate with varying drop rates of easy AEs in Table I. A lower worst-class error rate denotes better fairness. The results show that dropping easy AEs enhances both clean and robust fairness, verifying our motivation that the distinction between easy and hard AEs is strongly correlated with fairness issues in AT models.

Correlation with the robust overfitting issue. Robust overfitting occurs when in an AT model, the prediction confidence of AEs is even higher than that of the corresponding clean examples. We qualitatively illustrate this phenomenon in Figure 3, which plots the confidence of AE (denoted as adversarial confidence) and the corresponding clean confidence of each test example on an AT model using CIFAR-10. Points above the diagonal line are over-confident AEs where adversarial confidence is even higher than their clean confidence. We can observe that a large number of examples fall above this line, indicating the widespread occurrence of over-confident AEs during AT. We hypothesize that the interference between easy and hard AEs during training exacerbates this issue, and propose addressing it by discarding easy AEs.

(1) We hypothesize that the training gradients of easy and hard AEs are in conflict, leading to the robust overfitting issue in AT models. This conflict can be measured using cosine similarity [38], where two gradients are considered contradictory to each other if and only if their cosine similarity is below zero, i.e., the vector angle between gradient A (g_A) and B (g_B) is obtuse, as illustrated in Figure 4. Specifically, we exhaustively compute the gradient cosine similarity of each <easy AE, hard AE> pair and the average gradient cosine similarity is -0.1105. This negative value indicates that the angle between easy and hard AE gradients is obtuse,

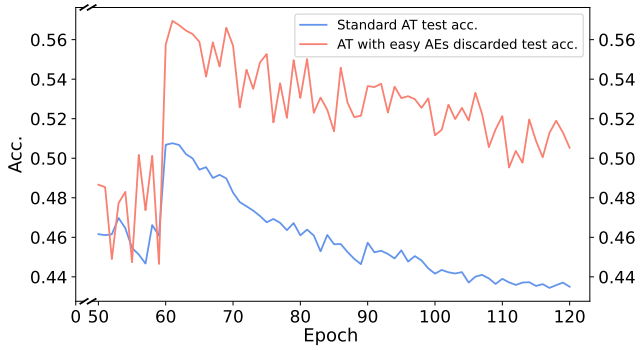


Fig. 5. The model trained by AT with easy AEs discarded has a faster convergence speed and a better robust generalization capability than the one trained by standard AT.

which means that these two training gradients are conflicted, supporting our hypothesis of gradient contradiction.

(2) The gradient contradiction can be addressed by simply discarding the easy AEs, which leads to a faster convergence speed and better robust generalization. We measure the robust generalization with the robust accuracy on the CIFAR-10 test set. As shown in Figure 5, under the same training conditions (120 epochs), AT with easy AEs discarded achieves significantly better robust generalization (50.53%) compared to standard AT (43.50%). These results demonstrate that discarding easy AEs effectively addresses the gradient contradiction issue in AT models, leading to improved performance.

C. HAM overview

We consider a robust classification task with a given dataset $X = \{x_i | i = 1, 2, \cdots, N\}$ and a perturbation budget ϵ. AT is expected to solve the following min-max objective:

$$\min_{f_\theta} \sum_{i=1}^{N} \max_{\|\tilde{x}_i^K - x_i\|_p \leq \epsilon} \mathcal{L}(f_\theta(\tilde{x}_i^K), y_i), \tag{4}$$

where $f_\theta(\cdot)$ is the DNN parameterized with θ, y_i is the true label corresponding to input x_i, $\mathcal{L}(\cdot)$ is the loss function, $\|\tilde{x}_i^K - x_i\|_p$ is the l_p-norm used to bound the adversarial perturbation ($p = 2$ or ∞), and $\tilde{x}_i$ is the adversarial example corresponding to clean example x_i. Take PGD-AT for instance, which adopts a K-step PGD attack iterations process to obtain the $\tilde{x}_i^K$, which can be formalized as:

$$\tilde{x}_i^{j+1} = Clip_\epsilon(\tilde{x}_i^j + \alpha \cdot sign(\nabla_{\tilde{x}_i^j} \mathcal{L}(f_\theta(\tilde{x}_i^j), y_i))), \tag{5}$$

where α is the step size of each iteration, and $Clip(\cdot)$ is the projection operation that guarantees $\tilde{x}_i$ is within the l_p-ball.

The intuition of the HAM is to reweight AEs in the adversarial training procedure, whose training objective can be formalized as:

$$\min_{f_\theta} \sum_{i=1}^{N} hard(x_i, y_i, f_\theta) \cdot \mathcal{L}(f_\theta(\tilde{x}_i^K), y_i), \tag{6}$$

where $hard(x_i, y_i, f_\theta)$ is the weight for each AE.

Figure 1 illustrates the pipeline of our proposed HAM framework. HAM judges an AE as easy or hard in terms of

whether the AE crosses the decision boundary in M attack iterations. Hard AEs are utilized in the subsequent training procedure, while easy AEs will be discarded. Different weights are allocated to each hard AE when calculating the loss function. To further save the computational cost of HAM, an early-dropping mechanism is applied to the easy AEs, which prevents them from following training. The early-dropped easy AEs have zero weights in Equation 6.

D. Hard adversarial example mining

HAM consists of an easy AE early-dropping stage and a hard AE reweighting stage. Easy AEs and hard AEs are distinguished in the first stage to prevent them from wasting the computational budget. Hard AEs are utilized and reweighted in the second training stage.

Easy AE early-dropping. In the early-dropping stage, given the total number of PGD iterations K, HAM first identifies easy AEs with M-step PGD ($M < K$). AEs that fail to cross the decision boundary within the M PGD attack steps are identified as easy AEs, which will be dropped in this training epoch and will not participate in the following $(K - M)$-step AE generation process. The easy AE early-dropping mechanism saves the computational budget and prevents AT from paying too much attention to the already-robust easy AEs.

Hard AE reweighting. HAM reweights the hard AEs in terms of the following metric:

$$hard(x_i, y_i, f_\theta) = \begin{cases} \omega(\max\limits_{1\le j\le K} \left\| \Delta f_\theta(\tilde{x}_i^j) \right\|_1), f_\theta(\tilde{x}_i^M) \neq y_i \\ 0, f_\theta(\tilde{x}_i^M) = y_i \end{cases} \tag{7}$$

where $\Delta f_\theta(\tilde{x}_i^j) = f_\theta(\tilde{x}_i^{j+1}) - f_\theta(\tilde{x}_i^j)$ is the logits variant between two adjacent attack steps, $\omega(\cdot)$ is a monotonically increasing function $\omega(z) = sigmoid(z + \lambda)$ with hyperparameter λ, $\tilde{x}_i^j$ represents the M-step AE used to distinguish the easy AE and hard AE. The AEs that fail to cross the decision boundary ($f_\theta(\tilde{x}_i^M) = y_i$) will be identified as easy AEs and will be assigned zero weights.

The hard AEs with non-zero weights are utilized in the adversarial training. As illustrated in Figure 6, hard AEs with larger maximum adversarial step-sizes (adversarial step-sizes denote the logits variants between two adjacent adversarial attack steps) will be assigned with larger weights, and vice versa. We adopt this reweighting strategy because AEs with different maximum step sizes pose varying threat levels to the model. For instance, an AE with a large maximum adversarial step size is much more vulnerable to the model, even when facing a single-step attack like FGSM. Thus, they deserve more attention in the training process. Besides, we do not directly use the predicted score or cross-entropy loss for reweighting because a hard example with a higher predicted score has a large loss value of its own. If assigned a larger weight, the model is prone to overfitting it. Our reweighting method in Eq.7 prevents such overfitting.

We summarize the proposed algorithm in Algorithm 1. In addition, HAM is a general plug-n-play technique that can be integrated with existing AT-based frameworks to mitigate the adversarial confidence overfitting issue and thus offers advantages in terms of both fairness and computational cost, as validated in *Appendix D*.

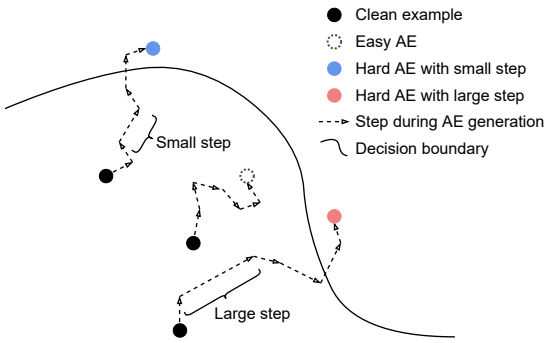


Fig. 6. The intuition behind HAM.

Algorithm 1 Pseudo-code of HAM

Input: Network f_θ, data $S = \{(x_i, y_i)\}_{i=1}^n$, batch size n_{bs}, number of batches n_b, learning rate η, training epochs T, whole attack step K, early-dropping step M

Output: Robust model f_θ

for epoch = $1, \cdots, T$ **do**
  Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{n_{bs}}$ from S
  for mini-batch = $1, \cdots, n_b$ **do**
    Generate $\{\tilde{x}_i^M\}_{i=1}^{n_{bs}}$ with M-step PGD
    Construct hard AE set: $H_M = \{\tilde{x}_j^M : f_\theta(\tilde{x}_j^M) \neq y_j\}$
    Finish hard AE generation with $(K-M)$ step PGD: $H_K = \{\tilde{x}_j^K : \tilde{x}_j^M \in H_M\}$
    Update $hard(x_i, y_i, f_\theta), i = 1, \cdots, n_{bs}$ by Equation 7
    Update θ with SGD by Equation 6
  end for
end for

Fairness benefits of HAM. As has been discussed, the class-wise unfairness problem can be attributed to the phenomenon that AT pays more attention to the classes with higher ratios of over-confident AEs and less attention to other classes. As illustrated in Figure 2, the largely diverged proportion of over-confident easy AEs of each class and the robust overconfidence issue are the root causes of the robust unfairness problem. By mitigating the adversarial overconfidence issue with sampling mining techniques, HAM thus makes AT pay enough attention to the less-confident class and mitigates the robust unfairness problem.

Efficiency benefits of HAM. The early-dropping mechanism in HAM effectively reduces computational expenses of AT while maintaining unaltered robust fairness, which is supported by the results in Figure 7. We can see that the vast majority of AEs that successfully attack the model succeed within the initial steps. This indicates that the early-dropping mechanism can identify easy AEs with a low false positive rate, which in turn does not compromise the final model's robust fairness. As a result, early-dropping easy AEs will

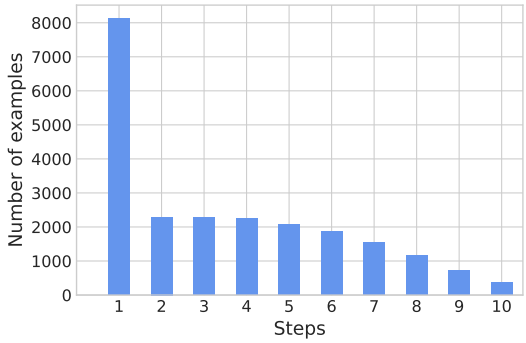


Fig. 7. Number of examples with different minimal PGD steps for a successful attack.

significantly improve AT efficiency.

IV. Experiments

To evaluate the proposed HAM, we perform several experiments on four widely adopted image datasets CIFAR-10, SVHN [39], Imagenette, and CIFAR-100. Section IV-B evaluates the fairness improvement of HAM models. Section IV-C presents the efficiency results of HAM. Parameter sensitivity results are in the *Appendix E*.

TABLE II
Error rate(%) of average & worst-class standard, boundary, and robustness for our HAM and other methods on CIFAR-10, SVHN, and Imagenette. The best results are in **bold**, and the second best results are marked with underlines.

Dataset	Method	Avg. Rob. (↓)	Worst Rob. (↓)	Maximum class discrepancy (↓)
CIFAR-10	PGD-AT	56.62 (±0.11)	85.50 (±0.25)	51.20 (±0.28)
	FRL	56.37 (±0.21)	80.10 (±0.35)	44.30 (±0.36)
	BAT	49.84 (±0.16)	75.90 (±0.28)	50.80 (±0.31)
	FAT	56.37 (±0.24)	83.90 (±0.32)	51.10 (±0.38)
	CFA	**44.10** (±0.31)	66.40 (±0.44)	44.90 (±0.47)
	HAM	48.05 (±0.19)	**64.20** (±0.26)	**30.90** (±0.29)
SVHN	PGD-AT	49.13 (±0.27)	65.90 (±0.63)	31.81 (±0.72)
	FRL	49.62 (±0.31)	64.27 (±0.52)	33.64 (±0.66)
	BAT	**34.28** (±0.42)	57.40 (±0.21)	46.11 (±0.42)
	FAT	58.30 (±0.24)	75.60 (±0.46)	**29.30** (±0.53)
	CFA	44.70 (±0.21)	60.78 (±0.54)	37.42 (±0.67)
	HAM	38.33 (±0.32)	**47.04** (±0.31)	32.93 (±0.43)
Imagenette	PGD-AT	64.48 (±0.43)	79.23 (±0.24)	40.00 (±0.54)
	FRL	66.40 (±0.32)	78.75 (±0.43)	39.78 (±0.46)
	BAT	63.85 (±0.42)	88.08 (±0.73)	51.67 (±0.32)
	FAT	63.44 (±0.43)	80.05 (±0.55)	38.25 (±0.58)
	CFA	66.33 (±0.24)	84.48 (±0.31)	**29.30** (±0.52)
	HAM	**60.99** (±0.42)	**73.03** (±0.51)	36.87 (±0.42)

A. Experimental setup

All the following experiments are performed on an Ubuntu 20.04 System with Intel Xeon Gold 6226R CPUs and RTX 3090 GPUs. When evaluating the training efficiency, we use a single GPU for a fair comparison. The deep learning framework we use is PyTorch 1.9. Our experiments were repeated three times, and the mean values are reported in II. We list the hyper-parameter settings on each dataset as follows.

CIFAR-10 & Imagenette. The experiments on CIFAR-10 and Imagenette share the same experimental settings. For all the AT methods compared in this section, we trained PreActResNet-18 [40] for 120 epochs with a batch size of 128. The optimizer is SGD with a momentum of 0.9 and a weight decay of 2×10^{-4}. The initial learning rate is 0.1. At the $60th$, $90th$, and $110th$ epochs, the learning rate is decayed to 10%, 1%, and 0.5%, respectively. We adopt the commonly used reprocessing and augmentation methods. Images on CIFAR-10 are normalized to $[0, 1]$ and augmented with random crop and random flip. We use the l_∞-norm PGD attack with the perturbation budget of $8/255$ to generate adversarial examples. 10-step PGD is used in the training stage and 20-step PGD is used in the test stage, which is consistent with previous works [13], [41]–[43]. The step size of the PGD attack is $2/255$. Our HAM method is started only after the 50th epoch, and the early-dropping hyper-parameter $M = 3$.

SVHN. For SVHN, the initial learning rate is 0.01. The early-dropping hyper-parameter of HAM is set to $M = 5$. All the other settings remain the same as on CIFAR-10.

Baseline method. We calculate the fairness and efficiency and compare our HAM with traditional PGD-AT (denoted as AT in the following) as well as several state-of-the-art fair adversarial training methods, including FRL [4], BAT [6], FAT [7] and CFA [44]. All these methods follow their original settings.

Evaluation metric. For evaluating the fairness, we follow the previous work [4], [6], [45] to use the maximum class-wise discrepancy (which represents the discrepancy of the maximum and minimum accuracy of class), and the worst class error rates (Worst Std., Worst Bndy. and Worst Rob.). Models with a lower maximum class-wise discrepancy and lower worst-class error rates are fairer.

$$Worst\ Std. = \max_{i\in N}(1 - Acc._{std}^{i}), \tag{8}$$

$$Worst\ Rob. = \max_{i\in N}(1 - Acc._{rob}^{i}), \tag{9}$$

$$Worst\ Bndy. = \max_{i\in N}((1 - Acc._{rob}^{i}) - (1 - Acc._{std}^{i})), \tag{10}$$

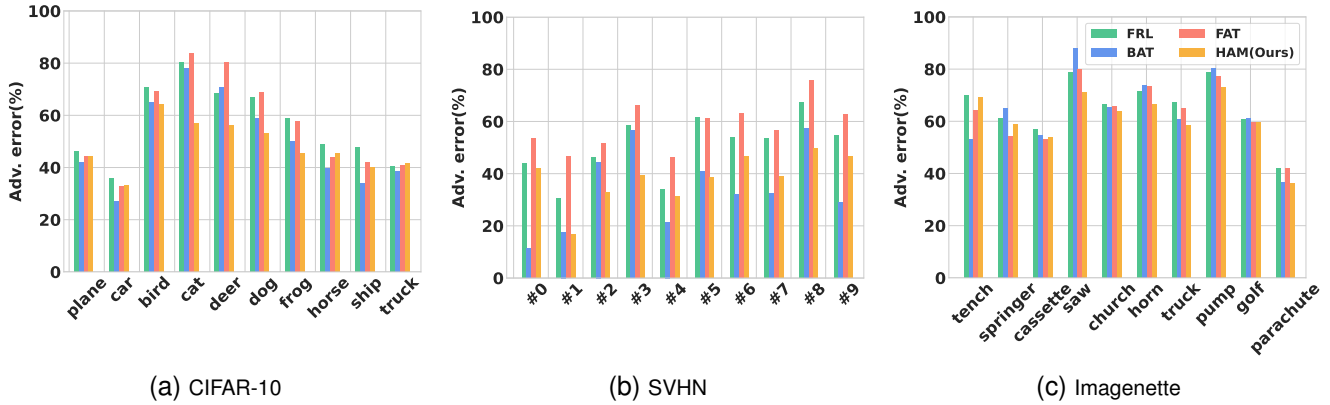
Where N denotes the total number of classes, and Acc.^{i} denotes the accuracy of the i-th class. Specifically, test accuracy on the standard dataset of the i-th class is denoted by Acc.$_{std}^{i}$, while Acc.$_{rob}^{i}$ denotes the test accuracy on the adversarial examples dataset.

For evaluating the clean accuracy and the robust accuracy, we follow the previous work [4] to use the average standard error (Avg.Std.) and the average boundary error (Avg.Bndy.), respectively. Please note that the robust error (Avg.Rob.) is the sum of the Avg.Std. and the Avg.Bndy.

$$Avg.Std. = \frac{1}{N}\sum_{i=1}^{N}(1 - Acc._{std}^{i}), \tag{11}$$

$$Avg.Rob. = \frac{1}{N}\sum_{i=1}^{N}(1 - Acc._{rob}^{i}), \tag{12}$$

$$Avg.Bndy. = Avg.Rob. - Avg.Std., \tag{13}$$



(a) CIFAR-10 (b) SVHN (c) Imagenette

Fig. 8. Fairness: Adversarial error under PGD-20 attack in each class of three datasets.

B. Fairness of HAM

This section reports the experimental results that evaluate the fairness of HAM. We report the same fairness metric as the previous fair AT works [4], [6] for a fair comparison. The results on CIFAR-10, SVHN, and Imagenette are reported in Table II. The results on CIFAR-100 are reported in the *Appendix B*. We can see that the class-wise robust fairness performance of HAM significantly outperforms other state-of-the-art methods on all three datasets, achieving the best or second-to-the best on three different fairness metrics. We analyze the experimental results on each dataset as follows.

CIFAR-10. As shown in Table II, HAM significantly reduces the worst robust error of PGD-AT by 21.3%. Compared to the FRL and BAT methods, the improvements are 15.9%, and 11.7%, respectively. In addition, HAM performs the best or second-to-the best among PGD-AT, FRL, BAT, and FAT on average robust error and maximum class-wise discrepancy.

SVHN. Similar improvements can be observed on the SVHN dataset, as shown in Table II. Compared to PGD-AT, HAM achieves an 18.8% reduction to the worst robust error. HAM also significantly improves BAT by 10.3%. And HAM performs better than FRL (best one) regarding the worst robust error with 17.2%. In terms of maximum class-wise discrepancy, HAM improves FRL, BAT, and CFA methods by a large margin and achieves comparable performance to FAT. These results highlight that HAM improves the class-wise robust fairness while not sacrificing the average robust performance and maximum class-wise discrepancy.

Imagenette. As with the previous two datasets, HAM performs best on the average robustness error and the worst robustness error of all the methods in our evaluation, as shown in Table II. Compared to the standard PGD-AT, HAM reduces the Worst Rob. error by 6.2%. Besides, our HAM does not severely degrade the maximum class-wise discrepancy performance compared to CFA, which is trivial compared to our Avg. Rob. and Worst Rob. reduction.

In addition, we report the robust performance of AT methods on each class in Figure 8, we can see that HAM outperforms other methods on several classes, especially those that get inferior in fairness evaluation when trained with baseline AT methods, e.g. *cat, deer, dog, and frog* (CIFAR-10) and *#1, #2, #3, #7, and #8* (SVHN). The models with HAM may exhibit lower defence scores in certain classes, but note that a fairer model does not necessarily mean it achieves higher scores in every class. Take 8a, for instance, although BAT achieves a lower error rate on "car", its robustness disparity on the best class (car) and worst class (cat) is much larger than that of HAM. The results in 8 show HAM is the fairest method.

TABLE III
COMPARING THE FAIRNESS(%) AND EFFICIENCY(SECONDS) OF HAM WITH BASELINE AT METHODS. THE BEST RESULTS ARE IN **BOLD**. WE REPORT THE IMPROVEMENT MAGNITUDE OF EACH METHOD COMPARED WITH PGD-AT (DENOTED AS AT) IN THE BRACKETS.

Algorithm	Worst Bndy. (↓)	Worst Rob. (↓)	Training time (↓)
AT	57.90	85.50	142
BAT	53.20 (-4.7)	75.90 (-9.6)	498 (+250%)
Fast AT	48.70 (-9.2)	86.30 (+0.8)	**16 (-88%)**
HAM	**35.30 (-22.6)**	**64.20 (-21.3)**	78 (-45%)

TABLE IV
THE ROBUST FAIRNESS(%) AND TRAINING EFFICIENCY(SECONDS) OF HAM WHEN BEING COMBINED WITH OTHER POPULAR AT METHODS, I.E., MART AND AWP. THE BEST RESULTS ARE IN **BOLD**.

Algorithm	Worst Bndy. (↓)	Worst Rob. (↓)	Training time (↓)
AT	57.9	85.5	142
HAM	**35.3 (-22.6)**	**64.2 (-21.3)**	**78 (-45%)**
MART	49.5	75.5	131
MART-HAM	**21.9 (-27.6)**	**51.0 (-24.5)**	**99 (-24%)**
AWP	38.2	75.3	203
AWP-HAM	**21.1 (-17.1)**	**50.6 (-24.7)**	**146 (-28%)**

C. Efficiency and scalability of HAM

The training time of HAM and other AT methods on CIFAR-10 is reported to verify the efficiency advantage of HAM. We also verify the scalability of HAM by combining it with other popular AT methods, MART [11] and AWP. [12]

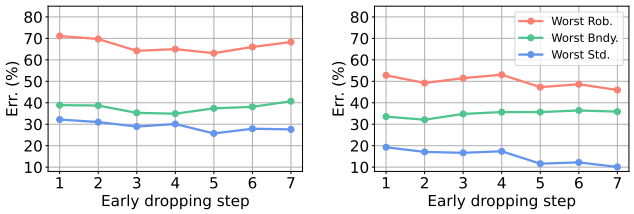
The training time in Table III represents the time (seconds) it takes PreActResNet-18 to train an epoch on CIFAR-10. We can see that accelerated AT (Fast AT) greatly saves training time while worsening the AT fairness. On the contrary, fair AT (BAT) improves fairness while sacrificing training efficiency,

TABLE V
FAIRNESS COMPARISON OF HAM AND TRAINING WITH A RANDOM DROP ON CIFAR-10.

Method	Random drop rate (%) 0(AT)	10	20	30	40	HAM
Worst Std.(%)	33.00	34.20	32.20	33.50	33.40	**28.90**
Worst Bndy.(%)	57.90	56.38	57.30	55.60	55.10	**35.30**
Worst Rob.(%)	85.50	84.80	83.70	84.30	83.50	**64.20**

TABLE VI
FAIRNESS COMPARISON OF HAM AND TRAINING WITH A RANDOM DROP ON SVHN.

Method	Random drop rate (%) 0(AT)	10	20	30	40	HAM
Worst Std.(%)	12.17	12.00	11.62	11.65	12.17	**11.62**
Worst Bndy.(%)	55.06	48.37	54.09	53.97	55.00	**35.66**
Worst Rob.(%)	65.90	65.96	64.81	64.81	65.84	**47.04**



(a) CIFAR-10 (b) SVHN

Fig. 9. Influence of early dropping step on fairness.

TABLE VII
ROBUSTNESS AND FAIRNESS(%) COMPARISON OF HAM AND INVERSE HAM (DISCARDING THE HARD AES) ON CIFAR-10.

Method	Avg. Rob.(↓)	Worst Rob.(↓)
HAM	**48.05**	**64.20**
Inverse HAM	52.94	83.90

making the already inefficient AT slower. HAM improves both results simultaneously.

In addition, HAM is a generic method, which means it can be easily combined with other AT methods to improve fairness and efficiency. Table IV shows the results of HAM being extended to MART and AWP. HAM reduces the Worst Bndy. error and Worst Rob. error of MART by 27.6% and 24.5%, respectively, while also reducing training time by 24%. Besides, AWP combined with HAM outperforms in Worst Bndy. error and Worst Rob. error.

D. Ablation study

Comparison with random dropping. To further confirm the contribution of our HAM method in improving class-wise robust fairness, we compare the HAM with the random dropping AT on CIFAR-10 (Table V) and SVHN (Table VI). Random dropping means that we randomly drop AEs in the training procedure without identifying the easy/hard AEs. We can see that the random dropping groups do not improve the fairness performance compared to the naive PGD-AT method, which highlights the contribution of our HAM method.

Hyper-parameters sensitivity. The early-dropping strategy benefits the time efficiency, while the attack step M of early-dropping directly affects the model performance. Therefore, we explore the relationship between different steps and model fairness. Additionally, the hyper-parameter analysis of the HAM start epoch can be found in the *Appendix E*. It can be seen in Figure 9 that on both two datasets, too few steps can negatively affect the final fairness of the model. As the step increases above 3, the negative impact diminishes substantially. Based on these findings, selecting a step from 3 to 5 can not only satisfy the model fairness but also significantly save training time.

Validating the effectiveness of HAM by discarding the hard AEs. This subsection validates the effectiveness of HAM from the opposite perspective, that is, discarding the hard AEs during the training procedure, instead of easy AEs as done in HAM and observing its impact on robustness and fairness performance. The results, presented in Table VII, show that both the robustness error rate (Avg. Rob.) (48.05% $\rightarrow$ 52.94%) and the unfairness rate (Worst Rob.) (64.20% $\rightarrow$ 83.90%)are increased after switching the discarding strategy to discarding hard AEs. These results validate the rationality and effectiveness of the HAM approach of discarding easy AEs.

V. LIMITATIONS AND FUTURE WORK

Despite the above achievements, several limitations remain that deserve to be solved in future work.

Evaluation of HAM on large-scale datasets like ImageNette. Due to computational constraints, most existing research on fair adversarial training, including our work, has focused on medium-sized datasets like CIFAR-10 and SVHN. While these datasets are widely used, they do not fully capture the complexities and scale of real-world applications. Future research should aim to extend the scope of evaluations to large-scale datasets such as ImageNette, which would provide a more comprehensive understanding of the model's fairness, robustness, and generalizability. Addressing this challenge will require more efficient optimization techniques or access to greater computational resources. Investigating how fairness in adversarial training scales with larger, more diverse datasets will be crucial to advancing this field.

Fair adversarial training beyond image classification. Currently, most research on fair adversarial training is centered on image classification tasks. However, fairness in adversarial robustness could extend beyond classification into other domains, such as object detection and semantic segmentation, which present different challenges due to their inherent complexity and diversity. Exploring whether these tasks face similar robust fairness issues, and developing methods to address them, would significantly broaden the impact of fair adversarial training.

VI. CONCLUSION

This paper focuses on improving the robust fairness and efficiency of AT while maintaining satisfactory overall robustness. We first reveal the limitations of class-wise reweighting methods used in recent fair adversarial training techniques, demonstrating that sample-level reweighting can be a more

effective and efficient method in enhancing the robust fairness of the model. This observation motivates us to propose Hard Adversarial Example Mining (HAM), which mitigates the unfairness issues by dropping easy AEs and reweighting the rest hard AEs during the adversarial training process. Despite HAM does not explicitly regularize the fairness of the model, extensive evaluation of HAM on multiple datasets verifies the superiority of HAM compared to state-of-the-art fair adversarial training methods in efficiently training a robust model while ensuring fairness.

APPENDIX

A. Detailed descriptions of other robust fairness literature

The robust fairness issue is first found in the classification task, which makes the machine learning models exhibit different clean accuracy and adversarial accuracy among different classes [46]. Later on, Nanda et al. [47] studied the robust fairness issue on the discriminative features of the input sample group. They found machine learning models exhibit a different level of robustness among different groups of individuals, which may lead to discriminative decisions in terms of age, skin color, gender, etc. We will briefly describe the design principle of each fair adversarial training method below. Please note that all of these methods regularize the fairness of the model from the perspective of class-wise disparity, which is different from the sample-level regularization scheme of our HAM.

Fair robust learning (FRL) [4]. FRL addresses the robust fairness issue of adversarial training, which studies both the clean accuracy fairness and the adversarial accuracy fairness of the model. FRL defines and regularizes the robust fairness of the model with a class-wise regularization objective function, which is formalized as:

$$\begin{aligned}\min_{\theta,\phi} \mathcal{L}(f_\theta,\phi) &= \mathcal{R}_{nat}(f_\theta)+\mathcal{R}_{bndy}(f_\theta)\\ &+\sum_{i=1}^{Y}\phi^i_{nat}(\mathcal{R}_{nat}(f_\theta,i)-\mathcal{R}_{nat}(f_\theta)-\tau_1)\\ &+\sum_{i=1}^{Y}\phi^i_{bndy}(\mathcal{R}_{bndy}(f_\theta,i)-\mathcal{R}_{bndy}(f_\theta)-\tau_1),\end{aligned} \tag{14}$$

where the $\mathcal{R}_{nat}$ and the $\mathcal{R}_{bndy}$ denote the clean error rate and the boundary error rate, respectively (Please refer to [4] for a detailed definition of them). The ϕ denotes the Lagrangian multiplier, the Y denotes the total number of classes, and the τ denotes the pre-defined threshold. FRL has two different regularization variants, FRL-reweight, and FRL-remargin. The FRL-reweighing readjusts the weights of each adversarial example used in the original cross-entropy loss function. The FRL-remargin adaptively adjusts the size of the perturbation during the adversarial training. Our HAM applies a sample-level regularization scheme, which is different from the class-wise regularization of FRL.

Fairly adversarial training (FAT) [7]. Based on the theoretical framework of FRL, FAT expands it to a more general setting. Specifically, FAT investigates both clean accuracy fairness and adversarial accuracy fairness from the perspective of robust radii, which measures the fairness issue with the variance of class-wise adversarial risk (VCAR):

$$VCAR(f)=\frac{1}{L}\sum_{i=1}^{L}(R_{adv}(f,i)-\hat{R}_{adv}(f))^2, \tag{15}$$

where $R_{adv}(f,i)$ denotes the adversarial accuracy of the i-th class, and the $\hat{R}_{adv}$ denotes the overall average adversarial accuracy.

Fair and robust classification (FRoC) [22]. FRoC is designed to achieve both fair and robust models, but it only addresses the clean accuracy disparity and does not discuss the robustness disparity. FRoC defines the fairness disparity as the differences between the accuracy across different groups of examples, which is the same as our HAM. FRoC has two variants, FRoC-In and FRoC-PRE, designed for different scenarios. FRoC-In regularizes fairness during the model training stage with a class-wise regularization term, which is formulated as (take a binary classification task for example):

$$\mathbf{F}(A,X)=|\frac{\sum_{i=1}^{n}(1-a_i)\mathbf{M}_c(x_i)}{\sum_{i=1}^{n}(1-a_i)}-\frac{\sum_{i=1}^{n}a_i\mathbf{M}_c(x_i)}{\sum_{i=1}^{n}a_i}|, \tag{16}$$

where $a_i = 0,1$ is the label, $\mathbf{M}_c$ is the measurement of the classification tasks, such as the accuracy or the F1-score, and x_i denotes the input examples. FRoC-PRE enhances the fairness of the model by preprocessing the training dataset. FRoC-PRE consists of two types of data modification: label flipping and adversarial data augmentation. The class-wise disparity is also directly applied in the decision process.

Balance adversarial training (BAT) [6]. BAT observes that two unfair phenomena may be the cause of the fairness issue of adversarial training: (1) different difficulties in generating adversarial examples from each class (source-class fairness) and disparate target class tendencies when generating adversarial examples (target class fairness). Based on the TRADES [14] framework, BAT uses the following objective function to improve the source-class fairness:

$$\begin{aligned}\mathcal{L}_{source-class}=\min_{\theta}\sum_{i=1}^{n}CE(f_\theta(x_{clean,i},y_i)\\ +\beta\max KL(f_\theta(x_i),f_\theta(x_{adv,i}))),\end{aligned}$$

where CE denotes the cross-entropy loss, KL is the Kullback–Leibler (KL) divergence, and β is a balancing parameter. BAT improves the target-class fairness with the following regularization term:

$$C_D(\theta)=\frac{1}{n}\sum_{i=1}^{n}(\delta_i-\hat{\delta}_i d_\theta(x_i)), \tag{17}$$

where δ_i is the adversarial perturbation of x_i, $\hat{\delta}_i$ is the average value of the adversarial perturbation, and $d_\theta(x_i)$ indicates the distance of x_i to the decision boundary of f. Please note that BAT is still a class-wise regularization scheme and our HAM fundamentally differs from BAT.

Class-wise calibrated fair adversarial training (CFA) [44]. CFA aims to enhance the fairness of the model without sacrificing its overall robustness from the perspective of the hyper-parameter configurations of adversarial training, including the

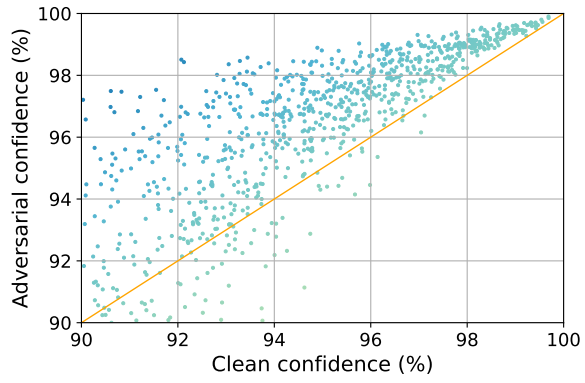


Fig. 10. Class-wise over-confident AE proportion (blue line) and robustness (red line) of AWP on CIFAR-10 dataset. Classes with higher over-confidence AE proportion have higher robustness in most cases.

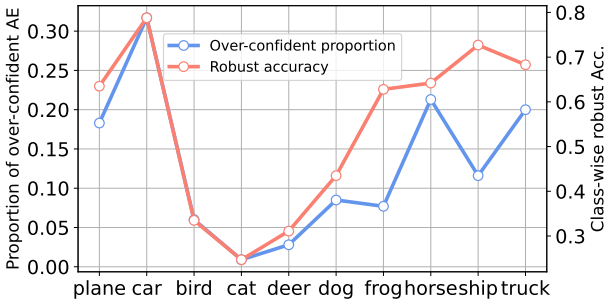


Fig. 11. For an AT model, classes with a higher proportion of easy AEs have higher robustness in most cases.

perturbation margin, regularization, and weight averaging. CFA can automatically optimize the configurations during adversarial training to realize better fairness.

Fair adversarial robustness distillation (Fair-ARD) [48]. Fair-ARD studies the robust fairness under the transfer-learning setting and finds that student models only partially inherit the robust fairness of the teacher model. Fair-ADR addresses this challenge by reweighting the training examples of different classes in terms of the class-wise difficulties, which is formulated as:

$$\min_{\theta_S} \frac{1}{C}\sum_{i=1}^{C}\frac{1}{n_i}\sum_{j=1}^{n_i}(\omega_i \mathcal{L}_{ARD}(S,T,x_i^j,y_i,\tau,\alpha)), \tag{18}$$

where ω_i is the reweighting parameter determined by the difficulty of each class; S and T denote the student and teacher models, respectively; C is the number of classes, $\mathcal{L}_{ARD}$ is the distillation loss function; τ and α are hyper-parameters.

B. Experimental results on CIFAR-100

The fairness results for CIFAR-100 are provided in the appendix. It's worth noting that the performance of HAM on CIFAR-100 may be influenced by the constrained amount of training data in this dataset (500 images in each class). Given HAM's characteristic of discarding easy adversarial examples, the relatively small number of training examples per class could potentially impact its performance. We acknowledge this limitation on the CIFAR-100 dataset and add some weights to the clean examples in the experiment. We plan to address it in our future work. The experimental results are shown in Table VIII. It can be seen that HAM exhibits superior performance compared to other state-of-the-art methods in terms of Avg. Rob. and Worst Rob. Although HAM achieves the highest value on the maximum class-wise discrepancy, since Worst Rob. across methods were approximate, this demonstrates that HAM possesses superior robustness compared to other methods.

TABLE VIII
ERROR RATE(%) OF AVERAGE & WORST-CLASS STANDARD, BOUNDARY AND ROBUST FOR OUR HAM AND OTHER METHODS ON CIFAR-100. THE BEST RESULTS ARE IN **BOLD**, AND THE SECOND BEST RESULTS ARE MARKED WITH UNDERLINES.

Method	Avg. Rob. (↓)	Worst Rob. (↓)	Maximum class-wise discrepancy (↓)
PGD-AT	79.40 (±0.88)	100.00 (±0.00)	73.00 (±0.54)
FRL	83.00 (±0.32)	100.00 (±0.43)	73.00 (±0.46)
BAT	77.10 (±0.42)	100.00 (±0.73)	71.00 (±0.32)
FAT	79.50 (±0.43)	**99.00** (±0.55)	**65.00** (±0.58)
CFA	80.10 (±0.24)	100.00 (±0.31)	67.00 (±0.52)
HAM(Ours)	**77.00** (±0.42)	**99.00** (±0.51)	75.00 (±0.42)

C. Adversarial over-confidence phenomenon on popular AT methods

To further validate the existence of the adversarial over-confidence phenomenon, we conduct an additional experiment using an advanced AT-based method, AWP. As shown in 10 and 11, the over-confidence phenomenon is also evident in other state-of-the-art methods. Notably, Fig.10 illustrates that such over-confidence examples represent a larger proportion compared to standard PGD-AT.

D. MART-HAM and AWP-HAM Algorithms

As mentioned in Section IV-C in the main paper, HAM is a general method that can be combined with other AT-based methods. This section describes how to integrate HAM into MART [11] and AWP [12] and summarizes the process in Algorithm 2 and Algorithm 3.

MART-HAM is a modified version of MART with the proposed HAM. It can be easily realized using the MART loss function:

$$\begin{aligned} loss_i^{MART}(x_i,y_i,f_\theta) &= BCE(\mathbf{p}(\tilde{x}_i,f_\theta),y_i) \\ &+ \lambda \cdot KL((x_i,f_\theta) \parallel \mathbf{p}(\tilde{x}_i,f_\theta)) \cdot (1-\mathbf{p}_{y_i}(x_i,f_\theta)) \end{aligned} \tag{19}$$

and it is described in detail in Algorithm 2.

AWP can also be modified to a version with the proposed HAM. It can be summarized in Algorithm 3.

E. Analysis of Starting epoch selection

Executing HAM from the very beginning will lead to the fluctuation of the model training and the problem of difficult convergence. Hence, determining the optimal activation timing for the HAM method becomes crucial and needs investigation. In this section, we fix M and analyze the selection of the starting epoch. Figure 12 shows that there exists a suitable range for the starting epoch. In a total of 120 epochs of training, with the learning rate being reduced to 10% of the

Algorithm 2 Pseudo-code of MART-HAM

Input: Network f_θ, data $S = \{(x_i, y_i)\}_{i=1}^{n}$, batch size n_{bs}, number of batches n_b, learning rate η, training epochs T, whole attack step K, early-dropping step M
Output: Robust model f_θ

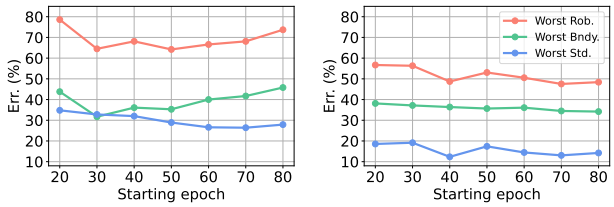
for epoch = $1, \cdots, T$ **do**
  Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{n_{bs}}$ from S
  for mini-batch = $1, \cdots, n_b$ **do**
    Generate $\{\tilde{x}_i^M\}_{i=1}^{n_{bs}}$ with M-step PGD
    Construct hard AE set: $H_M = \{\tilde{x}_j^M : f_\theta(\tilde{x}_j^M) \neq y_j\}$
    Finish hard AE generation with $(K-M)$ step PGD: $H_K = \{\tilde{x}_j^K : \tilde{x}_j^M \in H_M\}$
    Update $hard(x_i, y_i, f_\theta), i = 1, \cdots, n_{bs}$ by Equation 7
    Update θ with SGD by Equation 19
  end for
end for

Algorithm 3 Pseudo-code of AWP-HAM

Input: Network f_θ, data $S = \{(x_i, y_i)\}_{i=1}^{n}$, batch size n_{bs}, number of batches n_b, learning rate η, training epochs T, whole attack step K, early-dropping step M, AWP steps A, AWP step size η_{AWP}
Output: Robust model f_θ

for epoch = $1, \cdots, T$ **do**
  Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{n_{bs}}$ from S
  for mini-batch = $1, \cdots, n_b$ **do**
    Generate $\{\tilde{x}_i^M\}_{i=1}^{n_{bs}}$ with M-step PGD
    Construct hard AE set: $H_M = \{\tilde{x}_j^M : f_\theta(\tilde{x}_j^M) \neq y_j\}$
    Finish hard AE generation with $(K-M)$ step PGD: $H_K = \{\tilde{x}_j^K : \tilde{x}_j^M \in H_M\}$
    Update $hard(x_i, y_i, f_\theta), i = 1, \cdots, n_{bs}$ by Equation 7
    for a = $1, \cdots, A$ **do**
    $v = Clip(v + \eta_{AWP} \frac{\nabla_v \frac{1}{|H|} \sum_i loss(f_{\theta+v}(\tilde{x}_i), y_i)}{\left\| \nabla_v \frac{1}{|H|} \sum_i loss(f_{\theta+v}(\tilde{x}_i), y_i) \right\|} \|\theta\|)$
    end for
    Update θ with SGD by Equation 6
  end for
end for

maximum rate at the $60th$ epoch and the starting epoch set between 40 and 60, superior performance is observed. Based on these observations, we infer that initiating the HAM process slightly prior to the initial reduction in learning rate leads to optimal outcomes.



(a) CIFAR-10 (b) SVHN

Fig. 12. Influence of begin epoch on fairness.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2015.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *ArXiv*, vol. abs/1706.06083, 2018.

[4] H. Xu, X. Liu, Y. Li, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *ICML*, 2021.

[5] W. Wang, H. Xu, X. Liu, Y. Li, B. M. Thuraisingham, and J. Tang, "Imbalanced adversarial training with reweighting," *ArXiv*, vol. abs/2107.13639, 2021.

[6] C. Sun, C. Xu, C. Yao, S. Liang, Y. Wu, D. Liang, X. Liu, and A. Liu, "Improving robust fairness via balance adversarial training," *arXiv preprint arXiv:2209.07534*, 2022.

[7] X. Ma, Z. Wang, and W. Liu, "On the tradeoff between robustness and fairness," in *Advances in Neural Information Processing Systems*, 2022.

[8] A. Shrivastava, A. K. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, 2016.

[9] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," *ArXiv*, vol. abs/1812.05159, 2019.

[10] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware instance-reweighted adversarial training," *ArXiv*, vol. abs/2010.01736, 2021.

[11] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *ICLR*, 2020.

[12] D. Wu, S. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," *arXiv: Learning*, 2020.

[13] L. Rice, E. Wong, and J. Z. Kolter, "Overfitting in adversarially robust deep learning," in *ICML*, 2020.

[14] H. R. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," *ArXiv*, vol. abs/1901.08573, 2019.

[15] M. Kim, J. Tack, J. Shin, and S. J. Hwang, "Entropy weighted adversarial training," in *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

[16] R. Rade and S.-M. Moosavi-Dezfooli, "Reducing excessive margin to achieve a better accuracy vs. robustness trade-off," in *International Conference on Learning Representations*, 2022.

[17] X. Mao, Y. Chen, R. Duan, Y. Zhu, G. Qi, X. Li, R. Zhang, H. Xue *et al.*, "Enhance the visual representation via discrete adversarial training," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7520–7533, 2022.

[18] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan, "Better diffusion models further improve adversarial training," in *International Conference on Machine Learning*. PMLR, 2023, pp. 36 246–36 263.

[19] G. Jin, X. Yi, D. Wu, R. Mu, and X. Huang, "Randomized adversarial training via taylor expansion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 447–16 457.

[20] J. Dong, S.-M. Moosavi-Dezfooli, J. Lai, and X. Xie, "The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 678–24 687.

[21] Q. Li, Y. Hu, Y. Dong, D. Zhang, and Y. Chen, "Focus on hiders: Exploring hidden threats for enhancing adversarial training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24 442–24 451.

[22] H. Sun, K. Wu, T. Wang, and W. H. Wang, "Towards fair and robust classification," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 356–376.

[23] B. Li and W. Liu, "Wat: improve the worst-class robustness in adversarial training," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 12, 2023, pp. 14 982–14 990.

[24] Y. Zhang, T. Zhang, R. Mu, X. Huang, and W. Ruan, "Towards fairness-aware adversarial learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24 746–24 755.
[25] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," in *NeurIPS*, 2019.
[26] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash, "Efficient adversarial training with transferable adversarial examples," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1178–1187, 2020.
[27] N. Ye, Q. Li, X.-Y. Zhou, and Z. Zhu, "Amata: An annealing mechanism for adversarial training acceleration," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 691–10 699.
[28] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. P. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *NeurIPS*, 2019.
[29] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *ArXiv*, vol. abs/2001.03994, 2020.
[30] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash, "Efficient adversarial training with transferable adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1181–1190.
[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
[33] Z. Cui, J. Zhou, X. Wang, M. Zhu, and Y. Peng, "Learning continual compatible representation for re-indexing free lifelong person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 614–16 623.
[34] X. Xia, B. Han, Y. Zhan, J. Yu, M. Gong, C. Gong, and T. Liu, "Combating noisy labels with sample selection by mining high-discrepancy examples," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 1833–1843.
[35] F. Waseda, S. Nishikawa, T.-N. Le, H. H. Nguyen, and I. Echizen, "Closer look at the transferability of adversarial examples: How they fool different models differently," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1360–1368.
[36] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 2020.
[37] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. J. Belongie, "Class-balanced loss based on effective number of samples," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9260–9269, 2019.
[38] J. Pan, L. G. Foo, Q. Zheng, Z. Fan, H. Rahmani, Q. Ke, and J. Liu, "Gradmdm: Adversarial attack on dynamic networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 11 374–11 381, 2023.
[39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
[41] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, "Bag of tricks for adversarial training," *ArXiv*, vol. abs/2010.00467, 2021.
[42] T. Pang, X. Yang, Y. Dong, K. Xu, H. Su, and J. Zhu, "Boosting adversarial training with hypersphere embedding," *ArXiv*, vol. abs/2002.08619, 2020.
[43] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," in *IJCAI*, 2021.
[44] Z. Wei, Y. Wang, Y. Guo, and Y. Wang, "Cfa: Class-wise calibrated fair adversarial training," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8193–8201, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257767427
[45] Q. Tian, K. Kuang, K. Jiang, F. Wu, and Y. Wang, "Analysis and applications of class-wise robustness in adversarial training," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235254105
[46] P. Benz, C. Zhang, A. Karjauv, and I. S. Kweon, "Robustness may be at odds with fairness: An empirical study on class-wise accuracy," in *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*. PMLR, 2021, pp. 325–342.
[47] V. Nanda, S. Dooley, S. Singla, S. Feizi, and J. P. Dickerson, "Fairness through robustness: Investigating robustness disparity in deep learning," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 466–477.
[48] X. Yue, N. Mou, Q. Wang, and L. Zhao, "Revisiting adversarial robustness distillation from the perspective of robust fairness," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Chenhao Lin (Member, IEEE) received the B.Eng. degree in automation from Xi'an Jiaotong University in 2011, the M.Sc. degree in electrical engineering from Columbia University, in 2013, and the Ph.D. degree from The Hong Kong Polytechnic University, in 2018. He is currently a Professor at the Xi'an Jiaotong University of China. His research interests are in artificial intelligence security, adversarial machine learning, and intelligent identity security.

Xiang Ji received the M.E. degree at the School of Software Engineering, Xi'an Jiaotong University, Xi'an, China in 2024. His research interests include adversarial examples and Al system security.

Yulong Yang received the B.Eng. degree in Computer Science and Engineering from Xi'an Jiaotong University in 2022, where he is currently pursuing the Ph.D. degree in Cyberspace Security with the School of Cyber Science and Engineering. His current research interests include adversarial machine learning, model compression, and AI agent security.

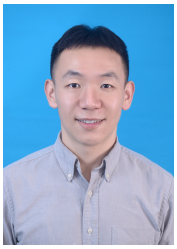
Qian Li (Member, IEEE) received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, China, in 2021. He is currently an Assistant Professor with the School of Cyber Science and Engineering, Xi'an Jiaotong University. His research interests include adversarial deep learning, artificial intelligence security, and optimization of theory.

Zhengyu Zhao (Member, IEEE) received the Ph.D. degree from Radboud University, The Netherlands. He is currently an Associate Professor at Xi'an Jiaotong University, China. His general research interests include machine learning security and privacy. Most of his work has concentrated on security (e.g., adversarial examples and data poisoning) and privacy (e.g., membership inference) attacks against deep learning-based computer vision systems.

Zhe Peng (Member, IEEE) Zhe Peng is currently a research assistant professor in the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University. He received the B.S. degree from Northwestern Polytechnical University, the M.S. degree from the University of Science and Technology of China, and the Ph.D. degree from Hong Kong Polytechnic University. He was a visiting scholar in the Department of Electrical and Computer Engineering, Stony Brook University. His research interests include blockchain, web3, artificial intelligence of things, data security and privacy.

Run Wang (Member, IEEE) is currently an associate professor at the School of Cyber Science and Engineering, Wuhan University, China. Before that, he worked as a Postdoc Research Fellow at Nanyang Technological University, Singapore from 2019 to 2021. He received his Ph.D. degree in information security from Wuhan University, China, in 2018. His research interests are at the intersection of security, privacy, and AI.

Liming Fang (Member, IEEE) received the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2012, where he is currently a Professor with the College of Computer Science and Technology. He was a Postdoctoral Researcher of Information Security with the City University of Hong Kong. He was a principle investigator and the main participant in more than 20 key projects, including the National Natural Science Foundation of China and the 863/973 plan. He has authored or co-authored more than 80 papers in journals and international conferences, including Theoretical Computer Science, Designs Codes and Cryptography, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. His current research interests include information security, applied cryptography, AI security, blockchain, and proxy re-encryption.



Chao Shen (Senior Member, IEEE) received the B.S. degree in Automation from Xi'an Jiaotong University, China in 2007; and the Ph.D. degree in Control Theory and Control Engineering from Xi'an Jiaotong University, China in 2014. He is currently a Professor in the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University of China. His current research interests include AI Security, insider/intrusion detection, behavioral biometrics, and measurement and experimental methodology.