Adversarial Initialization with Universal Adversarial Perturbation: A New Approach to Fast Adversarial Training

Chao Pan[1,2,3], **Qing Li**[3], **Xin Yao**[1,2*]
[1]Research Institute of Trustworthy Autonomous Systems,
Southern University of Science and Technology, Shenzhen 518055, China
[2]Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China
[3]The Hong Kong Polytechnic University, Hong Kong, China
11930665@mail.sustech.edu.cn, csqli@comp.polyu.edu.hk, xiny@sustech.edu.cn

Abstract

Traditional adversarial training, while effective at improving machine learning model robustness, is computationally intensive. Fast Adversarial Training (FAT) addresses this by using a single-step attack to generate adversarial examples more efficiently. Nonetheless, FAT is susceptible to a phenomenon known as catastrophic overfitting, wherein the model's adversarial robustness abruptly collapses to zero during the training phase. To address this challenge, recent studies have suggested adopting adversarial initialization with Fast Gradient Sign Method Adversarial Training (FGSM-AT), which recycles adversarial perturbations from prior epochs by computing gradient momentum. However, our research has uncovered a flaw in this approach. Given that data augmentation is employed during the training phase, the samples in each epoch are not identical. Consequently, the method essentially yields not the adversarial perturbation of a singular sample, but rather the Universal Adversarial Perturbation (UAP) of a sample and its data augmentation. This insight has led us to explore the potential of using UAPs for adversarial initialization within the context of FGSM-AT. We have devised various strategies for adversarial initialization utilizing UAPs, including single, class-based, and feature-based UAPs. Experiments conducted on three distinct datasets demonstrate that our method achieves an improved trade-off among robustness, computational cost, and memory footprint. Code is available at https://github.com/fzjcdt/fgsm-uap.

Introduction

Deep neural networks (DNNs) exhibit significant vulnerabilities in the face of adversarial examples, where meticulously crafted inputs can lead to erroneous model predictions (Szegedy et al. 2013). This vulnerability is not only counterintuitive, given the high performance of DNNs on standard datasets, but it also raises serious concerns about the security and reliability of these models in real-world applications.

To mitigate this issue, a common approach is to use adversarial training, a robust optimization technique that augments the training data with Adversarial Examples (AEs) (Madry et al. 2017; Rice, Wong, and Kolter 2020a,b; Rebuffi et al. 2021; Gowal et al. 2021). Notably, the Projected Gradient Descent (PGD) (Madry et al. 2017) method is often employed to generate these adversarial examples. However, this standard adversarial training comes with a significant cost, making it substantially more computationally expensive than ordinary training. This high cost renders the approach impractical for large-scale datasets, posing a substantial challenge to the widespread application of adversarial training.

In the pursuit of augmenting efficiency, the implementation of Fast Adversarial Training (FAT) has been proposed (Wong, Rice, and Kolter 2020; Sriramanan et al. 2021; Andriushchenko and Flammarion 2020; de Jorge Aranda et al. 2022; Zhang et al. 2019). FAT is typically characterized by the utilization of single-step attack methodologies, such as the Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015), to generate adversarial examples for training purposes. Nevertheless, it has been observed that conventional FGSM-AT (Li et al. 2022) is prone to catastrophic overfitting (Sriramanan et al. 2020; Kim, Lee, and Lee 2021). This phenomenon is characterized by a sudden plummet in the model's adversarial robustness to zero during the training phase. This predicament poses a substantial obstacle in the quest for efficacious and efficient adversarial training methodologies.

The phenomenon of catastrophic overfitting has been the subject of numerous hypotheses (Kim, Lee, and Lee 2021; Sriramanan et al. 2021; Li et al. 2020; He et al. 2023). A widely accepted perspective posits that after several iterations of training, the success rate of FGSM attacks significantly decreases (Jia et al. 2022). This implies that the quality of the generated Adversarial Examples (AEs) is poor, and they are insufficiently robust to resolve the maximum problem inherent in the adversarial training formula.

In response to this challenge, researchers have proposed the use of adversarial initialization strategies to enhance the quality of AEs (Jia et al. 2022, 2023). These strategies involve reusing perturbations from previous epochs. However, these approaches come with their own set of challenges. A notable drawback is the requirement to store the complete perturbation information from earlier iterations. This storage requirement becomes particularly problematic when dealing with large datasets, rendering these strategies impractical in such contexts.

*Corresponding Author.

In our exploration, we have observed that due to the use of data augmentation during training, the same sample differs at each iteration. As a result, the historical perturbations used in the current epoch are not the adversarial perturbations of the same sample. This realization uncovers a misconception held by previous researchers about this strategy: in reality, what is being utilized is not the historical adversarial perturbation of a specific sample, but a Universal Adversarial Perturbation (UAP) (Moosavi-Dezfooli et al. 2017) applicable to a sample and its data augmentations.

This key insight inspired us to use UAPs for adversarial initialization in fast adversarial training. To further reduce the memory demand of adversarial initialization, we have designed strategies that allow training samples with high similarity to utilize the same UAPs. This approach not only addresses the memory storage issue associated with storing individual perturbations, but also leverages the shared features among similar samples to enhance the robustness of the adversarial training.

The contributions of this paper are as follows:

- We reveal that the adversarial initialization used in fast adversarial training methods such as FGSM-PGI is essentially a form of universal adversarial perturbations. This understanding provides a fresh perspective on the nature of adversarial initialization and sets the stage for our proposed strategies.
- We propose three novel UAP-based adversarial initialization strategies for FAT. Experimental results on three datasets demonstrate that these strategies not only reduce memory usage but also enhance the robustness of adversarial training by exploiting shared features among similar samples. This represents a significant step forward in the efficiency and effectiveness of adversarial training.
- We conducted a comprehensive analysis of the sensitivity to hyperparameters in FGSM-UAP, and corroborated the robust performance of our proposed method across various settings and configurations.

Related Work

Adversarial Attacks

Deep neural networks, despite their exceptional performance in various tasks, are susceptible to deception by adversarial attacks (Goodfellow, Shlens, and Szegedy 2015; Szegedy et al. 2013; Madry et al. 2017). These attacks craft adversarial examples by adding small perturbations to the input data that can mislead the model into making false predictions. There are two types of adversarial attacks: image-dependent and image-agnostic (universal) attacks.

Image-dependent attacks craft adversarial perturbations for each specific input. The perturbation for an input image $\boldsymbol{x}$ can be represented as $\boldsymbol{\delta}^*_{\boldsymbol{x}}$, which is generated by maximizing the loss function $\mathcal{L}$ over a set $\boldsymbol{S}$ of allowed perturbations:

$$\boldsymbol{\delta}^*_{\boldsymbol{x}} = \arg\max_{\boldsymbol{\delta}\in\boldsymbol{S}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y). \tag{1}$$

On the other hand, image-agnostic or universal attacks create a universal perturbation $\boldsymbol{\delta}^*$ that can mislead the model for a wide range of inputs (Moosavi-Dezfooli et al. 2017). The universal perturbation is obtained by solving:

$$\boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\delta}\in\boldsymbol{S}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y)]. \tag{2}$$

Adversarial Training

Adversarial training (AT) is a defensive technique that aims to improve the model's robustness against adversarial attacks (Madry et al. 2017). It involves training the model on adversarial examples. The adversarial training process can be represented as the following min-max problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\max_{\boldsymbol{\delta}\in\boldsymbol{S}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y)]. \tag{3}$$

To solve the inner maximization problem, we usually use PGD attack (Madry et al. 2017). The PGD attack iteratively applies the gradient ascent on the loss function and projects the perturbed example back into the feasible set S:

$$\boldsymbol{\delta}^{\boldsymbol{t+1}} = \Pi_S(\boldsymbol{\delta}^{\boldsymbol{t}} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{\delta}}\mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}^{\boldsymbol{t}};\boldsymbol{\theta}), y))). \tag{4}$$

Fast Adversarial Training

While the multi-step AT method can achieve good robustness, the computational cost is high. Fast adversarial training (FAT) aims to generate adversarial samples using one-step adversarial attack for training, significantly reducing the computational cost.

FAT usually consists of two steps. First, it uses Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015) or its variants to generate adversarial samples:

$$\boldsymbol{\delta}_{\boldsymbol{x}} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}}\mathcal{L}(f(\boldsymbol{x};\boldsymbol{\theta}^{\boldsymbol{t}}), y)). \tag{5}$$

Next, it uses these generated adversarial samples to train the neural network:

$$\boldsymbol{\theta}^{\boldsymbol{t+1}} = \boldsymbol{\theta}^{\boldsymbol{t}} - \eta \cdot \nabla_{\boldsymbol{\theta}}\mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}_{\boldsymbol{x}};\boldsymbol{\theta}^{\boldsymbol{t}}), y). \tag{6}$$

Based on their approach, FAT methods can be categorized into two groups: those that utilize specialized initialization techniques to generate more potent adversarial samples for FGSM attacks (Wong, Rice, and Kolter 2020; de Jorge Aranda et al. 2022; Jia et al. 2022, 2023), and those that incorporate regularization terms during neural network training to address the maximization problem inherent in adversarial training with FGSM-generated samples (Andriushchenko and Flammarion 2020; Sriramanan et al. 2021, 2020; Jia et al. 2022).

Proposed Method

In this section, we initially reveal that the Fast Gradient Sign Method with Prior-Guided Initialization (FGSM-PGI) approach (Jia et al. 2022, 2023), which employs historically-generated AEs as suggested by prior researchers, fundamentally leverages a UAP. Enlightened by this realization, we harness this concept to formulate three distinct strategies for adversarial initialization utilizing UAPs. These strategies, specifically, single UAP, class-based UAPs, and feature-based UAPs, are designed to mitigate the risk of catastrophic overfitting.

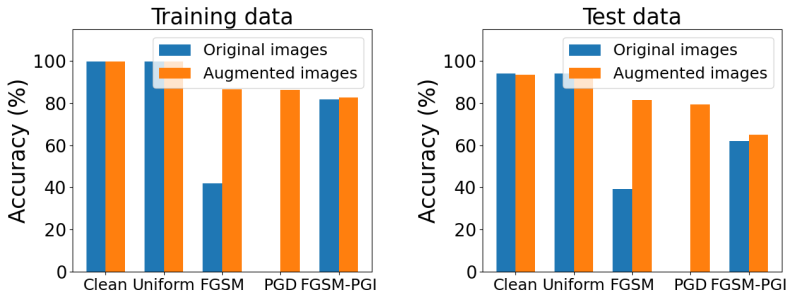

Figure 1: Comparison of the accuracy of original and augmented images under different perturbations.

Revisiting Prior-Guided Adversarial Initialization

(Jia et al. 2022) unveiled the potential of adversarial initialization in augmenting the quality of adversarial examples, and subsequently introduced a novel method known as prior-guided initialization, which employs previously generated adversarial examples, succinctly referred to as FGSM-PGI. This innovative approach successfully mitigates the issue of catastrophic overfitting without necessitating any additional computational resources. After meticulous exploration of various initialization strategies, they found that the most optimal results were obtained through the utilization of a momentum mechanism which leverages the buffered gradients accumulated from all preceding epochs.

The adversarial perturbation in this methodology can be mathematically expressed as follows:

$$\boldsymbol{g_x} = \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta_x^t}; \boldsymbol{\theta}), y)), \tag{7}$$

where $\boldsymbol{g_x}$ denotes the signed gradient of the sample x.

$$\boldsymbol{m_x^{t+1}} = \mu \cdot \boldsymbol{m_x^t} + \boldsymbol{g_x}, \tag{8}$$

where $\boldsymbol{m_x^{t+1}}$ represents the signed gradient momentum of the sample $\boldsymbol{x}$ at the $(t+1)$th epoch, and μ is the decay factor.

$$\boldsymbol{\delta_x^{t+1}} = \Pi_{\boldsymbol{S}}(\boldsymbol{\delta_x^t} + \alpha \cdot \text{sign}(\boldsymbol{m_x^{t+1}})), \tag{9}$$

where $\boldsymbol{\delta_x^{t+1}}$ signifies the adversarial perturbation of the sample $\boldsymbol{x}$ at the $(t+1)$th epoch, with its initialization being the gradient momentum which encapsulates information from all previous epochs.

Nonetheless, our investigation has unveiled certain deficiencies in comprehending the formulae related to FGSM-PGI, as expounded in previous research. Specifically, in the context of Equation 8, the term $\boldsymbol{m_x^{t+1}}$ is inaccurately characterized as the signed gradient momentum of the sample $\boldsymbol{x}$ at the $(t+1)$th epoch. This discrepancy arises due to the incorporation of data augmentation, which leads to the sample $\boldsymbol{x}$ undergoing variation across each epoch during the training phase.

Further, in relation to Equation 7, even a minor offset of the data-augmented $\boldsymbol{x}$ by a single pixel results in $\boldsymbol{g_x}$ deviating from the gradient sign of the original sample $\boldsymbol{x}$. Consequently, the entity that Equation 8 accumulates via the momentum mechanism is not the signed gradient of an identical sample across epochs, but rather the signed gradient of a sample $\boldsymbol{x}$ in conjunction with its data-augmented counterpart. The perturbation $\boldsymbol{\delta_x^{t+1}}$ possesses the capacity to not only attack the original sample $\boldsymbol{x}$, but also its data-augmented counterpart.

To substantiate the claim that $\boldsymbol{\delta_x^{t+1}}$ is a UAP with regard to the sample $\boldsymbol{x}$ and its data augmentations, we conducted a simple experiment. A ResNet-18 model was trained on the CIFAR-10 dataset, during which we computed the adversarial initialization perturbations for each sample according to Equations 7 through 9. In order to investigate whether $\boldsymbol{\delta_x^{t+1}}$ also serves as a UAP for the test sample $\boldsymbol{x}$ and its corresponding data augmentations, we calculated the adversarial initialization perturbations for these test samples, deliberately excluding the test data from the training phase.

Once the model was trained, we evaluated its accuracy on the original samples as well as their augmented versions. Adversarial perturbations were generated for the original samples using two image-dependent adversarial attack methods, namely the FGSM and PGD, and were applied to the corresponding augmented data. Lastly, we incorporated the perturbation $\boldsymbol{\delta_x^{t+1}}$ into the original and augmented data and reevaluated the model's accuracy.

As demonstrated in Figure 1, the model exhibits a high level of accuracy for both the training and testing datasets, irrespective of whether the samples are in their original or augmented form. The addition of uniform noise to the training and testing data, which is not characterized as an adversarial perturbation, imparts negligible effects on the model's performance. The FGSM and PGD both exhibit high attack success rates when applied to original samples. However, when the adversarial perturbations derived from these original samples are transferred to the augmented counterparts, the attack success rate diminishes significantly. This indicates that the adversarial perturbations generated by FGSM and PGD are inherently dependent on the image.

In contrast, the adversarial initialization perturbation instigated by FGSM-PGI exhibits a comparable attack success rate on both original and augmented samples, surpassing the success rate of FGSM and PGD when transferred to augmented samples. This substantiates the claim that the adversarial initialization perturbation $\boldsymbol{\delta_x^{t+1}}$, as generated by FGSM-PGI, essentially functions as a UAP for the sample $\boldsymbol{x}$ and its corresponding data augmentations.

Utilizing UAPs for Adversarial Initialization

The application of FGSM-PGI in generating adversarial initialization perturbations has proven to be an effective strategy against the problem of catastrophic overfitting in fast adversarial training (Jia et al. 2022). Nonetheless, this method is not without its drawbacks, as it requires the retention of historical gradient information for each sample. This requirement consequently leads to a significant increase in the usage of memory, presenting a considerable challenge when dealing with large-scale datasets.

In our preceding section, we have discussed how the adversarial initialization perturbation $\boldsymbol{\delta}_{\boldsymbol{x}}^{\boldsymbol{t+1}}$ employed by FGSM-PGI essentially acts as a UAP for the sample $\boldsymbol{x}$ and its corresponding data augmentation. This realization has prompted us to consider the use of UAPs for the purpose of adversarial initialization perturbation. FGSM-PGI, in its current form, necessitates the storage of a UAP for each sample and its data augmentation. In an attempt to mitigate this issue, we suggest a shift towards the use of more universally applicable UAPs.

In the context of fast adversarial training, we have investigated three distinct strategies: the utilization of a single UAP, class-based UAPs, and feature-based UAPs. These strategies are designed with the intention of allowing samples exhibiting high degrees of similarity to employ the same UAP. Not only do these strategies considerably lessen the demand on GPU memory, but they also enhance the robustness of adversarial training by capitalizing on shared features among similar samples.

Single UAP The proposed approach leverages a single UAP, denoted as $\boldsymbol{\delta}$, to serve as an adversarial initialization for fast adversarial training. This $\boldsymbol{\delta}$ is sustained throughout the training process, demonstrating a consistent level of attack effectiveness on all training instances. More specifically, for a particular instance $\boldsymbol{x}$, we introduce $\boldsymbol{\delta}$ to $\boldsymbol{x}$ and utilize the FGSM to generate adversarial examples for training the model. The expression for this adversarial initialization perturbation is as follows:

$$\boldsymbol{g} = \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta}), y)), \tag{10}$$

where $\boldsymbol{g}$ represents the sign of the gradient of the loss function $\mathcal{L}$ with respect to the input $\boldsymbol{x}$, f symbolizes the model parameterized by $\boldsymbol{\theta}$, and y denotes the true label.

We update the momentum of the signed gradient for all training instances, represented as $\boldsymbol{m}$, in accordance with the subsequent equation:

$$\boldsymbol{m} = \mu \cdot \boldsymbol{m} + \boldsymbol{g}, \tag{11}$$

where μ is the coefficient of momentum.

As the final step, we update the UAP, $\boldsymbol{\delta}$, as follows:

$$\boldsymbol{\delta} = \Pi_{\boldsymbol{S}}(\boldsymbol{\delta} + \alpha \cdot \text{sign}(\boldsymbol{m})), \tag{12}$$

where $\Pi_{\boldsymbol{S}}$ is the projection operation onto the set $\boldsymbol{S}$, α signifies the step size, and $\text{sign}(\boldsymbol{m})$ is the sign of the momentum.

Algorithm 1: Feature-based FGSM-UAP

Input: A classifier f_θ with loss function $\mathcal{L}$; A two-layer MLP h_{θ_h}; A sample $\boldsymbol{x}$ and label y; Allowed perturbation set $\boldsymbol{S}$; UAPs $\boldsymbol{\delta}$; The momentum of the signed gradient $\boldsymbol{m}$.

$\boldsymbol{o_x} = f(\boldsymbol{x}; \boldsymbol{\theta}^{'})$ {Compute penultimate layer feature}
$\boldsymbol{h_1}, \boldsymbol{h_2} = h(\boldsymbol{o_x}; \boldsymbol{\theta_h})$ {Determine scaled feature and outputs}
$\boldsymbol{\theta_h} = \boldsymbol{\theta_h} - \eta \cdot \nabla_{\boldsymbol{\theta_h}} \mathcal{L}(\boldsymbol{h_2}, y)$ {Update MLP h_{θ_h}}
$i = \arg\max(\boldsymbol{h_1})$ {Select i-th UAP}
$\hat{\boldsymbol{x}} = \Pi_S(\boldsymbol{x} + \boldsymbol{\delta_i})$ {Initialize adversarially}
$\boldsymbol{\delta_{adv}} = \boldsymbol{g} = \Pi_S(\text{ sign}(\nabla_{\hat{\boldsymbol{x}}} \mathcal{L}(f(\hat{\boldsymbol{x}}; \boldsymbol{\theta}), y)))$ {Compute adversarial perturbation}
$\boldsymbol{m_i} = \mu \cdot \boldsymbol{m_i} + \boldsymbol{g}$ {Update momentum of signed gradient}
$\boldsymbol{\delta_i} = \Pi_S(\boldsymbol{\delta_i} + \alpha \cdot \text{sign}(\boldsymbol{m_i}))$ {Update UAPs}
$\boldsymbol{x_{adv}} = \Pi_S(\boldsymbol{x} + \boldsymbol{\delta_{adv}})$ {Generate adversarial example}
$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}} (\mathcal{L}(f(\boldsymbol{x_{adv}}; \boldsymbol{\theta}), y) + \lambda \cdot ||f(\boldsymbol{x_{adv}}; \boldsymbol{\theta}) - f(\hat{\boldsymbol{x}}; \boldsymbol{\theta})||_2^2$ {Perform adversarial training}

Class-based UAPs In this approach, we incorporate a consistent adversarial initialization for training instances belonging to the same class throughout the learning procedure. The number of UAPs is kept equivalent to the total number of classes. Given the shared similarity in data properties within the same class, there is a higher likelihood for successful attacks by the same type of adversarial initialization. The class-based adversarial initialization perturbation, symbolized as $\boldsymbol{\delta_y}$, can be formulated as:

$$\boldsymbol{g} = \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta_y}; \boldsymbol{\theta}), y)). \tag{13}$$

In this equation, $\boldsymbol{\delta_y}$ signifies the adversarial initialization for the y-th class.

The momentum of the signed gradient for the y-th class, represented as $\boldsymbol{m_y}$, is updated according to:

$$\boldsymbol{m_y} = \mu \cdot \boldsymbol{m_y} + \boldsymbol{g}. \tag{14}$$

Finally, the adversarial initialization perturbation $\boldsymbol{\delta_y}$ is updated by the following equation:

$$\boldsymbol{\delta_y} = \Pi_{\boldsymbol{S}}(\boldsymbol{\delta_y} + \alpha \cdot \text{sign}(\boldsymbol{m_y})). \tag{15}$$

Feature-based UAPs Both single UAP and class-based UAPs for adversarial initialization fix the number of UAPs, which is not conducive to enhancing the effect of adversarial initialization by exploiting shared features among similar samples. Hence, we propose a more flexible feature-based UAPs adversarial initialization method.

Firstly, we extract the features from the penultimate layer of the neural network model for the sample $\boldsymbol{x}$,

$$\boldsymbol{o_x} = f(\boldsymbol{x}; \boldsymbol{\theta}^{'}), \tag{16}$$

where $\boldsymbol{\theta}^{'}$ represents the parameters of the neural network model excluding the final layer.

Subsequently, we employ a two-layer Multilayer Perceptron (MLP) h to scale down the number of features. The output of the first layer $h_1(\boldsymbol{o_x}; \boldsymbol{\theta_h})$ is the scaled feature, which is used for UAPs selection, while the output of the second layer $h_2(\boldsymbol{o_x}; \boldsymbol{\theta_h})$ is used for training the MLP. We select the index i of the feature with the highest activation value for the adversarial initialization of UAPs,

$$i = \arg\max h_1(\boldsymbol{o_x}; \boldsymbol{\theta_h}). \tag{17}$$

Next, we use the selected UAP $\boldsymbol{\delta_i}$ to perform adversarial initialization, and then update the UAPs.

$$\boldsymbol{g} = \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta_i}; \boldsymbol{\theta}), y)), \tag{18}$$

$$\boldsymbol{m_i} = \mu \cdot \boldsymbol{m_i} + \boldsymbol{g}. \tag{19}$$

Finally, the adversarial perturbation $\boldsymbol{\delta_i}$ is updated by the following equation:

$$\boldsymbol{\delta_i} = \Pi_{\boldsymbol{S}}(\boldsymbol{\delta_i} + \alpha \cdot \text{sign}(\boldsymbol{m_i})). \tag{20}$$

In this way, we leverage feature-based UAPs to enable more flexible adversarial initialization, thereby enhancing the robustness of our model against adversarial attacks. The methodology encompassed by the feature-based FGSM-UAP is succinctly encapsulated within Algorithm 1.

Experiments

In this section, we clarify the dataset used, outline the experimental framework, and offer a detailed analysis of the results garnered from the experiment.

Experiment Setup

Experiments were executed on three datasets that are prevalently employed for the evaluation of adversarial robustness, specifically CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), and Tiny ImageNet (Deng et al. 2009). In an effort to uphold impartiality across comparisons, the experimental configurations outlined in the study by (Jia et al. 2022) were adopted. This involved the utilization of ResNet-18 (He et al. 2016a) as the backbone architecture for CIFAR-10 and CIFAR-100, and PreActResNet18 (He et al. 2016b) for Tiny ImageNet.

The Stochastic Gradient Descent (SGD) optimizer was employed, configured with a learning rate of 0.1, a weight decay of 5e-4, and a momentum setting of 0.9. The total number of training iterations was fixed at 110. Within the CIFAR-10 and CIFAR-100 datasets, the learning rate was attenuated by a factor of 0.1 at the 100th and 105th epochs. Conversely, in the Tiny ImageNet dataset, the learning rate was similarly reduced at the 105th epoch. Regarding hyperparameters, the decay coefficient, represented by the symbol μ, has been set to 0.3, and a weighted average is employed with a τ value of 0.9995. We employed the same regularization term as in FGSM-PGI and designated the value of λ to be 10. Within the feature-based UAPs methodology, the quantity of UAPs utilized in CIFAR-10, CIFAR-100, and Tiny ImageNet were fixed at 50, 200, and 300, respectively.

The robustness of the proposed model is assessed through the application of PGD (Madry et al. 2017), C&W (Carlini and Wagner 2017), and AutoAttack (Croce and Hein 2020b). AutoAttack, in particular, is an amalgamation of four prominent attack methodologies frequently employed in the evaluation of model robustness (Croce and Hein 2020a; Andriushchenko et al. 2020; Croce and Hein 2020b). The maximum iterations for PGD attacks are designated as 10, 20, and 50, and are correspondingly referred to as PGD-10, PGD-20, and PGD-50. The maximum perturbation intensity is calibrated to a value of 8/255. In alignment with the parameters set forth by (Jia et al. 2022), we present the results of the model that demonstrates the highest robust accuracy during the training process under the conditions of PGD-10, as well as the model derived from the terminal epoch.

Results and Analysis

The average results replicated three times on the CIFAR-10, CIFAR-100, and Tiny ImageNet datasets are respectively delineated in Table 1, Table 2, and Table 3.

This improvement is primarily attributable to the incorporation of the feature-based FGSM-UAP, a mechanism that judiciously selects the optimal UAP for adversarial initialization, contingent upon the characteristics of the input samples. Our approach demonstrates enhanced robust accuracy in the face of PGD assaults on different datasets, relative to other FAT methodologies such as FGSM-PGI and NuAT. Even when compared to the multi-step adversarial training method PGD-AT, our approach maintains its superior performance. Moreover, the AutoAttack robust accuracy of the feature-based FGSM-UAP surpasses other FAT methodologies, irrespective of whether the model is evaluated at its optimal or final epoch.

Moreover, the methodology we propose necessitates a lesser data memory footprint in comparison to alternative adversarial initialization techniques, including the FGSM-PGI. The adversarial initialization that employs a singular UAP mandates auxiliary storage for a single instance only. The class-based UAPs adversarial initialization requires additional storage commensurate with the quantity of classes, while the feature-based UAPs adversarial initialization necessitates storage for a predetermined number of instances. These storage demands are substantially inferior to those associated with the complete training dataset, thereby augmenting the adaptability of our method in relation to FGSM-PGI.

Given the utilization of distinct computational devices in our experiment and other methods, a direct comparison of execution times is not practically viable. However, it becomes evident that single UAP, class-based UAPs for adversarial initialization, and FGSM-PGI exhibit identical computational complexities. When applied to the CIFAR-10 dataset, these methods demonstrate a superior speed, being approximately 1.4 times faster than NuAT and GAT, 2.5 times faster than FGSM-GA, and 3.6 times faster than PGD-AT. Our feature-based UAPs for adversarial initialization necessitates an additional forward propagation step. However, as it obviates the need for backpropagation to compute the model's gradient, it imposes only a negligible increment in

Method	Model	Clean	PGD10	PGD20	PGD50	C&W	AA	Data memory
PGD-AT	Best	82.32	53.76	52.83	52.60	51.08	48.68	One batch
(Madry et al. 2017)	Last	82.65	53.39	52.52	52.27	51.28	48.93	
FGSM-RS	Best	76.05	52.19	51.78	51.75	48.07	46.14	One batch
(Wong, Rice, and Kolter 2020)	Last	88.72	3.81	2.19	0.78	0.21	0.00	
FGSM-CKPT	Best	**83.89**	51.38	50.68	50.55	41.66	38.58	One batch
(Kim, Lee, and Lee 2021)	Last	**89.30**	49.07	46.99	46.57	39.72	36.98	
NuAT	Best	81.58	53.96	52.90	52.61	**51.30**	49.09	One batch
(Sriramanan et al. 2021)	Last	81.38	53.52	52.65	52.48	50.63	48.70	
GAT	Best	80.02	54.87	54.40	54.34	49.65	48.75	One batch
(Sriramanan et al. 2020)	Last	81.12	54.78	54.40	54.22	50.10	48.99	
FGSM-GA	Best	80.63	54.08	53.33	53.14	50.67	48.47	One batch
(Andriushchenko and Flammarion 2020)	Last	82.60	53.45	52.56	52.40	50.40	47.96	
Free-AT(m=8)	Best	80.38	47.10	45.85	45.62	44.42	42.17	One batch
(Shafahi et al. 2019)	Last	80.75	45.82	44.82	44.48	43.73	41.17	
FGSM-PGI	Best	79.30	55.76	55.16	55.07	50.81	48.97	Whole dataset
(Jia et al. 2022)	Last	79.91	55.73	55.14	55.04	50.73	48.76	
Single UAP	Best	79.28	**56.11**	55.50	55.46	50.79	48.67	One batch +
(Ours)	Last	79.64	**56.11**	55.50	**55.44**	50.73	48.45	1 sample
Class-based UAPs	Best	79.76	56.05	55.47	55.38	50.55	48.77	One batch +
(Ours)	Last	80.14	56.03	55.40	55.32	50.63	48.74	10 samples
Feature-based UAPs	Best	79.50	**56.11**	**55.63**	**55.57**	50.86	**49.18**	One batch +
(Ours)	Last	80.18	56.02	**55.53**	55.43	**50.94**	**49.19**	50 samples

Table 1: Comparative analysis of clean and robust accuracy (%) utilizing ResNet18 on the CIFAR-10 dataset. The bold number indicates optimal performance.

Method	Model	Clean	PGD10	PGD20	PGD50	C&W	AA	Data memory
PGD-AT	Best	57.52	29.60	28.99	28.87	28.85	25.48	One batch
(Madry et al. 2017)	Last	57.50	29.54	29.00	28.90	27.60	25.48	
FGSM-RS	Best	49.85	22.47	22.01	21.82	20.55	18.29	One batch
(Wong, Rice, and Kolter 2020)	Last	60.55	0.45	0.25	0.19	0.25	0.00	
FGSM-CKPT	Best	**69.45**	24.32	22.74	22.24	21.30	18.27	One batch
(Kim, Lee, and Lee 2021)	Last	**69.45**	24.32	22.74	22.24	21.30	18.27	
NuAT	Best	28.51	22.03	22.03	22.06	18.34	17.74	One batch
(Sriramanan et al. 2021)	Last	18.37	13.93	13.79	13.76	10.12	8.78	
GAT	Best	65.54	29.02	28.24	28.05	24.69	22.70	One batch
(Sriramanan et al. 2020)	Last	65.54	29.02	28.24	28.05	24.69	22.70	
FGSM-GA	Best	55.23	31.97	31.61	31.60	28.14	26.18	One batch
(Andriushchenko and Flammarion 2020)	Last	58.35	31.42	30.93	30.90	27.80	25.71	
Free-AT(m=8)	Best	52.49	24.07	23.52	23.36	21.66	19.47	One batch
(Shafahi et al. 2019)	Last	52.63	22.86	22.32	22.16	20.68	18.57	
FGSM-PGI	Best	54.65	31.81	31.53	31.53	28.20	26.50	Whole dataset
(Jia et al. 2022)	Last	57.92	31.15	30.80	30.74	27.91	26.13	
Single UAP	Best	56.30	32.43	32.11	32.06	28.20	26.50	One batch +
(Ours)	Last	58.59	31.90	31.53	31.40	27.66	25.98	1 sample
Class-based UAP	Best	55.97	32.36	32.01	32.00	27.96	26.48	One batch +
(Ours)	Last	57.48	31.98	31.68	31.61	27.70	26.18	100 samples
Feature-based UAP	Best	56.47	**32.50**	**32.23**	**32.17**	**28.37**	**26.63**	One batch +
(Ours)	Last	57.73	**32.29**	**31.91**	**31.87**	**27.92**	**26.24**	200 samples

Table 2: Comparative analysis of clean and robust accuracy (%) utilizing ResNet18 on the CIFAR-100 dataset. The bold number indicates optimal performance.

computational cost.

Sensitivity Analysis

We introduce an additional hyperparameter to both the single UAP and the class-based UAPs adversarial initialization techniques, specifically, the perturbation magnitude of the UAP. In a similar vein, another hyperparameter, the number of UAPs, is incorporated into the feature-based UAPs adversarial initialization. Consequently, in this section, we investigate the influence of these two hyperparameters on fast

Method	Model	Clean	PGD10	PGD20	PGD50	C&W	AA	Data memory
PGD-AT	Best	43.60	20.20	19.90	19.86	17.50	16.00	One batch
(Madry et al. 2017)	Last	45.28	16.12	15.60	15.40	14.28	12.84	
FGSM-RS	Best	40.72	23.01	22.92	22.88	18.05	16.32	One batch
(Wong, Rice, and Kolter 2020)	Last	46.58	0.49	0.34	0.24	0.09	0.02	
FGSM-CKPT	Best	**59.80**	18.15	17.20	16.91	13.44	10.82	One batch
(Kim, Lee, and Lee 2021)	Last	**60.43**	18.15	17.28	16.97	13.52	10.85	
NuAT	Best	36.82	23.26	23.26	23.24	18.29	16.85	One batch
(Sriramanan et al. 2021)	Last	55.94	17.89	15.27	13.91	10.09	3.40	
GAT	Best	56.96	18.25	17.58	17.38	12.83	10.73	One batch
(Sriramanan et al. 2020)	Last	56.96	18.25	17.58	17.38	12.83	10.74	
FGSM-GA	Best	38.96	22.60	22.46	22.46	17.74	16.05	One batch
(Andriushchenko and Flammarion 2020)	Last	37.77	0.00	0.00	0.00	0.00	0.00	
Free-AT(m=8)	Best	38.90	11.62	11.24	11.02	11.00	9.28	One batch
(Shafahi et al. 2019)	Last	40.06	8.84	8.32	8.20	8.08	7.34	
FGSM-PGI	Best	43.32	23.80	23.40	23.38	19.28	17.56	Whole dataset
(Jia et al. 2022)	Last	45.88	22.02	21.70	21.60	17.44	15.50	
Single UAP	Best	45.43	25.38	25.20	25.13	20.33	18.07	One batch +
(Ours)	Last	46.39	25.01	24.75	24.69	19.84	17.54	1 sample
Class-based UAPs	Best	45.35	25.53	25.37	25.31	20.39	18.33	One batch +
(Ours)	Last	45.35	25.53	25.37	25.31	20.39	18.33	200 samples
Feature-based UAPs	Best	45.49	**25.69**	**25.50**	**25.47**	**20.51**	**18.56**	One batch +
(Ours)	Last	45.49	**25.69**	**25.50**	**25.47**	**20.51**	**18.56**	300 samples

Table 3: Comparative analysis of clean and robust accuracy (%) utilizing PreActResNet18 on the Tiny ImageNet dataset. The bold number indicates optimal performance.

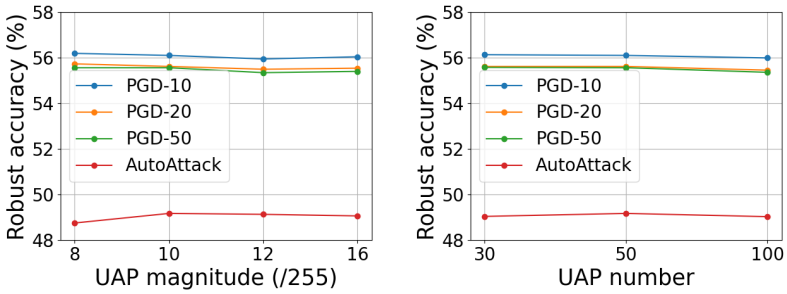

Figure 2: Sensitivity analysis of hyperparameters.

adversarial training.

We conducted experiments on the CIFAR-10 dataset. While maintaining consistency in other parameters, we varied the perturbation size of the UAP to 8/255, 10/255, 12/255, and 16/255 respectively, and set the number of UAPs to 30, 50, and 100. We then compared the robust accuracy of the model under these settings using both the PGD and AutoAttack.

The data presented in Figure 2 reveals a noticeable trend: an escalation in the perturbation size of UAP yields a marginal improvement in the model's robust accuracy, as evaluated under AutoAttack. However, this alteration appears to exert negligible influence on the robust accuracy as determined by PGD. Furthermore, when employing feature-based FGSM-UAP, the quantity of UAPs exhibits an inconsequential impact on the model's robust accuracy. This indicates that the two additional hyperparameters introduced in our proposed method exhibit strong robustness.

Conclusion

In this paper, we initially reveal that the adversarial initialization method utilizing historical information fundamentally leverages universal adversarial perturbations, which also inspires us to use UAPs for adversarial initialization. Addressing the limitation of the current best adversarial initialization method, FGSM-PGI, which requires substantial additional memory, we propose and test different strategies for adversarial initialization using UAPs, such as singular UAP, class-based UAPs, and feature-based UAPs. Our research provides a novel perspective on the use of adversarial initialization in FGSM-AT by incorporating UAPs. The results of our experiments conducted on three distinct datasets clearly indicate that these strategies not only maintain the high level of robustness accuracy offered by FGSM-AT but also significantly reduce the additional memory requirements.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62250710682), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), Research Institute of Trustworthy Autonomous Systems (RITAS), and Hong Kong Research Grants Council under the General Research Fund (Project No. 15200023).

References

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search. In *European Conference on Computer Vision*, 484–501. Springer.

Andriushchenko, M.; and Flammarion, N. 2020. Understanding and Improving Fast Adversarial Training. *Advances in Neural Information Processing Systems*, 33: 16048–16059.

Carlini, N.; and Wagner, D. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy*, 39–57. Ieee.

Croce, F.; and Hein, M. 2020a. Minimally Distorted Adversarial Examples with a Fast Adaptive Boundary Attack. In *International Conference on Machine Learning*, 2196–2205. PMLR.

Croce, F.; and Hein, M. 2020b. Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-Free Attacks. In *International Conference on Machine Learning*, 2206–2216. PMLR.

de Jorge Aranda, P.; Bibi, A.; Volpi, R.; Sanyal, A.; Torr, P.; Rogez, G.; and Dokania, P. 2022. Make Some Noise: Reliable and Efficient Single-step Adversarial Ttraining. *Advances in Neural Information Processing Systems*, 35: 12881–12893.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. Ieee.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Gowal, S.; Rebuffi, S.-A.; Wiles, O.; Stimberg, F.; Calian, D. A.; and Mann, T. A. 2021. Improving Robustness using Generated Data. *Advances in Neural Information Processing Systems*, 34: 4218–4233.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep Residual Learning for Iimage Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity Mappings in Deep Residual Networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 630–645. Springer.

He, Z.; Li, T.; Chen, S.; and Huang, X. 2023. Investigating Catastrophic Overfitting in Fast Adversarial Training: A Self-fitting Perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2313–2320.

Jia, X.; Zhang, Y.; Wei, X.; Wu, B.; Ma, K.; Wang, J.; and Cao, X. 2022. Prior-Guided Adversarial Initialization for Fast Adversarial Training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, 567–584. Springer.

Jia, X.; Zhang, Y.; Wei, X.; Wu, B.; Ma, K.; Wang, J.; and Cao Sr, X. 2023. Improving Fast Adversarial Training with Prior-Guided Knowledge. *arXiv preprint arXiv:2304.00202*.

Kim, H.; Lee, W.; and Lee, J. 2021. Understanding Catastrophic Overfitting in Single-Step Adversarial Training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8119–8127.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning Multiple Layers of Features from Tiny Images.

Li, B.; Wang, S.; Jana, S.; and Carin, L. 2020. Towards Understanding Fast Adversarial Training. *arXiv preprint arXiv:2006.03089*.

Li, Z.; Liu, L.; Wang, Z.; Zhou, Y.; and Xie, C. 2022. Bag of Tricks for FGSM Adversarial Training. *arXiv preprint arXiv:2209.02684*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv preprint arXiv:1706.06083*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal Adversarial Perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1765–1773.

Rebuffi, S.-A.; Gowal, S.; Calian, D. A.; Stimberg, F.; Wiles, O.; and Mann, T. 2021. Fixing Data Augmentation to Improve Adversarial Robustness. *arXiv preprint arXiv:2103.01946*.

Rice, L.; Wong, E.; and Kolter, Z. 2020a. Overfitting in Adversarially Robust Deep Learning. In *International Conference on Machine Learning*, 8093–8104. PMLR.

Rice, L.; Wong, E.; and Kolter, Z. 2020b. Overfitting in Adversarially Robust Deep Learning. In *International Conference on Machine Learning*, 8093–8104. PMLR.

Shafahi, A.; Najibi, M.; Ghiasi, M. A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial Training for Free! *Advances in Neural Information Processing Systems*, 32.

Sriramanan, G.; Addepalli, S.; Baburaj, A.; et al. 2020. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. *Advances in Neural Information Processing Systems*, 33: 20297–20308.

Sriramanan, G.; Addepalli, S.; Baburaj, A.; et al. 2021. Towards Efficient and Effective Adversarial Training. *Advances in Neural Information Processing Systems*, 34: 11821–11833.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199*.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast Is Better Than Free: Revisiting Adversarial Training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhang, D.; Zhang, T.; Lu, Y.; Zhu, Z.; and Dong, B. 2019. You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle. *Advances in Neural Information Processing Systems*, 32.