This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

Fast Adversarial Training with Adaptive Step Size

Zhichao Huang*, Yanbo Fan†, Chen Liu, Weizhong Zhang, Yong Zhang,
Mathieu Salzmann, Sabine Süsstrunk, Jue Wang

***Abstract*—While adversarial training and its variants have shown to be the most effective algorithms to defend against adversarial attacks, their extremely slow training process makes it hard to scale to large datasets like ImageNet. The key idea of recent works to accelerate adversarial training is to substitute multi-step attacks (e.g., PGD) with single-step attacks (e.g., FGSM). However, these single-step methods suffer from catastrophic overfitting, where the accuracy against PGD attack suddenly drops to nearly 0% during training, and the network totally loses its robustness. In this work, we study the phenomenon from the perspective of training instances. We show that catastrophic overfitting is instance-dependent, and fitting instances with larger input gradient norm is more likely to cause catastrophic overfitting. Based on our findings, we propose a simple but effective method, *Adversarial Training with Adaptive Step size (ATAS)*. ATAS learns an instance-wise adaptive step size that is inversely proportional to its gradient norm. Our theoretical analysis shows that ATAS converges faster than the commonly adopted non-adaptive counterparts. Empirically, ATAS consistently mitigates catastrophic overfitting and achieves higher robust accuracy on CIFAR10, CIFAR100, and ImageNet when evaluated on various adversarial budgets.**

***Index Terms*—Adversarial Examples, Fast Adversarial Training**

I. Introduction

ADVERSARIAL examples [1], [2] cause serious safety concerns in deploying deep learning models. In order to defend against adversarial attacks, many approaches have been proposed [3]–[9]. Among them, adversarial training and its variants [7], [8], [10] have been recognized as the most effective defense mechanism. Adversarial training (AT) is generally formulated as a minimax problem

$$\min_{\boldsymbol{\theta}} \max_{\mathbf{x}_i^* \in \mathcal{B}_p(\mathbf{x}_i,\varepsilon)} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{x}_i^*, y_i; \boldsymbol{\theta}) \ , \tag{1}$$

where $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is the training set and $\ell(\mathbf{x}, y; \boldsymbol{\theta})$ is the loss function parametrized by $\boldsymbol{\theta}$. $\mathcal{B}_p(\mathbf{x}_i, \varepsilon)$ represents a L_p norm ball centered at $\mathbf{x}_i$ with radius ε. AT in Equation (1) boosts the adversarial robustness by adopting adversarial examples generated in the inner maximization. Despite the effectiveness of AT, solving the inner maximization requires multiple steps of projected gradient descent (PGD) [7], [11]. Therefore, AT is much slower than vanilla training (*e.g.*, 10 times longer training time for AT in [11]), making it challenging to scale AT to large datasets such as ImageNet.

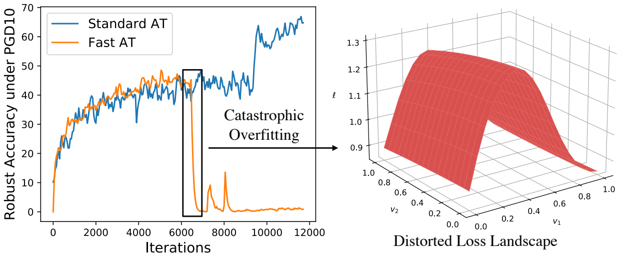


Fig. 1: The left plot shows the robust accuracy of Standard AT and Fast AT on CIFAR10 w.r.t. the number of training iterations. The Fast AT suffers from catastrophic overfitting, where the robust accuracy against PGD suddenly decreases to nearly 0%. The right plot visualizes the loss landscape when catastrophic overfitting happens.

Currently, the typical solution to accelerate AT is to substitute multi-step attacks (e.g., PGD) with single-step attacks (e.g., FGSM). Several works have been proposed following this direction, including FGSM-RS [12], ATTA [13], *etc*. These methods achieve the best robust accuracy for fast AT. However, recent works [14], [15] demonstrate that the single-step method suffers from catastrophic overfitting, where the model's robustness against PGD attack suddenly drops to nearly 0% while the robust accuracy against FGSM attack rapidly increases [12], as shown in Figure 1. This will completely destroy the robustness of the networks. It is worth noting that catastrophic overfitting is different from robust overfitting mentioned in [11]. The latter one refers to the generalization gap between training and test data while catastrophic overfitting means the overfitting to a specific type of attack that is irrelevant to the training and test set. Some works have been proposed to understand and alleviate the catastrophic overfitting [14], [15]. However, their solutions significantly increase the training time. For example, the gradient align regularizer

$$\mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{U}([-\varepsilon,\varepsilon]^d)} \left[1 - \cos\left(\nabla_{\mathbf{x}} \ell(\mathbf{x}, y; \boldsymbol{\theta}), \nabla_{\mathbf{x}} \ell(\mathbf{x} + \boldsymbol{\delta}, y; \boldsymbol{\theta})\right)\right]$$

in [14] requires calculating the second order gradient and it is still 5 times slower than vanilla training, compared with 2 times in [12]. In addition, [15] needs to check several points within the ℓ_p norm ball, which needs several forward propagations and is still about 4 times slower than vanilla training. Therefore, existing methods are still unsatisfactory in terms of both training efficiency and robust performance.

In this work, we analyze catastrophic overfitting from the perspective of training instances. By taking the gradient norm

Zhichao Huang is with the Department of Mathematics, Hong Kong University of Science of Technology. Yanbo Fan, Yong Zhang, and Jue Wang are with Tencent AI Lab. Chen Liu is with Department of Computer Science, City University of Hong Kong. Weizhong Zhang is with School of Data Science, Fudan University. Mathieu Salzmann and Sabine Süsstrunk are with École polytechnique fédérale de Lausanne.

*Part of this work is done during the internship of Zhichao Huang at Tencent AI Lab.

†Yanbo Fan (fanyanbo0124@gmail.com) is the corresponding author.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

$\|\nabla_{\mathbf{x}}\ell(\mathbf{x}, y; \boldsymbol{\theta})\|$ as an indicator, we find that different training instances have different probabilities of causing catastrophic overfitting. Instances with large gradient norms are more sensitive to adversarial noise and their loss landscape is less smooth. Thus, fitting them with FGSM is more likely to distort the loss landscape, resulting in catastrophic overfitting.

Furthermore, catastrophic overfitting is closely related to the optimization process of the inner maximization, *e.g.*, the setting of step size. When catastrophic overfitting does not occur, the larger step size leads to a stronger attack and thus strengthens the robustness of the network [12]. On the other side, a larger step size is more likely to cause catastrophic overfitting in the training process [12], [14]. Based on these findings, we propose *Adversarial Training with Adaptive Step size (ATAS)*, a simple but effective fast AT method that uses the previous initialization in ATTA [13] and takes the step size of the inner maximization inversely proportional to the input gradient norm. Instances with large gradient norms are given a small step size to prevent catastrophic overfitting. By contrast, instances with small gradient norms will have large step sizes to improve the strength of the attack.

We theoretically analyze the convergence of ATAS and prove that it converges faster than the non-adaptive counterpart, which is commonly adopted in existing works [13], especially when the distribution of the input gradient norm is long-tailed. Empirically, We evaluate ATAS on CIFAR10, CIFAR100 [16], and ImageNet [17] with different network architectures and adversarial budgets, showing that ATAS mitigates catastrophic overfitting and achieves higher robust accuracy under various attacks including PGD10, PGD50 [7] and AutoAttack [18].

Our contributions are summarized as follows: 1) To the best of our knowledge, we are the first to analyze catastrophic overfitting from the perspective of training instances, and demonstrate that instances with large input gradient norms are more likely to cause catastrophic overfitting. 2) We propose a new algorithm, ATAS, which takes the step size of the inner maximization to be inversely proportional to the input gradient norm in order to prevents catastrophic overfitting and maintain the strength of the attack. 3) Theoretically, we prove that ATAS converges faster than its non-adaptive counterpart. 4) Empirically, we conduct extensive experiments to evaluate ATAS on different datasets, network architectures, and adversarial budgets, showing that ATAS consistently improves the robust accuracy and mitigates catastrophic overfitting.

II. Background and Related Work

A. Adversarial Examples.

Adversarial examples are first discussed in [1], where a small perturbation of the input significantly changes the prediction. Adversarial examples can be generated using the gradient of the input $\mathbf{x}$. Fast Gradient Signed Method (FGSM) [19] approximates the loss function $\ell(\mathbf{x}, y; \boldsymbol{\theta})$ with the first order Taylor expansion so that adversarial examples can be generated with one step of projected gradient $\mathbf{x}^{\text{FGSM}} = \mathbf{x} + \varepsilon \cdot \text{sgn}(\nabla_{\mathbf{x}}\ell(\mathbf{x}, y; \boldsymbol{\theta})))$, where sgn is the sign function, ε is the adversarial budget. Projected Gradient Descent (PGD) [7] extends FGSM to multiple steps to strengthen the attack. With a step size α, the adversarial example at the t-th step is $\mathbf{x}^{t+1} = \Pi_{\mathcal{B}_p(\mathbf{x},\varepsilon)}[\mathbf{x}^t + \alpha \cdot \text{sgn}(\nabla_{\mathbf{x}^t}\ell(\mathbf{x}^t, y; \boldsymbol{\theta}))]$, where $\Pi_{\mathcal{B}_p(\mathbf{x},\varepsilon)}$ means the projection onto $\mathcal{B}_p(\mathbf{x}, \varepsilon)$. Several stronger attacks are proposed to reliably evaluate the models' robustness [18], [20], [21]. Among them, Autoattack [18] stands out as the strongest attack.

While many methods [3], [4], [7], [8], [10], [22] have been proposed to defend adversarial attacks, adversarial training (AT) and its variants [7], [8], [10], [23], [24] are shown to be the most effective methods to train a truly robust network. AT can be formulated as a minimax problem in Equation (1). Finding solutions for minimax optimization has been a major endeavor in mathematics and computer science [25], [26]. Theoretically, the well-known Stochastic Gradient Descent Ascent (SGDA) finds an ε-approximate stationary point in $\mathcal{O}(1/\varepsilon^2)$ iterations with averaging for convex-concave games [27]. However, it is not appropriate to formulate the optimization of AT as SGDA or SGDmax [28], since AT only updates a part of the coordinates in $\mathbf{x} = [x_1, x_2, \cdots x_n]$ for the maximization. The inner maximization actually corresponds to the stochastic block coordinate ascent. Empirically, the neural network is non-concave with respect to the input, so perfectly solving the inner maximization is NP-hard [29]. It is usually approximated by a strong attack like PGD [7], which needs multiple steps of the calculation the gradients. Therefore, AT is much slower than vanilla training.

B. Fast Adversarial Training.

There have been a series of work on accelerating AT [12], [14], [15], [24], [30]–[33]. FreeAT [30] first proposes a fast AT method by simultaneously optimizing the model's parameter and the adversarial perturbations by batch replaying. YOPO [31] adopts a similar strategy to optimize the adversarial loss function. Later on, single-step methods are shown to be more effective than FreeAT and YOPO [12]. FGSM with Random Start (FGSM-RS) [12] can be used to generate adversarial perturbations in one step to train a robust network if the hyperparameters are carefully tuned. ATTA [13] utilizes the transferability of adversarial examples between epochs, using adversarial example of the previous epoch as the initialization, optimizing the model parameters with

$$\begin{aligned}
&\mathbf{x}_i^j = \Pi_{\mathcal{B}_p(\mathbf{x}_i,\varepsilon)}[\mathbf{x}_i^{j-1} + \alpha \cdot \text{sgn}(\nabla_{\mathbf{x}_i^{j-1}}\ell(\mathbf{x}_i^{j-1}, y; \boldsymbol{\theta}))] \\
&\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\ell(\mathbf{x}_i^j, y; \boldsymbol{\theta})) \ ,
\end{aligned} \tag{2}$$

where $\mathbf{x}_i^j$ means the adversarial examples generated for the i-th instance $\mathbf{x}_i$ at the j-th epoch. ATTA shows comparable robust accuracy with FGSM-RS. SLAT [34] perturbs both inputs and the latents simultaneously with FGSM, ensuring more reliable performance.

As mentioned above, these single-step methods suffer from *catastrophic overfitting*, meaning the robustness against PGD attack suddenly drops to nearly 0% while the robust accuracy against FGSM attack rapidly increases. In order to prevent catastrophic overfitting, FGSM-GA [14] adds a regularizer that aligns the direction of the input gradient. SSAT [15] studies the phenomenon from of the perspective of loss landscape, finding that catastrophic overfitting is a result of highly distorted loss

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

surface. It proposes a new algorithm to resolve catastrophic overfitting by checking the loss value along the direction of the gradient. However, both algorithms require much more computation than FGSM-RS [12] and ATTA [13]. Compared with these works, we study catastrophic overfitting from the perspective of training instances and show that using adaptive step sizes in single-step methods prevents catastrophic overfitting. Our method achieves better performance with negligible computational overhead.

Appropriate initialization of fast adversarial training (AT) can effectively mitigate catastrophic overfitting, as demonstrated in the studies of FGSM-SDI [33] and FGSM-PGI [35]. These methods complement the adaptive step size studied in our paper and can be synergistically combined with ATAS to further enhance the performance of fast AT. In addition, NuAT [36] adds nuclear norm in the fast AT. SubAT [32] studies the relationship between parameter gradient norm and catastrophic overfitting.

Adaptive step sizes have been widely used in training neural networks such as AdaGrad [37], RMSProp [38] and ADAM [39]–[41]. However, our motivation is different, and to the best of our knowledge, we are the first to introduce the adaptive step size in fast AT.

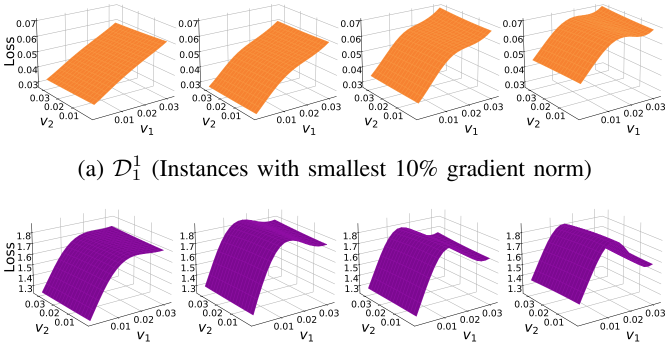
Several related studies explore adaptive adversarial budget techniques [23], [42]–[44] aimed at enhancing the robustness and trade-off capabilities of AT. However, these papers targets at enhancing the robustness of standard adversarial training instead of acclerating adversarial training algorithms. Besides, some of these methods [43] may employ complex algorithms to determine the adversarial boundary, making it challenging for them to be efficiently applied in fast AT scenarios.

III. Motivation

Catastrophic overfitting is interpreted as a result of highly distorted loss landscapes of the input [15]. For example, FGSM-RS [12] uses large step sizes in the inner maximization to generate adversarial examples. It may only minimize the classification loss near the boundary of the adversarial budget, while the loss inside the adversarial budget may increase, leading to a highly distorted loss landscape. Figure 1 gives an illustration.

Recalling that different inputs have different loss landscapes, they may result in different probabilities of causing catastrophic overfitting. Instances with large gradient norms are more sensitive to adversarial noise. Thus, the network may simply minimize the loss on the FGSM-perturbed examples near the boundary instead of the whole space within the adversarial budget. This leads to highly distorted loss landscapes and catastrophic overfitting. The following experiments verify our hypothesis of catastrophic overfitting in FGSM-RS and ATTA.

Metrics of Input Gradient Norm. To verify the hypothesis that instances with large gradient norms cause catastrophic overfitting, we divide the training instances into different subsets according to their gradient norms. Following the grouping method in [45], we also average the gradient norm across the training process to reduce the variance. Formally speaking, we perform FGSM-RS and ATTA to train a ResNet-18 (RN-18) on CIFAR10 for $N = 30$ epochs with $\varepsilon = 8/255$ and step size $\alpha = 10/255$. Catastrophic overfitting does not happen in this case. The average gradient norm $GN(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{N} \|\nabla_{\tilde{\mathbf{x}}_i^j} \ell(\tilde{\mathbf{x}}_i^j, y_i; \boldsymbol{\theta})\|_2$, where $\tilde{\mathbf{x}}_i^j$ is the initialization of $\mathbf{x}_i$ at the j-th epoch. For FGSM-RS, $\tilde{\mathbf{x}}_i^j$ adds random perturbation to $\mathbf{x}_i$. And for ATTA, $\tilde{\mathbf{x}}_i^j$ is the adversarial perturbation of $\mathbf{x}_i$ at the $(j-1)$-th epoch. We sort $\mathbf{x}_i$ according to $GN(\mathbf{x}_i)$ and define $\text{rank}(\mathbf{x}_i) = \frac{1}{n}\sum_{j=1}^{n} 1(GN(\mathbf{x}_j) < GN(\mathbf{x}_i))$ as the fraction of instances with smaller average gradient norm than $\mathbf{x}_i$. We divide the subsets according to $\text{rank}(\mathbf{x}_k)$: $\mathcal{D}_i^j = \{\mathbf{x}_k | \frac{10(i-1)}{n} \le \text{rank}(\mathbf{x}_k) < \frac{10j}{n}\}$. The classes of each subset are balanced. The maximum and minimum proportion of one class in all subsets is 10.86% and 8.98% in CIFAR10. In the supplementary materials, we present a comprehensive characterization of the subset splits, demonstrating their consistency across various seeds and step sizes.



(a) $\mathcal{D}_1^1$ (Instances with smallest 10% gradient norm)

(b) $\mathcal{D}_{10}^{10}$ (Instances with largest 10% gradient norm)

Fig. 2: The loss surface of the subsets $\mathcal{D}_1^1$ and $\mathcal{D}_{10}^{10}$. We average the loss of the instances from each subset. v_1 is the direction of adversarial noise and v_2 is a random direction. Figures from left to right plot the loss surface as the training step increases and each column of (a) and (b) corresponds to the same step of FGSM-RS.

Loss Landscape. We train a new RN-18 using FGSM-RS and enlarge the step size to $\alpha = 14/255$ to cause catastrophic overfitting. Figure 2 shows the loss surface of the subsets with the smallest ($\mathcal{D}_1^1$) and the largest gradient norm ($\mathcal{D}_{10}^{10}$) when the catastrophic overfitting happens. $\mathcal{D}_{10}^{10}$ first exhibits the catastrophic overfitting, where the loss surface of the input gets highly distorted and the loss function reaches its highest value in the middle of the adversarial budget. By contrast, the loss surface of $\mathcal{D}_1^1$ is much less distorted. Figure 2 infers that the subsets with large gradient norm are more likely to suffer from catastrophic overfitting.

Training with Different Subsets. We perform FGSM-RS and ATTA on different subsets of CIFAR10 with different adversarial budgets ε and step size α to show that fitting examples with larger gradient norms is more likely to cause catastrophic overfitting. We train the RN-18 on instances with small gradient norm $\mathcal{D}_1^2$, $\mathcal{D}_1^3$, $\mathcal{D}_1^4$ and instances with large gradient norm $\mathcal{D}_7^{10}$, $\mathcal{D}_8^{10}$, $\mathcal{D}_9^{10}$. While different subsets contain different numbers of instances, we keep the number of training iterations the same for a fair comparison. In Figure 3, we show

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

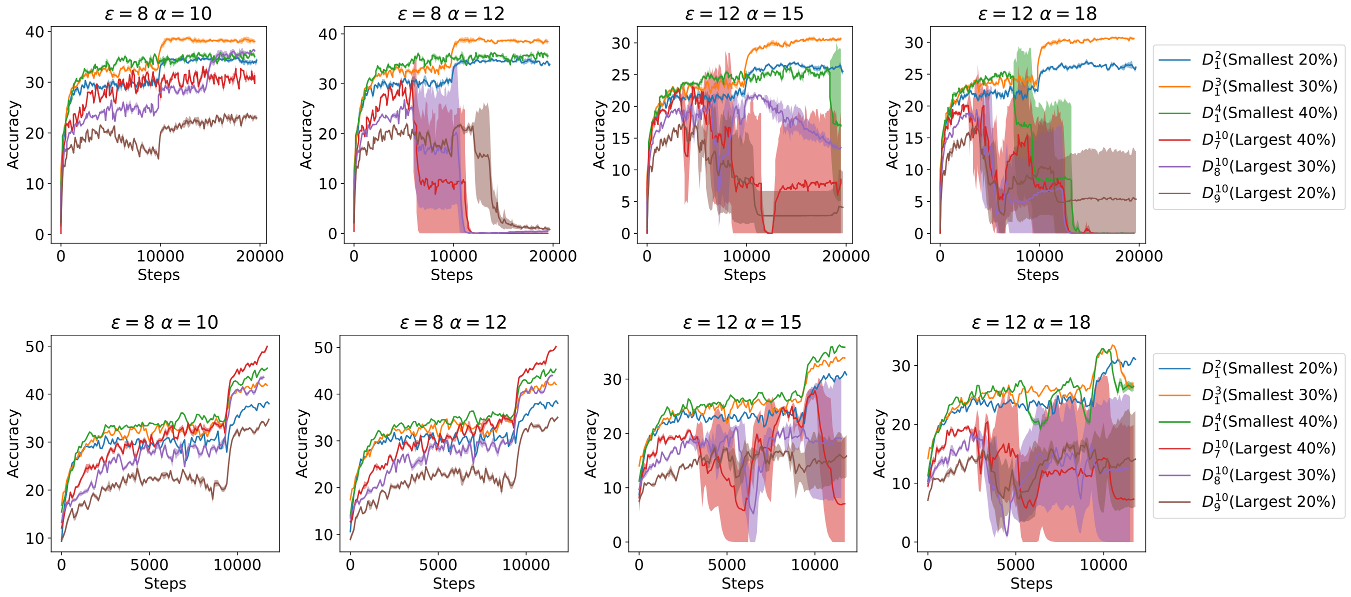
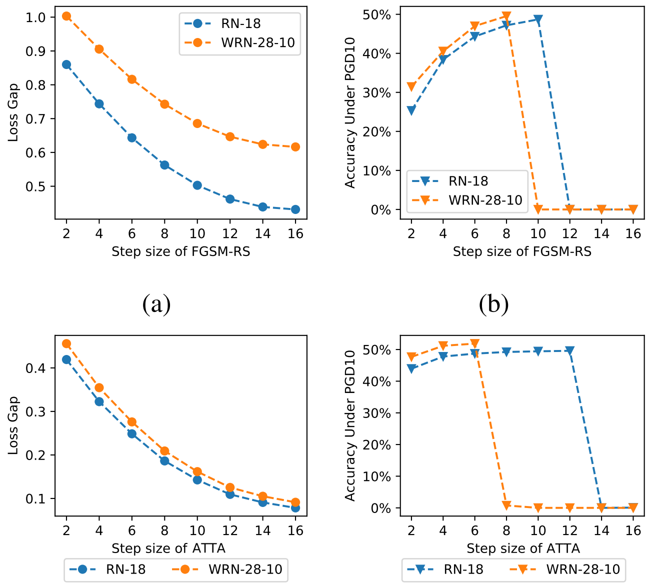


Fig. 3: The robust training accuracy curve of FGSM-RS (top) and ATTA (bottom) trained on different subsets of CIFAR10. The adversarial budgets and the step sizes are shown on top of each figure. The sudden decrease in accuracy indicates catastrophic overfitting. The shadow represents the standard deviation of accuracy calculated across five experimental runs. The plots with more different values of adversarial budgets and step sizes are provided in the *supplementary materials*.

the robust accuracy of the whole training set under PGD-10. For $\varepsilon = 8/255$ with $\alpha = 10/255$, the models trained with all subsets do not exhibit catastrophic overfitting. However, as the step size α increases, subsets with large norms first exhibit catastrophic overfitting, while catastrophic overfitting is less likely to occur in the model trained with the subsets of small gradient norm. Figure 3 indicates 1) for each subset, catastrophic overfitting is more likely to occur when increasing the step size; 2) for a fixed step size, catastrophic overfitting is less likely to happen for subsets with small gradient norms.



(a) (b)

(c) (d)

Fig. 4: (a) The loss gap of training instances between PGD10 and FGSM-RS $\ell(\mathbf{x}^{\text{PGD}}, y) - \ell(\mathbf{x}^{\text{FGSM-RS}}, y)$ with different step sizes for a FGSM-RS trained robust model; (b) The test robust accuracy of the models trained by FGSM-RS with different step sizes. (c) Similar results to (a) and (b) with training methods replaced by ATTA.

IV. Proposed Method

From our analysis in Section III, the step size of the inner maximization plays an important role in the performance of the single step methods. Overly large step size draws all perturbed noise near the boundary, causing catastrophic overfitting and thus the robust accuracy under PGD decreases to zero. However, we cannot simply reduce the step size. As shown in Figure 4, increasing step size can strengthen the adversarial attack and improves the robust accuracy. To strengthen the attack as much as possible as well as avoid catastrophic overfitting, we advocate utilizing the instance-wise step-size. The analysis in Section III reveals that we should use small step sizes for instances with large gradient norms to prevent catastrophic overfitting, and large step sizes for instances with small gradient norms to strengthen the attack. Thus, we propose to use the moving average of the gradient norm

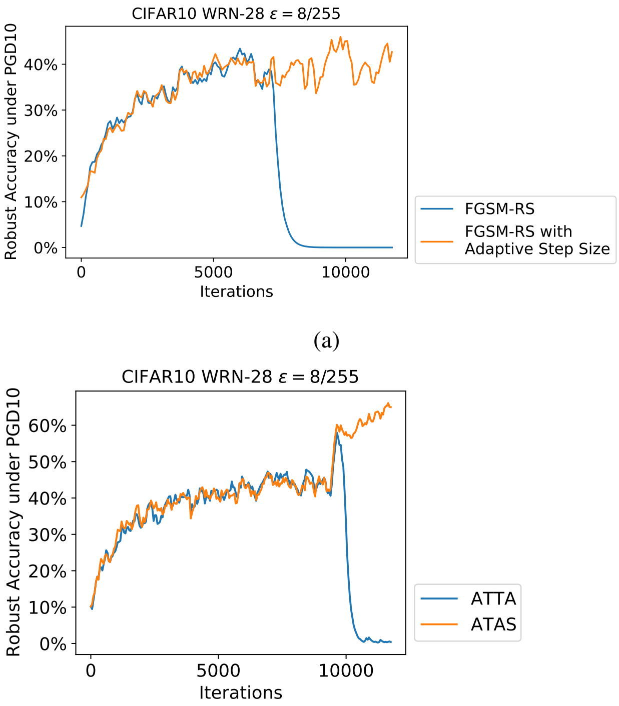
$$v_i^j = \beta v_i^{j-1} + (1-\beta)\|\nabla_{\tilde{\mathbf{x}}_i} \ell(\tilde{\mathbf{x}}_i, y_i; \boldsymbol{\theta})\|_2^2 \,, \qquad (3)$$

to adjust the step size α_i^j for the $\mathbf{x}_i$ at the j-th epoch. Here, $\tilde{\mathbf{x}}_i$ is the initialization of $\mathbf{x}_i$ and β is the momentum factor stabilizing the step size. The step size α_i^j is inversely proportional to v_i^j:

$$\alpha_i^j = \gamma / (c + \sqrt{v_i^j}) \,, \qquad (4)$$

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398



(a)

(b)

Fig. 5: (a) The robust accuracy against PGD10 attack of WideResNet-28-10 trained with FGSM-RS and FGSM-RS with adaptive step size, respectively. The average step size for the adaptive step size methods is $10.8/255$. (b) The robust accuracy against PGD10 attack of WideResNet-28-10 trained with ATTA and ATAS, respectively. The average step size for ATAS is $9.3/255$. Even if the average step sizes of adaptive methods are larger than FGSM-RS and ATTA ($\alpha = 8/255$), catastrophic overfitting does not occur in ATAS.

```
Algorithm 1 ATAS
Input: Training set $\mathcal{D}$, The model $f_{\boldsymbol{\theta}}$ with loss function $\ell$,
    Adversarial budget $\varepsilon$
Output: Optimized model $f_{\boldsymbol{\theta}^*}$
$v_i^0 = 0$ for $i = 1, \cdots, n$
$\mathbf{x}_i^0 = \mathbf{x}_i + \text{Uniform}(-\varepsilon, \varepsilon)$ for $i = 1, \cdots, n$
for $j = 1$ to $N$ do
    for $\mathbf{x}_i, y_i \in \mathcal{D}$ do
        $v_i^j = \beta v_i^{j-1} + (1-\beta) \|\nabla_{\mathbf{x}_i^{j-1}} \ell(\mathbf{x}_i^{j-1}, y_i; \boldsymbol{\theta})\|_2^2$
        $\alpha_i^j = \gamma / (c + \sqrt{v_i^j})$
        $\mathbf{x}_i^j = \Pi_{\mathcal{B}_p(\mathbf{x}_i, \varepsilon)}[\mathbf{x}_i^{j-1} + \alpha_i^j \cdot \text{sgn}(\nabla_{\mathbf{x}_i^{j-1}} \ell(\mathbf{x}_i^{j-1}, y; \boldsymbol{\theta}))]$
        $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}_i^j, y; \boldsymbol{\theta}))$
    end for
end for
```

where γ is a pre-defined learning rate and c is a constant preventing α_i^j from being too large.

We conduct preliminary experiments to incorporate the adaptive step size α_i^j with FGSM-RS, which randomly initializes the perturbation at each inner maximization step. The comparison of FGSM-RS and FGSM-RS with adaptive step size is shown in Figure 5a. It can be observed that the catastrophic overfitting does not occur with adaptive step size. In addition, the average step size of the adaptive step size method is $10.8/255$ for FGSM-RS, which is even larger than the fixed step size of $8/255$, leading to a stronger attack and better adversarial robustness.

In addition, for the inner maximization, the random initialization in FGSM-RS may limit the magnitude of perturbations for instances with small step sizes, weakening the attack strength. In order to make the whole space within the adversarial budget reachable, we further consider the previous initialization [13], which utilizes the transferability of adversarial examples and uses the adversarial perturbation obtained in the previous epoch as the initialization for the current iteration. Combined with the previous initialization, ATAS does not need large α_i^j to reach the whole ℓ_p norm ball. For each instance, we use adaptive step size α_i^j and perform the following inner maximization to obtain the adversarial examples:

$$\mathbf{x}_i^j = \Pi_{\mathcal{B}_p(\mathbf{x}_i, \varepsilon)}[\mathbf{x}_i^{j-1} + \alpha_i^j \cdot \text{sgn}(\nabla_{\mathbf{x}_i^{j-1}} \ell(\mathbf{x}_i^{j-1}, y_i; \boldsymbol{\theta}))], \quad (5)$$

where $\mathbf{x}_i^j$ is the adversarial example at the j-th epoch. Then the parameter $\boldsymbol{\theta}$ is updated with $\mathbf{x}_i^j$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}_i^j, y_i; \boldsymbol{\theta}) . \quad (6)$$

The detailed algorithm of ATAS is shown in Algorithm 1. Figure 5b shows the comparisons of ATTA [13] and ATAS, where ATAS does not suffer from catastrophic overfitting. In contrast to previous methods [14], [15] that needs large computational overhead to resolve the problem of catastrophic overfitting, the overhead of ATAS is negligible, since the input gradient $\nabla_{\mathbf{x}_i^{j-1}} \ell(\mathbf{x}_i^{j-1}, y_i; \boldsymbol{\theta})$ is already calculated in the attack step in Equation (5). Thus, calculating the pre-conditioner v_i^j and the step size α_i^j does not need additional forward-backward passes of the network. The training time of ATAS is almost the same as ATTA [13] and FGSM-RS [12].

A. *Theoretical Analysis of ATAS.*

We analyze the convergence of ATAS with L_∞ adversarial budget. The proof is deferred to the *supplementary material*. Given the objective function

$$\phi(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{x}_i, y_i; \boldsymbol{\theta}) , \quad (7)$$

the minimax problem can be formulated as follows:

$$\min_{\boldsymbol{\theta}} \max_{\mathbf{x}^* = [\mathbf{x}_1^*, \mathbf{x}_2^*, \cdots, \mathbf{x}_n^*] \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)} \phi(\boldsymbol{\theta}, \mathbf{x}^*) , \quad (8)$$

where $\mathbf{x}^*$ is the optimal adversarial example depending on $\boldsymbol{\theta}$. We consider the minimax optimization in convex-concave and smooth settings. And the loss function ℓ satisfies the following assumptions.

Assumption IV.1. *The training loss function ℓ satisfies the following constraints:*

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

1. ℓ in convex and L_θ-smooth in $\boldsymbol{\theta}$; $\boldsymbol{\theta}$ and the gradient of $\boldsymbol{\theta}$ are bounded in the L_2 norm balls

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 \leq D_{\theta,2}, \quad \frac{1}{n}\sum_{i=1}^{n} \|\nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}'_i, y_i; \boldsymbol{\theta})\|_2^2, \leq G_{\theta,2}^2 \ ,$$

where $\boldsymbol{\theta}^* = arg\,min_{\boldsymbol{\theta}} \max_{\mathbf{x}^* \in \mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\boldsymbol{\theta}, \mathbf{x}^*)$.
2. ℓ in concave and L_x-smooth in each $\mathbf{x}_i$. $\mathbf{x}_i \in \mathbb{R}^d$ is bounded in an L_∞ norm ball with $D_{x,\infty} = 2\varepsilon$. For any $\mathbf{x}$ and $\mathbf{x}'$, $\|\mathbf{x} - \mathbf{x}'\|_\infty \leq D_{x,\infty}$, and the gradients of the inputs also satisfy

$$\|\nabla_{\mathbf{x}'_i} \ell(\mathbf{x}'_i, y_i; \boldsymbol{\theta})\|_2^2 \leq G_{x_i,2}^2, \ \sum_{i=1}^{n} \|\nabla_{\mathbf{x}'_i} \ell(\mathbf{x}'_i, y_i; \boldsymbol{\theta})\|_2^2 \leq G_{x,2}^2$$

We average the trajectory of T-steps $\bar{\boldsymbol{\theta}}^T = \frac{\sum_{t=1}^T \boldsymbol{\theta}^t}{T}$ and $\bar{\mathbf{x}}^T = \frac{\sum_{t=1}^T \mathbf{x}^{t+1}}{T}$ to get the near-optimal points. It is a standard technique for analyzing stochastic gradient methods [37]. The convergence gap $\max_{\mathbf{x}^* \in \mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\bar{\boldsymbol{\theta}}^T, \mathbf{x}^*) - \max_{\mathbf{x}^* \in \mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\boldsymbol{\theta}^*, \mathbf{x}^*)$ is upper bounded by the regret $R(T)$

$$R(T) = \sum_{t=1}^{T} [\max_{\mathbf{x}^* \in \mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\boldsymbol{\theta}^t, \mathbf{x}^*) - \min_{\boldsymbol{\theta}^*} \phi(\boldsymbol{\theta}^*, \mathbf{x}^t)] \ . \quad (9)$$

Lemma IV.1. *For ℓ satisfying assumption IV.1, the objective function ϕ defined in Equation* (7)

$$\max_{\mathbf{x}^* \in \mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\bar{\boldsymbol{\theta}}^T, \mathbf{x}^*) - \min_{\boldsymbol{\theta}^*} \max_{\mathbf{x}^* \in \mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\boldsymbol{\theta}^*, \mathbf{x}^*) \leq \frac{R(T)}{T}$$

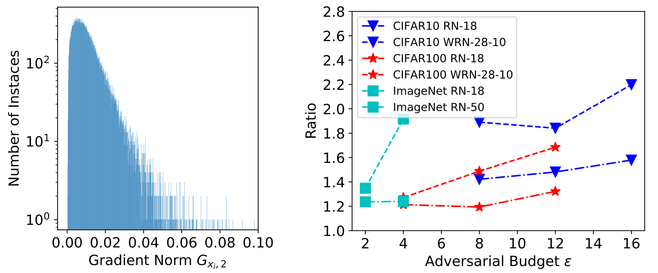
Adaptive Stochastic Gradient Descent Block Coordinate Ascent (A-SGD$_{\mathbf{BCA}}$). ATAS can be formulated as A-SGD$_{\text{BCA}}$, which randomly picks an instance $\mathbf{x}_k$ at the step t, applying stochastic gradient descent to the parameter $\boldsymbol{\theta}$ and adaptive block coordinate ascent to the input $\mathbf{x}$. Unlike SGDA [28], where all dimensions of $\mathbf{x}$ get updated in each iteration, A-SGD$_{\text{BCA}}$ only updates some dimensions of $\mathbf{x}$. A-SGD$_{\text{BCA}}$ first calculates the pre-conditioner v_i^t as

$$v_k^{t+1} = \begin{cases} \beta v_i^t + (1-\beta)\|\nabla_{\mathbf{x}_i^t} \ell(\mathbf{x}_i^t, y_k; \boldsymbol{\theta}^t)\|_2^2 & i = k \\ v_i^t & i \neq k \end{cases}$$
$$\hat{v}_i^{t+1} = \max(\hat{v}_i^t, v_i^{t+1}) \ .$$

Then $\mathbf{x}$, $\boldsymbol{\theta}$ are optimized with

$$\mathbf{x}_i^{t+1} = \begin{cases} \Pi_{\mathcal{B}_\infty(\mathbf{x}_i,\varepsilon)}[\mathbf{x}_i^t + \frac{\eta_x}{\sqrt{\hat{v}_i^{t+1}}} \nabla_{\mathbf{x}_i^t} \ell(\mathbf{x}_i^t, y_i; \boldsymbol{\theta}^t)] & i = k \\ \mathbf{x}_i^t & i \neq k \end{cases}$$
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_\theta \nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}_k^{t+1}, y_k; \boldsymbol{\theta}^t) \ .$$

The difference between A-SGD$_{\text{BCA}}$ and ATAS is $\hat{v}_k^t$. To prove the convergence of A-SGD$_{\text{BCA}}$, the pre-conditioner needs to be non-decreasing. Otherwise, ATAS may not converge like ADAM [41]. However, the non-convergent version of ADAM actually works better for neural networks in practice [40]. Therefore, ATAS still uses v_k^t as the pre-conditioner.



(a) (b)

Fig. 6: (a) Histogram of $G_{x_i,2}$ of a ResNet-18 trained CIFAR10 with $\varepsilon = 8/255$. (b) Ratio $\frac{\sum_{i=1}^n G_{x_i,2}^2}{n} \Big/ (\frac{\sum_{i=1}^n G_{x_i,2}}{n})^2$ for different datasets and network architectures with different ε.

Theorem IV.1 (Regret Bound for A-SGD$_{\text{BCA}}$)**.** *Under Assumption IV.1, with $\eta_\theta = \frac{D_{\theta,2}}{G_{\theta,2}\sqrt{T}}$ and $\eta_x = \frac{\sqrt{d}D_{x,\infty}}{\sqrt{T}(1-\beta)^{-1/4}}$, the regret of A-SGD$_{BCA}$ is bounded by:*

$$R^{\text{A-SGD}_{\text{BCA}}}(T) \leq G_{\theta,2} D_{\theta,2}\sqrt{T} + \frac{D_{x,\infty}\sum_{i=1}^n G_{x_i,2}\sqrt{dT}}{n(1-\beta)^{1/4}} + \frac{dL_x D_{x,\infty}^2}{2n^2\sqrt{1-\beta}}$$

Comparison with the Non-adaptive Version (SGD$_{\mathbf{BCA}}$). The non-adaptive version of ATAS reduces to ATTA [13], which can be formulated as the Stochastic Gradient Descent Block Coordinate Ascent (SGD$_{\text{BCA}}$):

$$\mathbf{x}_i^{t+1} = \begin{cases} \Pi_{\mathcal{B}_\infty(\mathbf{x}_i,\varepsilon)}[\mathbf{x}_i^t + \eta_x \nabla_{\mathbf{x}_i^t} \ell(\mathbf{x}_i^t, y_i; \boldsymbol{\theta}^t)] & i = k \\ \mathbf{x}_i^t & i \neq k \end{cases}$$
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_\theta \nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}_k^{t+1}, y_k; \boldsymbol{\theta}^t) \ ,$$

Theorem IV.2 (Regret Bound for SGD$_{\text{BCA}}$)**.** *Under assumption IV.1, with constant learning $\eta_\theta = \frac{D_{\theta,2}}{G_{\theta,2}\sqrt{T}}$ and $\eta_x = \frac{\sqrt{nd}D_{x,\infty}}{G_{x,2}\sqrt{T}}$, the regret $R^{\text{SGD}_{\text{BCA}}}(T)$ of SGD$_{BCA}$ is bounded by:*

$$R^{\text{SGD}_{\text{BCA}}}(T) \leq G_{\theta,2} D_{\theta,2}\sqrt{T} + G_{x,2} D_{x,\infty}\sqrt{\frac{dT}{n}} + \frac{dL_x D_{x,\infty}^2}{2n}$$

Theorem IV.1 and IV.2 shows that A-SGD$_{\text{BCA}}$ converges faster than SGD$_{\text{BCA}}$. When T is large, the third term of the regret in both SGD$_{\text{BCA}}$ and A-SGD$_{\text{BCA}}$ is negligible. Consider their first terms are the same, the main difference is the regret bound about $\mathbf{x}$ in the second term: $G_{x,2}D_{x,\infty}\sqrt{\frac{dT}{n}}$ and $\frac{D_{x,\infty}\sum_{i=1}^n G_{x_i,2}\sqrt{dT}}{n(1-\beta)^{1/4}}$. The ratio between them is

$$\text{Ratio} = \frac{1}{(1-\beta)^{\frac{1}{4}}}\sqrt{\frac{\sum_{i=1}^n G_{x_i,2}^2}{n} \Big/ (\frac{\sum_{i=1}^n G_{x_i,2}}{n})^2} \quad (10)$$

The Cauchy-Schwarz inequality indicates the ratio is always larger than 1 for $\beta = 0$. The gap between A-SGD$_{\text{BCA}}$ and SGD$_{\text{BCA}}$ gets larger when $G_{x_i,2}$ has long-tailed distribution, which demonstrates the relatively faster convergence of ATAS than the non-adaptive counterparts. We show the empirical

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

TABLE I: Accuracy and training time of different methods on CIFAR10, CIFAR100 and ImageNet. ATAS improves the robust accuracy under various attacks including PGD10, PGD50 and AutoAttack (AA). The method "*PGD10*" refers to the standard AT using PGD10 for the inner maximization. Note that, we do not have enough computational resources to perform standard AT and SSAT on ImageNet because of computational complexity. Besides, we are unable to train the ResNet-50 on ImageNet with FGSM-GA as its memory requirement exceeds the maximum GPU memory of our devices (*i.e.* NVIDIA Tesla V100). For CIFAR10 and CIFAR100, the training time is evaluated on a single GPU. And we use two GPUs to train the models for ImageNet. We use default step size from the original papers for the baselines so that catastrophic overfitting seldom happens in these methods.

(a) CIFAR10 with $\varepsilon = 8/255$. The accuracy with $\varepsilon = 12/255$ and $16/255$ is in the *supplementary material.*

Methods	ResNet-18					WideResNet-28-10				
	Clean	PGD10	PGD50	AA	Time(h)	Clean	PGD10	PGD50	AA	Time(h)
PGD10	*80.13*	*50.59*	*48.94*	*45.97*	*1.23*	*85.00*	*55.51*	*53.53*	*51.27*	*8.49*
FreeAT	78.37	40.90	39.02	36.00	0.33	84.54	46.09	43.80	41.19	2.31
YOPO	74.72	37.51	35.79	33.21	0.28	82.92	44.62	42.14	40.23	1.90
FGSM-RS	83.99	48.99	46.36	42.95	**0.22**	80.21	0.01	0.00	0.00	1.67
FGSM-GA	80.10	49.14	47.21	43.44	0.57	75.84	45.57	43.28	39.44	3.82
SSAT	**88.83**	42.31	38.99	37.06	0.61	**90.40**	44.04	40.40	38.82	3.53
ATTA	82.16	47.47	45.32	42.51	0.30	85.90	51.52	48.94	46.84	1.70
NuAT	76.68	49.36	48.13	**45.96**	0.35	79.60	52.59	**51.45**	47.86	1.86
SubAT	77.43	46.29	45.03	41.49	0.41	80.44	49.33	48.25	44.65	2.13
FGSM-SDI	78.15	48.79	47.52	42.04	0.48	73.58	44.84	43.79	39.40	2.51
FGSM-PGI	67.27	46.47	45.98	39.22	0.38	71.02	49.03	48.63	43.27	1.89
ATAS	81.22	**50.03**	**48.18**	45.38	0.30	85.96	**53.43**	51.03	**48.72**	**1.63**

(b) CIFAR100 with $\varepsilon = 8/255$. The accuracy with $\varepsilon = 4/255$ and $12/255$ is in the *supplementary material.*

Methods	ResNet-18					WideResNet-28-10				
	Clean	PGD10	PGD50	AA	Time(h)	Clean	PGD10	PGD50	AA	Time(h)
PGD10	*54.08*	*28.03*	*27.23*	*23.04*	*1.32*	*60.04*	*31.70*	*30.67*	*27.11*	*8.53*
FreeAT	50.56	19.57	18.58	15.09	0.33	59.38	24.41	23.00	19.60	2.30
YOPO	51.55	20.65	19.17	16.05	0.29	50.35	19.44	18.36	15.43	1.92
FGSM-RS	59.35	26.40	24.29	19.73	**0.21**	51.83	0.00	0.00	0.00	**1.60**
FGSM-GA	50.61	24.48	24.07	19.42	0.57	54.29	25.86	24.56	20.74	3.80
SSAT	**71.03**	9.79	4.80	1.09	0.62	**75.01**	0.21	0.01	0.00	3.50
ATTA	57.21	25.76	24.90	21.03	0.28	63.04	28.93	27.18	24.42	1.63
NuAT	24.96	17.10	16.93	13.31	0.32	22.58	18.73	18.55	14.82	1.76
SubAT	44.16	21.81	21.37	16.95	0.45	50.11	12.67	11.19	7.08	2.18
FGSM-SDI	53.03	25.42	24.79	20.33	0.50	60.66	29.28	**28.39**	24.11	2.56
FGSM-PGI	40.07	21.14	20.81	16.51	0.40	49.09	7.22	3.62	0.47	1.96
ATAS	55.49	**27.68**	**26.60**	**22.62**	0.31	62.34	**29.89**	28.35	**25.03**	1.61

(c) ImageNet with $\varepsilon = 2/255$. The accuracy with $\varepsilon = 4/255$ is available in the *supplementary material.*

Methods	ResNet-18					ResNet-50				
	Clean	PGD10	PGD50	AA	Time(h)	Clean	PGD10	PGD50	AA	Time(h)
FreeAT	58.80	35.56	34.78	31.77	**40.01**	65.81	44.12	43.34	40.80	**108.3**
YOPO	47.69	28.50	28.10	25.22	48.22	55.68	33.46	32.19	29.56	111.8
FGSM-RS	55.26	37.33	36.98	33.28	43.46	67.83	46.12	45.56	43.58	115.0
FGSM-GA	37.01	24.15	24.05	19.98	182.7	/	/	/	/	/
ATTA	58.32	39.62	38.32	36.08	45.83	66.62	48.27	47.65	45.00	111.7
NuAT	51.98	21.52	19.35	16.08	46.53	55.73	24.17	21.86	19.53	114.7
SubAT	31.38	20.95	20.90	18.27	64.18	36.78	25.09	23.19	21.13	180.8
FGSM-SDI	60.07	40.23	**40.13**	37.16	63.21	65.21	48.14	48.04	44.42	132.6
FGSM-PGI	50.83	37.28	37.26	31.31	44.82	68.42	49.00	**48.87**	44.90	114.2
ATAS	**61.20**	**40.84**	39.86	**37.25**	45.70	**69.10**	**49.05**	48.05	**46.01**	120.4

histogram of $G_{x_i,2}$ of a RN-18 and the ratio in Figure 6, which demonstrates the long-tailed distribution for common datasets.

V. Experiments

Baselines. We compare ATAS with the SOTA fast AT algorithms including FreeAT [30], YOPO [31], FGSM-RS [12], FGSM-GA [14], SSAT [15], NuAT [24], SubAT [32], FGSM-SDI [33], FGSM-PGI [35] and ATTA [13]. We also compare ATAS with standard AT whose inner maximization is solved by PGD10, providing a reference for the ideal performance.

Attack Methods. We consider three attacks: PGD10, PGD50 [7] and AutoAttack (AA) [18]. Square Attack, a black-box attack, is included in AutoAttack to eliminate the effect of gradient masking.

Experimental Settings. ATAS uses the techniques proposed in ATTA [13]: the adversarial perturbations are transformed according to the data augmentation and get reset every several epochs. The previous initialization is stored in the GPU memory and thus brings negligible storing latency to ATAS. We consider adversarial attacks with the ℓ_∞-norm budget. We evaluate fast AT algorithms on CIFAR10 and CIFAR100 [16] with

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

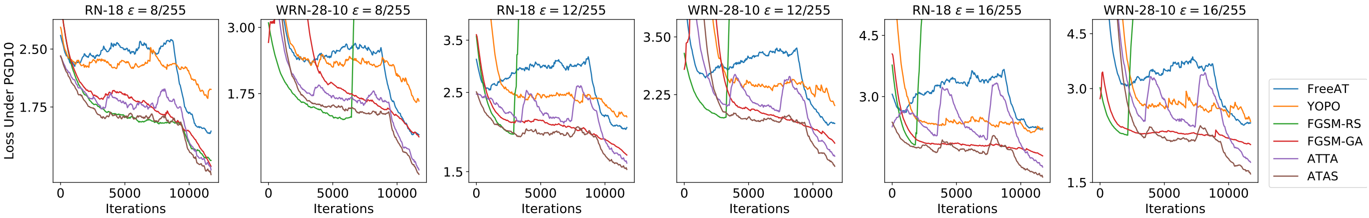


Fig. 7: Robust training cross-entropy loss under PGD10 of CIFAR10 with different network architectures and adversarial budgets. The curve is smoothed to clearly show the convergence.

WideResNet-28-10 (WRN-28-10) [46] and ResNet-18 (RN-18), and on ImageNet [17] with ResNet-18 (RN-18) and ResNet-50 (RN-50). In this study, we have adhered to the default splits for all the datasets employed. Specifically, for CIFAR10 and CIFAR100, our training set consists of 50,000 images, while 10,000 images are reserved for evaluation purposes. As for ImageNet, we utilized an extensive collection of 1,281,167 training instances spanning across 1000 distinct classes, and our testing was conducted on the 50,000 images from the validation set. While early stopping is widely used in the standard AT [11], the computational overhead to perform PGD attack on a separate validation set is large. Besides, considering the small budget of training time in fast AT, even if early stopping is applied to terminate the training before catastrophic overfitting occurs, the training is far from convergence, resulting in poor performance [14]. Therefore, we follow the previous works [12]–[14] and do not use early stopping. We set $\beta = 0.5$ and $\gamma/c = 16/255$, which is close to the adversarial budget. And we set $c = 0.01$ for CIFAR10 and CIFAR100 and $c = 0.1$ for ImageNet. More detailed experiment settings are in the Appendix, and additional experiments are deferred to the *supplementary materials*.

A. Convergence.

Figure 7 shows the curve of the training loss $\max_{\mathbf{x}^*=[\mathbf{x}_1^*,\cdots,\mathbf{x}_n^*]\in\mathcal{B}_\infty(\mathbf{x},\varepsilon)} \phi(\boldsymbol{\theta},\mathbf{x}^*)$ on CIFAR10 with different network architectures and different adversarial budgets, where $\mathbf{x}^*$ is approximated by PGD10 and the objective function ϕ is approximated by mini-batches of training instances at each step. ATAS achieves smaller robust training loss at the end of training, demonstrating the faster convergence of ATAS than ATTA and other baselines.

In Table III, we show the relationship between the *Ratio* in Equation (10) and the convergence gap $\ell_{\text{ATTA}} - \ell_{\text{ATAS}}$ and convergence ratio $\ell_{\text{ATTA}}/\ell_{\text{ATAS}}$ in the last epoch of training. Here, ℓ is the loss of each method. The ratio is obtained from Figure 6b for CIFAR10 with ResNet-18. It shows that larger *Ratio* (more long-tailed distribution) leads to larger convergence gap between ATTA and ATAS.

B. Robust Accuracy

We provide our main results in Table I, showing the robust accuracy of CIFAR10, CIFAR100 and ImageNet, respectively. Table II shows the robust accuracy under AutoAttack for different adversarial budgets.

CIFAR10 and CIFAR100. As shown in Table Ia, The robust accuracy of FreeAT and YOPO is much lower than the other methods. While FGSM-RS maintains non-trivial robust accuracy when using RN-18, it suffers from catastrophic overfitting when using large networks such as WRN-28-10. The regularizer in FGSM-GA prevents catastrophic overfitting. However, it may over-regularize the network so that the clean accuracy and the robust accuracy decrease on WRN-28-10. In addition, the regularizer also brings computational overhead: FGSM-GA needs nearly double the training time compared with other methods. ATAS achieves the best robust accuracy among all fast AT algorithms while keeping the training time nearly the same. Furthermore, for small networks like RN-18, the performance of ATAS is on par with standard AT (PGD10) but needs only one-fifth of the training time. Despite incurring a lower computational overhead compared to FGSM-SDI, ATAS demonstrates superior robust accuracy in 13 out of the 16 experiments as shown in Table II. Notably, ATAS exhibits superior robustness in the case of large networks such as WideResNet, which achieve higher robust accuracy than smaller networks. Table Ib shows the robust accuracy on CIFAR100 and ATAS also outperforms other algorithms. Catastrophic overfitting also happens in SSAT even if the losses of inner points are checked.

ImageNet. ATTA and ATAS need to memorize the adversarial noise for the whole training set. Since frequently loading and storing from the disks significantly lowers the training speed, all perturbations should be stored in the memory. Thus, we utilize the local property of the adversarial examples [47] and only store the interpolated perturbation in the memory. We resize the perturbations from 224 × 224 to 32 × 32 for storage and up-sample it back when used as the initialization for the next epoch. The detailed algorithm is available in the Appendix. Table Ic shows the robust accuracy on ImageNet on $\varepsilon = 2/255$. Although both FGSM-SDI and FGSM-PGI achieve higher robust accuracy for PGD50, it is important to note that ATAS demonstrates higher robust accuracy when subjected to the strongest AutoAttack. As the robustness should calculated based on the worst-case accuracy, this result showcases the superior true robustness of ATAS compared to all the baseline methods. FGSM-GA needs the calculate the second-order gradient of the parameters, which needs a huge amount of GPU memory. Thus, we could not train a big network such as ResNet-50 on ImageNet.

Robust accuracy at different adversarial budgets. Table II shows the robust accuracy of fast AT algorithms under

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

TABLE II: Robust Accuracy under AutoAttack in CIFAR10, CIFAR100 and ImageNet.

(a) Robust Accuracy under AutoAttack for small models.

Methods	CIFAR10 ResNet18			CIFAR100 ResNet18			ImageNet ResNet18	
	$\varepsilon = 8/255$	$\varepsilon = 12/255$	$\varepsilon = 16/255$	$\varepsilon = 4/255$	$\varepsilon = 8/255$	$\varepsilon = 12/255$	$\varepsilon = 2/255$	$\varepsilon = 4/255$
PGD10	*45.97*	*32.99*	*23.34*	*37.75*	*23.04*	*15.41*	/	/
FreeAT	36.00	19.95	10.04	28.26	15.09	8.33	31.77	15.90
YOPO	33.21	17.16	8.66	31.45	16.05	7.41	25.22	10.30
FGSM-RS	42.95	0.00	0.00	36.35	19.73	0.00	33.28	21.11
FGSM-GA	43.44	28.57	18.92	32.03	19.42	12.14	19.98	10.94
SSAT	37.06	0.03	0.00	29.81	1.09	0.09	/	/
ATTA	42.51	27.85	17.02	36.03	21.03	12.97	36.08	22.47
NuAT	**45.96**	22.45	0.08	33.60	13.31	6.41	16.08	8.26
Sub-AT	41.49	26.01	14.48	31.85	16.95	12.43	18.27	10.76
FGSM-SDI	42.04	**30.74**	20.27	37.29	20.33	**14.73**	37.16	23.85
FGSM-PGI	39.22	25.80	16.01	29.04	16.51	10.85	31.31	19.65
ATAS	45.38	30.56	**21.09**	**37.30**	**22.62**	14.41	**37.25**	**24.13**

(b) Robust Accuracy under AutoAttack for large models.

Methods	CIFAR10 WideResNet-28-10			CIFAR100 WideResNet-28-10			ImageNet ResNet50	
	$\varepsilon = 8/255$	$\varepsilon = 12/255$	$\varepsilon = 16/255$	$\varepsilon = 4/255$	$\varepsilon = 8/255$	$\varepsilon = 12/255$	$\varepsilon = 2/255$	$\varepsilon = 4/255$
PGD10	*51.27*	*36.95*	*27.24*	*43.30*	*27.11*	*18.13*	/	/
FreeAT	41.19	18.98	12.53	35.52	19.60	10.64	40.80	22.40
YOPO	40.23	25.42	11.00	35.39	15.43	9.68	29.56	11.83
FGSM-RS	0.00	0.00	0.00	39.64	0.00	0.00	43.58	0.00
FGSM-GA	39.44	29.01	11.44	41.03	20.74	14.77	/	/
SSAT	38.82	0.00	0.00	33.34	0.00	0.00	/	/
ATTA	46.84	29.85	19.11	40.99	24.42	15.24	45.00	29.46
NuAT	47.86	0.16	0.02	37.53	14.82	5.17	19.53	9.41
Sub-AT	44.65	28.05	15.68	35.85	7.08	12.58	21.13	11.98
FGSM-SDI	39.40	30.30	20.98	**41.80**	24.11	15.65	44.42	30.04
FGSM-PGI	43.27	29.40	17.76	0.15	0.47	10.95	44.90	28.51
ATAS	**48.72**	**33.58**	**22.58**	41.32	**25.03**	**16.27**	**46.01**	**30.07**

TABLE III: Convergence gap and the ratio on CIFAR10 with ResNet-18.

Ratio	1.4 (ε=8/255)	1.5 (ε=12/255)	1.6 (ε=16/255)
$\ell_{\text{ATTA}} - \ell_{\text{ATAS}}$	0.05	0.10	0.12
$\ell_{\text{ATTA}}/\ell_{\text{ATAS}}$	1.03	1.11	1.13

TABLE IV: Ablation study of hyperparameters γ (left) and c (right) on CIFAR10 and RN-18 under AA.

$\gamma/0.01 * 255$	12	14	16	18	20
$\varepsilon = 8/255$	45.20	45.21	45.38	45.50	45.60
$\varepsilon = 12/255$	30.84	31.06	30.56	31.21	31.04
$\varepsilon = 16/255$	21.38	21.23	21.09	21.13	20.94
c	0.005	0.007	0.01	0.02	0.04
$\varepsilon = 8/255$	45.01	45.28	45.38	45.52	45.48
$\varepsilon = 12/255$	30.08	30.80	30.56	30.69	30.52
$\varepsilon = 16/255$	20.36	20.84	21.09	21.07	20.48

AutoAttack on different datasets, network architectures and adversarial budgets. The robust accuracy decreases when enlarging the adversarial budget, but ATAS always outperforms all the baselines for different adversarial budgets, datasets and network architectures. This demonstrates that the improvement of ATAS is consistent.

Repeated Training of ATAS Figure 8 shows the five times repeat of ATAS and FGSM-RS [12]. All five experiments show that catastrophic overfitting occurs in FGSM-RS with WRN-28-10 while ATAS is a stable method that does not suffer from catastrophic overfitting.

Ablation Study. Table IV provides the ablation study on hyperparameters, showing that ATAS is not sensitive to them. Besides, as the only difference between ATAS and ATTA is the step size, the superior performance of ATAS over ATTA forms an ablation study to demonstrate the effectiveness of the adaptive step size.

C. Understanding of ATAS

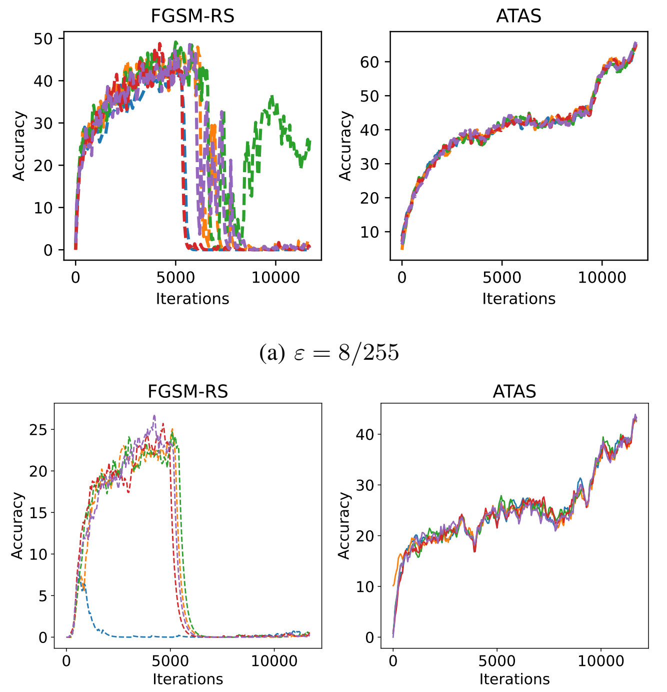
Landscape of ATAS. Figure 9 shows the loss landscapes of 3 random instances after training of ATAS. ATAS does not have distorted loss landscapes, which means it does not suffer from catastrophic overfitting.

Steps size and Gradient Norm Figure 10 plots the changes of gradient norm and step size for ATAS after 5 epochs. We divide CIFAR10 into 10 subsets according to their gradient norm and plot the gradient norm and step size for $\mathcal{D}_1^1$, $\mathcal{D}_5^5$ and $\mathcal{D}_{10}^{10}$, which has the smallest, medium and largest input gradient norm among 10 subsets. The figure shows that the input gradient norm and step size is relatively stable for each subset along the training process. It shows that the input gradient norm is more like a property of training instances themselves, which is consistent with our motivation. It is worth noting that the sudden changes in gradient norm are the result of the initialization reset used in ATTA.

Diversity of Adversarial Noises Figure 11 shows the level of adversarial noise stored in ATAS in the last epoch. As there exist some pixels that reach the ℓ_∞ boundary for each instance, the ℓ_∞ norms of nearly all instances' perturbations are $8/255$. However, the ℓ_2 norm, which can measure whether all pixels reach the ℓ_∞ boundary, is diverse for different instances. It shows that ATAS lowers the adversarial at different regions

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398



(a) $\varepsilon = 8/255$

(b) $\varepsilon = 16/255$

Fig. 8: Five times repeated training accuracy under PGD10 on CIFAR10 with WideResNet-28-10 and $\varepsilon = 8/255$ and $\varepsilon = 16/255$.

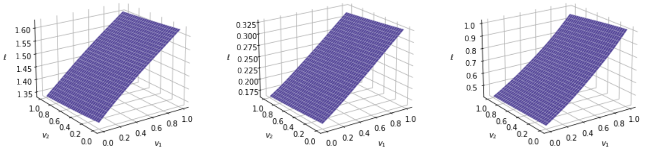


Fig. 9: Loss landscape of ATAS on CIFAR10 with ResNet-18 and $\varepsilon = 8/255$ for 3 random instances.

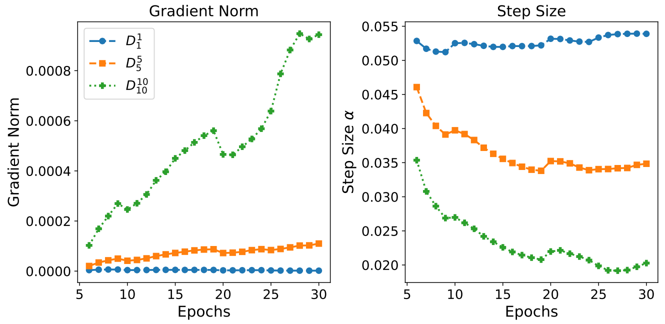


Fig. 10: The input gradient norm and step size for CIFAR10 with $\varepsilon = 8/255$ after 5 epochs on WideResNet-28-10.

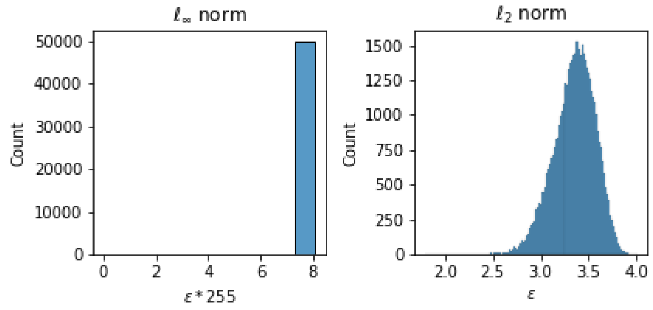


Fig. 11: The histogram of the norms of the adversarial noise for CIFAR10 with $\varepsilon = 8/255$ in the last epoch of ATAS.

within the adversarial budget instead of only forcing robustness near the boundary.

Supplementary materials include 1) the proof of Lemma IV.1, Theorem IV.1, and Theorem IV.2; 2) additional experimental results including the robust accuracy under more evaluated adversarial budgets on CIFAR10, CIFAR100, and ImageNet, the convergence curve of ImageNet, and catastrophic overfitting in FGSM-RS with more evaluated step sizes.

VI. Conclusion

In this paper, we investigate catastrophic overfitting from the perspective of training instances and show that instances with large gradient norms are more likely to cause catastrophic overfitting in the single-step fast AT methods. This finding motivates the adaptive training method, ATAS, which applies the adaptive step size of inner maximization inversely proportional to the input gradient norm. We theoretically analyze the convergence of ATAS, showing that our method converges faster than the non-adaptive counterpart especially when the distribution of input gradient norm is long-tailed. Extensive experiments on CIFAR10, CIFAR100, and ImageNet with different network architectures and adversarial budgets show that ATAS mitigates catastrophic overfitting and achieves higher robust accuracy under various strong attacks.

Appendix A
Algorithms for ATAS in ImageNet

In the experiments of ATTA and ATAS, we utilize the local property of the adversarial examples [47], [48] and only store the interpolated perturbation in the memory. We resize the perturbations from 224 × 224 to 32 × 32 for storage in the memory and up-sample it back when using it as the initialization for the next epoch. The detailed algorithm is shown in Algorithm 2.

Appendix B
Details for the Experiments

As we focus on fast AT, we reduce the training epochs like [12], [14]. For single-step methods FGSM-RS, FGSM-GA, ATTA and ATAS, the training lasts for 30 epochs on CIFAR10 and CIFAR100, and 90 epochs on ImageNet. For FreeAT and YOPO, we keep the number of the forward-backward passes the same as the single-step methods so that the total training time of these methods will be similar. We use two kinds of learning rate scheduler: piece-wise decay used in [13] and cyclic learning rate used in [12], and choose the best scheduler for each method.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

Algorithm 2 ATAS for ImageNet

```
Input: Training set $\mathcal{D}$, The model $f_{\boldsymbol{\theta}}$ with loss function $\ell$,
    Adversarial budget $\varepsilon$, Hyperparameters $\gamma, \eta, c, N$
Output: Optimized model $f_{\boldsymbol{\theta}^*}$
$v_i^0 = 0$ for $i = 1, \cdots, n$
$\delta_i^0 = \text{Uniform}(-\varepsilon, \varepsilon)$ for $i = 1, \cdots, n$
Resize $\delta_i^0$ to $32 \times 32$ for $i = 1, \cdots, n$ and store them.
for $j = 1$ to $N$ do
    for $\mathbf{x}_i, y_i \in \mathcal{D}$ do
        Resize $\delta_i^{j-1}$ to $224 \times 224$
        $\mathbf{x}_i^{j-1} = \mathbf{x}_i + \delta_i^{j-1}$
        $v_i^j = \beta v_i^{j-1} + (1-\beta)\|\nabla_{\mathbf{x}_i^{j-1}} \ell(\mathbf{x}_i^{j-1}, y_i; \boldsymbol{\theta})\|_2^2$
        $\alpha_i^j = \gamma / (c + \sqrt{v_i^j})$
        $\mathbf{x}_i^j = \Pi_{\mathcal{B}_p(\mathbf{x}_i, \varepsilon)}[\mathbf{x}_i^{j-1} + \alpha_i^j \cdot \text{sgn}(\nabla_{\mathbf{x}_i^{j-1}} \ell(\mathbf{x}_i^j, y; \boldsymbol{\theta}))]$
        $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}_i^j, y; \boldsymbol{\theta}))$
        $\delta_i^j = \mathbf{x}_i^j - \mathbf{x}_i$
        Resize $\delta_i^j$ to $32 \times 32$ and store it.
    end for
end for
```

FreeAT. We use the default hyperparameters from [30] except training epochs to make fair comparison between different methods. We select the best number of batch replaying from [30]. For CIFAR10 and CIFAR100, we use Free-8 in their paper (Free-m means the number of batch replaying is m) and train the network for 10 epochs. For ImageNet, we use Free-4 and train the network for 45 epochs.

YOPO. We use YOPO-5-3 in [31] as it achieves the best performance. The training lasts for 12 epochs for CIFAR10 and CIFAR100. For ImageNet, the training lasts for 36 epochs to make the training time similar to other methods. Other hyperparameters are the same as the original paper [31].

FGSM-RS. We directly download the code from the official repository[1]. The training lasts for 30 epochs for CIFAR10 and CIFAR100, and 90 epochs for ImageNet. Following the hyperparameters in the paper, the step size $\alpha = 1.25\varepsilon$. Other hyperparameters are the same as their paper.

FGSM-GA. We directly download the code from the official repository[2]. The training lasts for 30 epochs for CIFAR10 and CIFAR100, and 90 epochs for ImageNet. Other hyperparameters are the same. For the experiments not involved in their paper, we keep them same as the experiments of CIFAR10 except for the hyperparameter λ balancing the gradient align regularizer, which also varies for different datasets and adversarial budgets in their code. λ for CIFAR10 is provided in their code. For CIFAR100 and ImageNet, we run several experiments and provide the result with best λ. These λ are provided in Table V.

SSAT. We directly download the code from the official repository[3]. The training lasts for 30 epochs for CIFAR10 and CIFAR100. And we use the check points $c = 3$, which achieves the best performance in their paper.

TABLE V: Hyperparameter λ for FGSM-GA

(a) CIFAR100

ε	$4/255$	$8/255$	$12/255$
λ	0.2	0.5	1.0

(b) ImageNet

ε	$2/255$	$4/255$
λ	0.005	0.01

ATTA. We follow the hyperparameters setting for ATTA-1 in [13] and set the step size $\alpha = 4/255$. We reduce the number of training epochs to 30 for CIFAR10 and CIFAR100. And the epochs of piece-wise learning rate are rescheduled accordingly. The learning rate η starts at 0.1 and decays to 0.01 and 0.001 at the 24th and 28th epochs. The training of ImageNet lasts for 90 epochs and the learning rate also starts at 0.1 and decays to 0.01 and 0.001 at the 50th and 75th epochs. The weight decay is 5×10^{-4} for CIFAR10 and CIFAR100. For ImageNet, it is 1×10^{-4}. The batch size is 128 for all the experiments. Other hyperparameters are the same as their paper.

NuAT and SubAT. We download the directly download the code from the official repository. We only alter the the epochs of training and fix other hyperparameters as the same.

FGSM-SDI and FGSM-PGI. To ensure a fair comparison among various methods, we standardized the number of epochs to 30 for both CIFAR10 and CIFAR100 datasets. Additionally, for the ImageNet dataset, we set the batch size to 128, aligning it with the same number of steps used in prior experiments. All remaining hyperparameters were kept consistent with those outlined in the provided code.

ATAS. The hyperparameters γ and c are used to control the minimum and maximum step size for the training instances. When the moving average of gradient norm $v_i^j \to 0$, the step size $\alpha_i^j = \gamma/c$. We choose $\gamma/c = 16/255$, which is close to the adversarial budget. And c should be close to the magnitude of v_i^j. As the gradient norm increases with the dimension of the inputs, c should be larger for ImageNet. Therefore, we set $c = 0.01$ and for CIFAR10 and CIFAR100 and we let $c = 0.1$ for ImageNet. Momentum of gradient norm β is set to 0.5 for all the experiments. ATAS is not sensitive to the choice of hyperparameters. Other hyperparameters are the same as ATTA.

Environments of the Experiments All the training time is evaluated on a machine with *Intel Xeon 8255C* and *NVIDIA Tesla V100*. For CIFAR10 and CIFAR100, we use a single GPU. For ImageNet, we use two GPUs. We run all the experiments with *Pytorch 1.4*.

[1] https://github.com/locuslab/fast_adversarial

[2] https://github.com/tml-epfl/understanding-fast-adv-training

[3] https://github.com/Harry24k/catastrophic-overfitting

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[2] J. Wang, A. Liu, X. Bai, and X. Liu, "Universal adversarial patch attack for automatic checkout using perceptual and attentional bias," *IEEE Transactions on Image Processing*, vol. 31, pp. 598–611, 2021.

[3] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=SyJ7ClWCb

[4] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

[5] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, "Image super-resolution as a defense against adversarial attacks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1711–1724, 2019.

[6] S. Zhang, H. Gao, and Q. Rao, "Defense against adversarial attacks by reconstructing images," *IEEE Transactions on Image Processing*, vol. 30, pp. 6117–6129, 2021.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[8] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7472–7482.

[9] A. Liu, X. Liu, H. Yu, C. Zhang, Q. Liu, and D. Tao, "Training robust deep neural networks via adversarial noise propagation," *IEEE Transactions on Image Processing*, vol. 30, pp. 5769–5781, 2021.

[10] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *International Conference on Learning Representations*, 2019.

[11] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8093–8104.

[12] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BJx040EFvH

[13] 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.

[14] M. Andriushchenko and N. Flammarion, "Understanding and improving fast adversarial training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 048–16 059, 2020.

[15] H. Kim, W. Lee, and J. Lee, "Understanding catastrophic overfitting in single-step adversarial training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 8119–8127.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[18] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2206–2216.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[20] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *European Conference on Computer Vision*. Springer, 2020, pp. 484–501.

[21] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2196–2205.

[22] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJUYGxbCW

[23] Y. Balaji, T. Goldstein, and J. Hoffman, "Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets," *arXiv preprint arXiv:1910.08051*, 2019.

[24] G. Sriramanan, S. Addepalli, A. Baburaj *et al.*, "Guided adversarial attack for evaluating and enhancing adversarial defenses," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 297–20 308, 2020.

[25] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[26] T. Roughgarden, "Algorithmic game theory," *Communications of the ACM*, vol. 53, no. 7, pp. 78–86, 2010.

[27] A. Mokhtari, A. Ozdaglar, and S. Pattathil, "A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 1497–1507.

[28] T. Lin, C. Jin, and M. Jordan, "On gradient descent ascent for nonconvex-concave minimax problems," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6083–6093.

[29] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5286–5295.

[30] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] T. Li, Y. Wu, S. Chen, K. Fang, and X. Huang, "Subspace adversarial training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 409–13 418.

[33] X. Jia, Y. Zhang, B. Wu, J. Wang, and X. Cao, "Boosting fast adversarial training with learnable adversarial initialization," *IEEE Transactions on Image Processing*, vol. 31, pp. 4417–4430, 2022.

[34] G. Y. Park and S. W. Lee, "Reliably fast adversarial training via latent adversarial perturbation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7758–7767.

[35] X. Jia, Y. Zhang, X. Wei, B. Wu, K. Ma, J. Wang, and X. Cao, "Prior-guided adversarial initialization for fast adversarial training," in *European Conference on Computer Vision*. Springer, 2022, pp. 567–584.

[36] G. Sriramanan, S. Addepalli, A. Baburaj *et al.*, "Towards efficient and effective adversarial training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 821–11 833, 2021.

[37] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[38] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude." *COURSERA: Neural Networks for Machine Learning*, 2012.

[39] B. Fang and D. Klabjan, "Convergence analyses of online adam algorithm in convex setting and two-layer relu neural network," *arXiv preprint arXiv:1905.09356*, 2019.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[41] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.

[42] M. Cheng, Q. Lei, P.-Y. Chen, I. Dhillon, and C.-J. Hsieh, "Cat: Customized adversarial training for improved robustness," in *IJCAI*, 2022.

[43] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "Mma training: Direct input space margin maximization through adversarial training," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HkeryxBtPB

[44] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *International conference on machine learning*. PMLR, 2021, pp. 11 492–11 501.

[45] C. Liu, Z. Huang, M. Salzmann, T. Zhang, and S. Süsstrunk, "On the impact of hard adversarial instances on overfitting in adversarial training," *arXiv preprint arXiv:2112.07324*, 2021.

[46] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds. BMVA Press, September 2016, pp. 87.1–87.12. [Online]. Available: https://dx.doi.org/10.5244/C.30.87

[47] Z. Huang, Y. Huang, and T. Zhang, "Corrattack: Black-box adversarial attack with structured search," *arXiv preprint arXiv:2010.01250*, 2020.

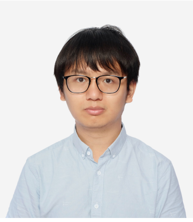
[48] A. Ilyas, L. Engstrom, and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," in *International Conference on Learning Representations*, 2018.

Zhichao Huang received the B.Sc. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2018, and the Ph.D. degrees from the Department of Mathematics, The Hong Kong University of Science of Technology, Hong Kong, China in 2022. The work was done during his internship at Tencent AI Lab. He currently works as a Research Scientist in Bytedance AI Lab. His research interests include machine learning and computer vision with a particular focus on adversarial robustness.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3326398

Yanbo Fan is currently a Senior Researcher at Tencent AI Lab. He received his Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2018, and his B.S. degree in Computer Science and Technology from Hunan University in 2013. His research interests are computer vision and machine learning.

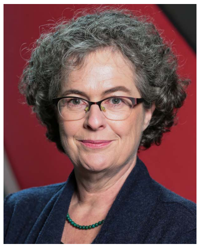


Chen Liu (Member, IEEE) received his bachelor's degree in computer science from Tsinghua University, Beijing, China, in 2015, and the master's and Ph.D degrees in computer science from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2017 and 2022, respectively. He is currently an Assistant Professor at the City University of Hong Kong. He received the Microsoft Research Scholarship from 2017 to 2019. He is working on algorithms to build robust deep learning models that are certifiable, scalable, and efficient. His research focus is on machine learning and optimization, especially deep learning.

Weizhong Zhang is a tenure-track professor at School of Data Science, Fudan University. He received the B.S. and Ph.D. degrees both from Zhejiang University in 2012 and 2017, respectively. His research interests include sparse neural network training, robustness and out-of-distribution generalization.

Yong Zhang received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2018. From 2015 to 2017, he was a Visiting Student with the Rensselaer Polytechnic Institute. He is currently with the Tencent AI Lab. His research interests include computer vision, machine learning, and probabilistic graphical models.

Mathieu Salzmann (Member, IEEE) received the Ph.D. degree from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2009. He was a Senior Researcher and a Research Leader with the NICTA's Computer Vision Research Group, Canberra, Australia, a Research Assistant Professor with TTI-Chicago, Chicago, IL, USA, and a Post-Doctoral Fellow with ICSI and EECS, UC Berkeley, Berkeley, CA, USA. He is a Senior Researcher with EPFL, and an Artificial Intelligence Engineer with ClearSpace, Renens, Switzerland. His research interests lie at the intersection of machine learning and computer vision.



Sabine Süsstrunk (Fellow, IEEE) was the first Director of the École polytechnique fédérale de Lausanne (EPFL)'s, Digital Humanities Institute, Lausanne, Switzerland, from 2015 to 2020. She is a Full Professor with EPFL and the Director of the Image and Visual Representation Laboratory (IVRL), School of Computer and Communication Sciences (IC). Her main research areas are computational photography, computational imaging, color image processing and computer vision, machine learning, and computational image quality and esthetics.

Prof. Süsstrunk is a fellow of IS&T and a Full Member of the Swiss Academy of Engineering Sciences. She is the President of the Swiss Science Council (SSC), a Founding Member and a member of the Board (President from 2014 to 2018) of the EPFL-WISH (Women in Science and Humanities) Foundation, a member of the Board of the SRG SSR (Swiss Radio and Television Corporation), and a Co-Founder and a member of the Board of Largo Films. She was a recipient of the IS&T/SPIE 2013 Electronic Imaging Scientist of the Year Award for her contributions to color imaging, computational photography, and image quality, and the 2018 IS&T Raymond C. Bowman and the 2020 EPFL AGEPoly IC Polysphere Awards for excellence in teaching.

Jue Wang (SM'12) is Distinguished Scientist at Tencent, leading research efforts in intelligent content generation for the gaming and entertainment industry. He is the Director of the Visual Computing Center at Tencent AI lab, a multidisciplinary research hub for Computer Vision, Computer Graphics and HCI. He received his BE and MS from Tsinghua University in Beijing, and his PhD from the University of Washington at Seattle. He was Senior Director at MEGVII Research from 2017 to 2020, and was Principal Scientist at Adobe Research before that. He has published more than 150 research articles in top-tier academic journals and conferences in the areas of Computer Vision, Computer Graphics, Machine Learning and HCI, and holds more than 80 international patents. He is an Associated Editor for IEEE Transactions on Pattern Analysis and Machine Intelligence. Dr. Wang has a respectable record of transferring advanced technologies into consumer products. His early work in grad school on Image Matting has been transferred into a commercial product called Silhouette that won 2019 Oscar technical award. At Adobe he has developed a dozen of release-defining features for Photoshop and After Effects such as Refine Edge, Shake Reduction, Warp Stabilizer and RotoBrush, etc. At MEGVII he led the development of many cutting-edge mobile photography solutions that have been shipped in tens of millions Android smartphones.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on November 05,2023 at 05:45:52 UTC from IEEE Xplore. Restrictions apply.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.