Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Towards sustainable adversarial training with successive perturbation generation*

Wei LIN[1,3], Lichuan LIAO‡[2]

[1]*College of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China*

[2]*College of Economics and Management, Xi'an University of Technology, Xi'an 710048, China*

[3]*Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China*

E-mail: wlin@fjut.edu.cn; liaolijuan@xaut.edu.cn

Received July 12, 2023; Revision accepted Oct. 8, 2023; Crosschecked Feb. 23, 2024

Abstract: Adversarial training with online-generated adversarial examples has achieved promising performance in defending adversarial attacks and improving robustness of convolutional neural network models. However, most existing adversarial training methods are dedicated to finding strong adversarial examples for forcing the model to learn the adversarial data distribution, which inevitably imposes a large computational overhead and results in a decrease in the generalization performance on clean data. In this paper, we show that progressively enhancing the adversarial strength of adversarial examples across training epochs can effectively improve the model robustness, and appropriate model shifting can preserve the generalization performance of models in conjunction with negligible computational cost. To this end, we propose a successive perturbation generation scheme for adversarial training (SPGAT), which progressively strengthens the adversarial examples by adding the perturbations on adversarial examples transferred from the previous epoch and shifts models across the epochs to improve the efficiency of adversarial training. The proposed SPGAT is both efficient and effective; e.g., the computation time of our method is 900 min as against the 4100 min duration observed in the case of standard adversarial training, and the performance boost is more than 7% and 3% in terms of adversarial accuracy and clean accuracy, respectively. We extensively evaluate the SPGAT on various datasets, including small-scale MNIST, middle-scale CIFAR-10, and large-scale CIFAR-100. The experimental results show that our method is more efficient while performing favorably against state-of-the-art methods.

Key words: Adversarial training; Adversarial attack; Stochastic weight average; Machine learning; Model generalization

https://doi.org/10.1631/FITEE.2300474 **CLC number:** TP391.1

1 Introduction

Adversarial attacks (Moosavi-Dezfooli et al., 2016; Papernot et al., 2016b; Kurakin et al., 2017; Ding et al., 2021; Hu and Sun, 2021; Liu et al., 2022) have been emerging as a significant threat to deep learning model robustness, which constitutes a reason for alarm, since such attacks can fool the models into arriving at wrong predictions through imperceptible but maliciously manipulated perturbations, i.e., adversarial examples, resultant to which there arises a serious safety issue for concerning real-world deployment of deep learning models (Cheng et al., 2018; Eykholt et al., 2018; Finlayson et al., 2019; Doan et al., 2020). To mitigate the adversarial attacks, traditional methods leverage manual intervention or prior assumptions to pick out the adversarial

‡ Corresponding author

* Project supported by the Scientific Research and Development Foundation of Fujian University of Technology, China (No. GYZ220209)

ORCID: Wei LIN, https://orcid.org/0000-0002-7797-2022; Lichuan LIAO, https://orcid.org/0000-0001-8999-1573

© Zhejiang University Press 2024

examples in the inference phase (Guo et al., 2017; Buckman et al., 2018; Huang et al., 2019). However, such defenses can be easily circumvented by stronger attacks (Athalye et al., 2018). Alternatively, adversarial training relies on online defending in lieu of the hand-crafted detection pre-processing. It minimizes the loss on online-generated adversarial examples against the model at each training epoch, forces the model to generate more robust features, and thereby improves the model robustness.

To improve the efficiency, there are emerging efforts to design efficient and scalable adversarial training methods. On one hand, some researchers work on the single-step adversarial perturbation generation with fast gradient sign method (FGSM) (Goodfellow et al., 2015) to relieve the computational overhead. Despite the effectiveness of FGSM, the methods based on it generally suffer from catastrophic overfitting due to the insufficient diversity of training samples. The dominant solutions employed to address this problem include early stopping (Wong et al., 2020), drop layers (Vivek and Babu, 2020), and loss regularizations (Andriushchenko and Flammarion, 2020). On the other hand, some research follows the idea of model compression to reduce the model parameters and eliminate the additional training time. Typical methods include pruning (Madaan et al., 2020), knowledge distillation (Papernot et al., 2016a; Goldblum et al., 2020), and adversarial generator (Baluja and Fischer, 2018).

Although much effort has been made, there is still a lack of specific efficient designs for adversarial training methods considering the need to take advantage of consistency of adversarial examples and models across training epochs. In particular, adversarial training methods newly generate adversarial examples and update model parameters via feedback propagation at each epoch. Such a design can suffer from two limitations. First, the generated adversarial examples at each training epoch have high redundancy. This statement can be supported by existing methods (Zheng et al., 2020), where the adversarial examples generated for one model can still stay adversarial to another model trained on the same dataset. As a result, the generation of discrete new online adversarial examples may lead to a huge redundancy and cost large additional computational overhead. Second, the approach of newly generated adversarial examples corresponding to every epoch is characterized by a lack of consistent robust feature representation generation when it comes to convolutional neural network (CNN) models (Lecun et al., 1998) trained on limited adversarial data, and this generation is a feature well-known to be essential for the maintenance of high accuracy on clean data. According to Yang et al. (2020), most existing adversarial training methods often lead to a significant increase in the generalization error on clean testing data.

To solve the above limitations, we propose a successive perturbation generation scheme for adversarial training (SPGAT), which successively generates adversarial examples with a single-step attack and shifts models across the training epochs to enhance the efficiency of adversarial training. Specifically, we first use the adversarial examples from the previous epoch instead of the original input as the starting point in the next epoch to accumulate the attack strength. This design is reasonable based on two observations: (1) adversarial examples from adjacent epochs are proved to have high transferability (Zheng et al., 2020), and (2) recent studies show that it is not necessary to use strong adversarial attacks in the early stage of adversarial training because the model is fragile. When the model becomes more robust as the training goes deeper, strong attacks are needed for further improving the model robustness. In contrast with Li et al. (2021), in which FGSM attack was used in the early stage of training and then it switched to project gradient descent (PGD) attacks to enhance the attack strength, we propose a more efficient successive adversarial example generation scheme to progressively accumulate the attack strength via only FGSM attack. Second, to preserve the generalization performance of trained models under strong attacks, we further periodically aggregate the models from the previous epochs to alleviate drastic model parameter change in the training process.

It needs to be emphasized that the proposed SPGAT has several advantages, as indicated below:

1. It brings considerable computational savings and improves the scalability of current adversarial training methods. With a successive adversarial example generation scheme, we do not need to generate perturbations from the original input data iteratively at each training epoch.

2. It naturally leads to a higher accuracy on

clean testing data, similar to the stochastic weight average (SWA) scheme (Izmailov et al., 2018) in network optimization, which achieves an ensemble effect with negligible additional computation cost.

3. It can be easily incorporated with these existing adversarial training methods where the proposed scheme has no conflicts with these other methods.

We apply our proposed method on the CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton, 2009) datasets and achieve impressive results. For example, as can be seen in Fig. 1, our proposed method improves the PGD-20 accuracy of WideResNet trained with standard adversarial training (PGD-10) from 48.93% to 55.94% on the CIFAR-10 dataset. Moreover, compared with PGD-10 adversarial training which consumes 4100 min, our SPGAT takes a significantly lesser duration of time, at only 900 min. Our contributions can be summarized as follows:

1. We propose a successive adversarial training method that connects the adversarial examples and shifts models across training epochs, which significantly improves the efficiency and the generalization performance of CNN models.

2. Extensive experiments show that, with comparable training time, our proposed method outperforms the competitive baseline adversarial methods on image classification benchmarks, including CIFAR-10 and CIFAR-100.

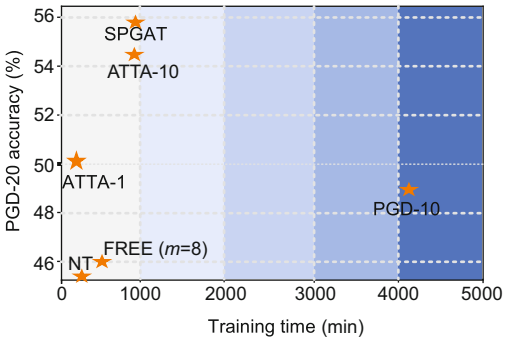

Fig. 1 Training time and PGD-20 accuracy of the WideResNet model trained on the CIFAR-10 dataset using different adversarial training methods (With comparable training time, our proposed SPGAT surpasses state-of-the-art ATTA-1 by 6% in the PGD-20 accuracy)

2 Related works

2.1 Adversarial attacks

Adversarial attacks have been emerging as one of the significant reasons for compromise in the robustness of deep learning models, which constitute a source of alarm, because given a well-trained task model and a clean input image, adversarial attacks try to fool the task model with a similar-looking but maliciously hand-crafted version of the original image (Carlini et al., 2017). Recent studies concerning adversarial attacks can be divided into two categories: optimization-based attacks and gradient-based attacks.

1. Optimization-based attacks. Szegedy et al. (2014) first proposed a method for both distilled and undistilled models by solving the optimization problem with limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS). Zhang JB et al. (2022) further extended the BFGS algorithm for limited pixel attack, and proposed the limited pixel BFGS (LP-BFGS) attack method. Chen PY et al. (2017) proposed a zero-order optimization algorithm to estimate the gradient by performing finite difference on the query results.

2. Gradient-based attacks. Goodfellow et al. (2015) proposed the FGSM algorithm requiring only one step of gradient update and achieved state-of-the-art (SOTA) results. Madry et al. (2018) transformed the one-step generation manner into multi-step one and proposed PGD, which has become one of the most popular attack methods. Croce and Hein (2020) further ensembled the gradient-based attack into a stronger attack, termed as AutoAttack (AA), which has been recognized as the strongest attack so far. Yamamura et al. (2022) employed the conjugate gradient (CG) method to enhance the search process of gradient descent, thereby avoiding the non-convexity and non-linearity issues presented in existing gradient descent techniques. As a result, they achieved SOTA results on large-scale datasets.

Both gradient- and optimization-based attacks are designed to generate model-specific adversarial examples. Szegedy et al. (2014) demonstrated the transferability of adversarial examples; i.e., an adversarial example generated from one target model can also disturb other models. An amount of research has been devoted to improving the transferability of adversarial examples. Specifically, Dong et al. (2018) proposed MI-FGSM by adding a momentum term to the iterative process, yielding a high attack efficiency. Wang et al. (2021) argued that brute-force

degradation would introduce model-specific local optimum into adversarial examples, thus limiting the transferability. Hence, they proposed the feature importance-aware attack (FIA), which disrupts important object-aware features that dominate model decisions consistently. Chen B et al. (2023) developed an adaptive ensemble transfer attack to boost the transferability across models with wide differences, such as from CNNs to vision Transformers (ViTs).

2.2 Adversarial training

The vulnerability of CNN models to adversarial examples has motivated studies into adversarial training. The initial idea of adversarial training is brought to light by Madry et al. (2018), in which CNN models were trained on a mixture of adversarial examples and clean examples to improve the model robustness. Goodfellow et al. (2015) proposed FGSM to produce strong adversarial examples during training. However, these methods remain vulnerable to stronger iterative attacks. Recent studies proposed the training of CNN models with adversarial data only and formed the training process as a game between adversarial example strength maximization and model loss minimization. To generate strong adversarial examples, current adversarial training methods can be divided into two categories: iterative-based methods and ensemble-based methods.

1. Iterative-based methods. These methods enhance the strength of adversarial examples through iterative accumulation. Madry et al. (2018) first proposed the iterative gradient-based attack PGD-k for accumulating the attack strength and significantly increased the model robustness against adversarial examples. Following this design, derivative methods have been further developed. Cai et al. (2018) proposed curriculum adversarial training (CAT) to gradually increase the iteration step number (i.e., the value of k) of PGD until the model achieves a high accuracy against the current attack. Contrarily, friendly adversarial training (FAT) (Zhang JF et al., 2020) employs early stopping when performing PGD iterations to realize the training process in a more practical way. In general, iterative-based methods can generate strong adversarial examples but are faced with high computation cost due to the iterative calculation at each training epoch.

2. Ensemble-based methods. These methods introduce ensemble learning into the adversarial training process, where individually non-robust submodels that produce diverse outputs are assembled to improve the overall robustness. Tramer et al. (2018) first proposed ensemble adversarial training (EAT), in which the adversarial examples were generated from a pre-trained model instead of the current model. To further promote the diversity of features among target models, Pang et al. (2019) proposed an adaptive diversity promoting regularizer to force different methods prefer diverse predictions. Kariyappa and Qureshi (2019) proposed the maximization of the cosine similarity among the input gradients of each sub-model. Yang et al. (2020) measured the feature overlapping to diversify the vulnerability in each sub-model in EAT.

Nevertheless, constituents of existing literature all focus on maximizing the attack strength without considering the computation cost for practicability. In the present study, we propose simple yet effective modifications to improve the efficiency of current adversarial training methods.

2.3 Efficient adversarial training

The key efficiency bottleneck of standard adversarial training is that the iterative generation of adversarial examples has quadratic memory and computational complexity. Thus, many attempts are proposed to relieve the computational overhead by simplifying the inner maximization process. Specifically, free adversarial training proposed by Shafahi et al. (2019) reuses the gradients in forward pass, and thus model parameters and image perturbations can be updated simultaneously. Based on free adversarial training, Wong et al. (2020) further proposed fast adversarial training, in which single-step FGSM was initialized randomly to reduce the magnitude of perturbations. However, it is found that single-step adversarial training generally suffers from catastrophic overfitting due to the lack of diverse training examples. To alleviate the overfitting, many improvements have been proposed, such as early stopping (Wong et al., 2020), drop layers (Vivek and Babu, 2020), and loss regularizations (Andriushchenko and Flammarion, 2020).

Another line taken by studies in the literature is the adoption of model compression to reduce the adversarial examples and computational overhead. For

example, Madaan et al. (2020) proposed a Bayesian framework to prune features with high vulnerability to enhance the robustness of CNN models. Zheng et al. (2020) used the transferability of adversarial examples across the training epochs and proposed to reuse the adversarial examples to reduce the generation time. Papernot et al. (2016a) extracted the distilled knowledge from CNN models to reduce the amplitude of network gradients exploited by adversaries to craft adversarial samples. These methods can slightly reduce the training time, but they neglect to consider the generalization performance on clean datasets and are thus incapable of handling model training on large datasets. To bridge this gap, we propose a successive single-step perturbation generation regime that can improve the robustness of CNN models while significantly reducing the computation cost.

3 Method

In this section, we first briefly revisit the preliminaries of adversarial training and then introduce our proposed method.

3.1 Preliminaries

For a given network f parameterized by θ, $\mathcal{L}(f_\theta, x, y)$ denotes the loss of the network on the example $(x, y) \sim \mathcal{D}$, where $\mathcal{D}$ is the data generating distribution. The formulation of adversarial training can be represented as follows:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{x^{\mathrm{adv}}} \mathcal{L}(f_\theta, x^{\mathrm{adv}}, y)], \tag{1}$$

where x^{adv} denotes the adversarial example, and is obtained by adding a perturbation δ over the original data x:

$$x^{\mathrm{adv}} = x + \delta \quad \text{s.t.} \quad \delta \in \mathcal{S}, \tag{2}$$

where $\mathcal{S}$ denotes the region within the ϵ-perturbation range under the ℓ_∞ threat model for each example, i.e., $\mathcal{S} = \{\delta : \|\delta\|_\infty \leq \epsilon\}$, where the adversary can change the input coordinate x_i at most ϵ.

In brief, the basic idea of adversarial training is a min–max optimization, which, given the image data x, involves the objective of finding a perturbation δ that maximizes the model loss on $x + \delta$. Then the model is trained on the generated adversarial example x^{adv} to minimize the loss. Instead of solving non-concave optimization independently, adversarial attacks are usually used to approximate the internal maximization over $\mathcal{S}$. There are two adversarial training methods: FGSM attack and PGD-k attack.

1. FGSM attack. Goodfellow et al. (2015) performed single-step gradient descent to find adversarial perturbations x^{adv} to approximate the internal maximization, which is formalized as follows:

$$x^{\mathrm{adv}} = x + \epsilon \cdot \mathrm{sign}(\nabla_x \mathcal{L}(f_\theta, x, y)). \tag{3}$$

The use of the single-step generation scheme ensures that the FGSM adopted in their study is fast, although it can easily lead to overfitting as observed in Wong et al. (2020).

2. PGD-k attack. Madry et al. (2018) used multi-step PGD to approximate the inner maximization, an approach that offers greater accuracy compared with FGSM but is computationally expensive, formalized as follows:

$$x_{t+1}^{\mathrm{adv}} = \Pi_{x+\mathcal{S}}(x_t^{\mathrm{adv}} + \alpha \cdot \mathrm{sign}(\nabla_x \mathcal{L}(f_\theta, x, y))), \tag{4}$$

where x_t^{adv} is initialized as the clean input x, and Π refers to the projection operation, which ensures projecting the adversarial examples back to the ball within the radius ϵ of the clean data point. PGD-k denotes the k-step PGD adversarial attack; the larger the k, the stronger the attack and the higher the computation cost.

3.2 Successive adversarial training

In the present study, we propose the SPGAT method to progressively enhance the attack strength of adversarial examples and encourage the robust feature generation for improving the efficiency of adversarial training. An overview of our proposed method is shown in Fig. 2. Unlike in standard adversarial training, where adversarial examples are generated from the original input in each training epoch, shown in Fig. 2a, we propose to successively generate adversarial examples and shift models from previous epochs, shown in Fig. 2b.

3.2.1 Adversarial example generation

Recent studies (Wei et al., 2019; Zhu et al., 2019) on adversarial training show that the adversarial examples generated from adjacent epochs tend to carry similar adversarial information. Therefore, it would

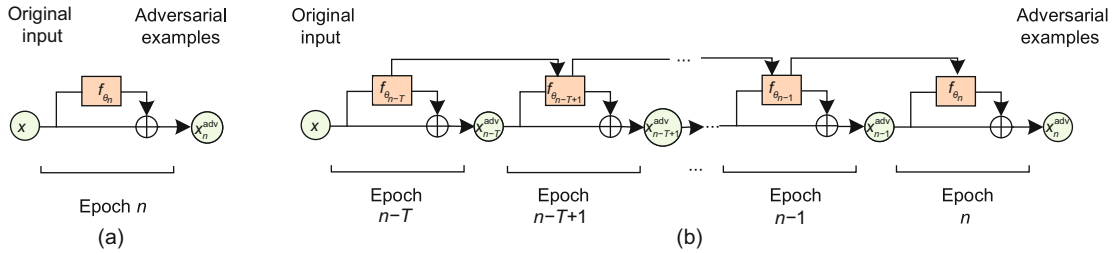

Fig. 2 Standard adversarial training process where the adversarial examples are generated from the original input at each epoch (a) and details of the proposed SPGAT method, which successively generates the adversarial examples and shift models from adjacent epochs (b)

be beneficial to reduce these redundancies for improving efficiency. Motivated by Zheng et al. (2020), for epoch $n+1$, we use the adversarial examples generated in epoch n as the starting point to generate the adversarial examples, which can be represented as

$$x_{n+1}^{\mathrm{adv}} = \mathcal{A}(f_{\theta_n}, x, y, M(x_n^{\mathrm{adv}})), \tag{5}$$

where $\mathcal{A}$ is the attack algorithm, corresponding to which the present study uses, FGSM attack, f_{θ_n} is the model in the n^{th} epoch, and M is a transformation function, which transforms x_n^{adv} into x_{n+1} as the starting input for the next epoch.

To adapt the perturbation on the newly augmented images in each epoch, we adopt the inverse data augmentation technique used in Zheng et al. (2020) to enhance the transferability of the adversarial examples. Specifically, the inverse transformation $T^{-1}(\cdot)$ is adopted to calculate the inverse perturbation $T^{-1}(\nabla_n)$ on the newly augmented image x^{aug}. By adding $T^{-1}(\nabla_n)$ to x^{aug}, we can store and transfer all perturbation information for the next epoch. The process can be described as

$$M(x_n^{\mathrm{adv}}) = T^{-1}(\nabla_n) + x_n^{\mathrm{adv}}. \tag{6}$$

Note that to ease the effect of early perturbations, we follow the method adopted in Zheng et al. (2020) to reset the perturbation and let adversarial perturbations be accumulated from the beginning periodically.

3.2.2 Model updates

Previous studies update the model parameters by minimizing the loss on online-generated adversarial data as described in Eq. (1). However, such a scheme relies solely on the generated adversarial data with limited capacity while neglecting the consistent robust feature generation in CNN models, leading a decrease in the generalization performance on clean data. Moreover, due to the lack of diverse training samples, the model parameters usually suffer from drastic change during the training process, which can easily cause the problem of overfitting. To this end, we propose to shift the trained models across training epochs to further boost the model robustness on clean data and alleviate the overfitting. Specifically, we adopt the SWA model updating scheme (Izmailov et al., 2018), where the model weights generated from previous epochs are aggregated to the average and the model is updated with the averaging weight periodically. The process can be represented as follows:

$$\theta^{\mathrm{swa}} = \frac{\theta^{\mathrm{swa}} T + \theta}{T+1}, \tag{7}$$

where T is the number of models to be aggregated, which is also referred to the shift cycle. Note that to generate the adversarial examples properly in forward pass, the batch normalization statistics in the network are recalculated every time corresponding to the resetting of the perturbation accumulation.

3.2.3 Training routine

The overall training pseudo code is provided in Algorithm 1. We first initialize the perturbation δ randomly from a uniform distribution within the range of $(-\epsilon, \epsilon)$. Then for each batch of training data during the training process, we generate the adversarial examples in each epoch following the function of Eq. (5) to accumulate the attack strength. Note that we use FGSM as the adversarial attack in the perturbation generation. Backward propagation is then performed on the trained model to update the model parameters. To make the training process more stable and escape from overfitting, we periodically update the model parameters by Eq. (7) and reset the perturbation for accumulating attack from

the beginning. This training process is performed on all B batches of training data for N epochs.

Algorithm 1: Successive perturbation generation scheme for adversarial training

Input: training data $\mathcal{D}=\{X,Y\}$, perturbation boundary ϵ, shift cycle T, hyperparameter C

Output: trained model f with parameter θ

Initialize δ from a uniform distribution within $(-\epsilon,\epsilon)$

for epoch $=1, 2, ..., N$ **do**

  for $i=1,2,...,B$ **do**

    $\delta_i \leftarrow \delta_i+\alpha\cdot\text{sign}(\nabla_{\delta_i}\mathcal{L}(f_\theta,x_i+\delta_i,y_i))$

    $\delta_i \leftarrow \max(\min(\delta_i,\epsilon),-\epsilon)$

    $x_i^{\text{adv}} \leftarrow M(x_{i-1}^{\text{adv}})+\delta_i$

    $\theta \leftarrow \theta-\nabla_\theta\frac{\partial\mathcal{L}(f_\theta,x_i^{\text{adv}},y_i)}{\partial\theta}$

  if epoch $\%$ $T=0$ **then**

    $\theta^{\text{swa}} \leftarrow \frac{\theta^{\text{swa}}\cdot t+\theta}{t+1}$

    $t \leftarrow t+1$

  if epoch $\%$ $C=0$ **then**

    $t \leftarrow 0$

    $f_{\theta^{\text{swa}}} \leftarrow \text{UPDATE\_BN}(f_{\theta^{\text{swa}}},\mathcal{D},\delta)$

    $f_\theta \leftarrow f_{\theta^{\text{swa}}}$

    $x_i^{\text{adv}} \leftarrow x_i$

return f_θ

4 Discussion

4.1 Analysis of adversarial example generation

In standard adversarial training, adversarial examples are generated from clean data using strong attacks in each epoch. In this study, we demonstrate the gradual accumulation of the attack strength by successively generating adversarial examples with a single-step FGSM attack. Specifically, we take the adversarial examples generated in the last epoch as the starting point in the next epoch to generate new adversarial examples. To verify the fact that such successive generation schemes can effectively accumulate adversarial attack strength, we train models on the MNIST and CIFAR-10 datasets with FGSM attacks and evaluate the accuracy and loss values of the model by using adversarial examples generated from accumulated attacks. The experimental results are reported in Fig. 3. It can be observed that when there is no accumulation (accumulation epoch = 0), i.e., when the adversarial examples are generated from clean data, the robustness of the model achieves 56% on the CIFAR-10 dataset. With the increase in the number of accumulational adversarial perturbations, the accuracy of the CIFAR-10 model decreases rapidly, and the corresponding adversarial attack loss increases. This indicates that the adversarial examples generated in the successive manner can effectively enhance the attack strength even with single-step FGSM, which allows us to achieve significant savings in terms of computation cost resultant to the removal of time-consuming iterative perturbation generation in each epoch.

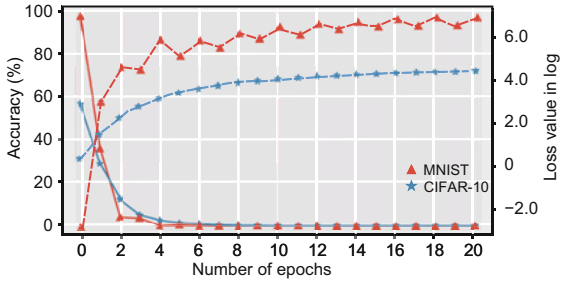

Fig. 3 Performance comparison of attack strength accumulation through epochs (Solid lines denote the accuracy, y-axis on the left, and dash lines denote the loss values, y-axis on the right)

When compared to the adversarial training with transferable adversarial examples (ATTA) method, the study that is most closely related to our research is Zheng et al. (2020). This work involved transferring adversarial examples from the final epoch to serve as the starting point in the subsequent epoch, followed by the application of a PGD attack to progressively strengthen the attack. However, this study focused only on finding strong adversarial examples without considering the generalization ability of the studied model, leading to a significant decrease in clean data. To better fit the successive generation scheme in adversarial training, we elaborately involve model shifting in our SPGAT to avoid drastic model parameter change. Moreover, we use only FGSM attack in our SPGAT to make it suitable for model training, since weak adversarial examples are enough for the early stage of training; however, as the training goes deeper, the adversarial examples get stronger via accumulation from previous epochs, and this enables the training exercise to produce more robust models. Please refer to the ablation study in Section 5.3 for more comparisons and analysis.

4.2 Analysis of model shifting

Traditional adversarial training augments the datasets with adversarial samples to encourage the model to learn the adversarial sample distribution; however, this also leads to a poor generalization ability of the model. Taking ATTA as an example, we observe that the robust accuracy against PGD-20 achieved 10% improvements over standard adversarial training but the natural accuracy dropped by 5%, as reported in Zheng et al. (2020). When the SPGAT explained in the present study is applied, the robust accuracy increases by 12% and the clean accuracy increases by 3% over standard adversarial training, as shown in Fig. 1. The model shifting equips our SPGAT with strong generalization ability in terms of the performance on clean data. Such an endowment provides benefits in terms of two resources: (1) The model is encouraged to learn more towards the distribution of adversarial samples in adversarial training since the model is trained only on adversarial data. Therefore, it would be beneficial to involve model characteristics from previous epochs in our method. (2) The model from a different epoch can be seen as a different version that produces different feature representations, and hence aggregating the trained models from different epochs can be analogous to the ensemble skills in ensemble learning for eliminating overfitting on a limited amount of training data. More experimental data are provided in the ablation study in Section 5.3.

5 Experiments

5.1 Experimental setup

1. Architecture and datasets. Following previous studies (Madaan et al., 2020; Zheng et al., 2020), we use WideResNet34 as our defalut model architecture, and train the model on two popular benchmark datasets: CIFAR-10 and CIFAR-100. The CIFAR-10 dataset consists of 60 000 32×32 color images within 10 classes, with 6000 images per class. There are 50 000 training images and 10 000 test images. The CIFAR-100 is just like the CIFAR-10, except that it has 100 classes containing 600 images each, and contains 500 training images and 100 test images per class.

2. Adversarial attacks. For the adversarial attacks, we generate the adversarial examples by PGD attack using an l_∞ threat model with 20 steps for the CIFAR-10 and CIFAR-100 datasets. The maximum perturbation ϵ is fixed at 0.031, with the step size of $2\epsilon/k$, where k is the iteration number.

3. Compared methods. We compare the performance of our proposed SPGAT with those of well-known SOTA methods including standard PGD-10 adversarial training, FREE ($m = 8$) (Shafahi et al., 2019), which uses single-step FGSM with eight hop steps, and ATTA-k (Zheng et al., 2020), which is the most pertinent research working in k-step iteration PGD adversarial training with transferable adversarial examples. Here, we set $k = 1$ and 10 in ATTA for fair comparison. All the compared methods are performed using the default codes and settings as mentioned by the respective authors.

4. Training details. In our experiments, we train models on the CIFAR-10 and CIFAR-100 datasets for 200 epochs using a stochastic gradient descent (SGD) optimizer with the initial learning rate reckoned as 0.1, and the decay is established by a factor of 0.1 at 50% and 75% of the total epoch. We use the batch size of 64 for all the experiments. In our proposed SPGAT, the checkpoint of the 160^{th} epoch is chosen as the initial point of model shifting, which is similar to that of the SWA scheme applicable in the case of network optimization. The adversarial examples are generated by the single-step attack FGSM for training during the whole training process. All the experiments are implemented using PyTorch on an Intel Core i9 processor with 32 GB of memory and an NVIDIA TITAN Xp GPU.

5.2 Main results

5.2.1 White-box attacks

We first conduct a series of white-box strong adversarial attacks to evaluate the efficiency and robustness of the trained models: PGD with 20 and 50 iterations on cross-entropy loss (PGD-k), Carlini–Wagner loss (CW-k) (with a step size of $2\epsilon/k$, where k represents the number of iterations), and AA (which has been recognized as the strongest attack so far). The confidence parameter of the CW loss is set at 50. We report the robust accuracy and training time of the final trained model in Table 1. As we can see, stronger adversarial example generation during adversarial training can lead to better model robust accuracy. With our successive adversarial

example generation, the model trained with SPGAT achieves the robust accuracies of 55.94% and 27.45% against PGD-20 on the CIFAR-10 and CIFAR-100 datasets, respectively. Compared to baselines, our SPGAT achieves around 5% better robust accuracy than ATTA-1 with comparable training time. It has been indicated in the literature that ATTA-10 had a better robust accuracy than ATTA-1, a finding consistent with our observations. However, the time consumption involved in PDG-10, FREE, and ATTA is, on average, about two times greater than ours. More importantly, concomitant with the increase of model robust accuracy with the use of different adversarial training methods, the natural accuracy on clean testing data decreases. This is because the models trained on strong adversarial examples are encouraged to learn the adversarial sample distribution, which thus leads to poor generalization ability of these models on clean data. With the model shifting to relieve the model change, our proposed SPGAT can maintain high accuracy on clean data, as evidenced from the fact that the natural accuracies achieved 88.90% and 61.94% on the CIFAR-10 and CIFAR-100 datasets, respectively.

5.2.2 Unseen adversaries

We also evaluate our approach against unforeseen adversaries, e.g., robustness on different attack threat radii ϵ, or even on different norm constraints (e.g., l_2 and l_1). The results are reported in Table 2. It can be observed that compared with SOTA methods, the proposed SPGAT could effectively improve the robustness against l_∞ and l_2 adversaries on both the CIFAR-10 and CIFAR-100 datasets. For

Table 1 Robust accuracy of models trained with PGD-10, FREE, ATTA, and SPGAT without early stopping on the CIFAR-10 and CIFAR-100 datasets

Dataset	Method	Accuracy (%)						Time (s)
		Natural	PGD-20	PGD-50	CW-20	CW-50	AA	
CIFAR-10	PGD-10	85.23	48.93	48.63	48.74	48.28	44.12	1235.10
	FREE (m=8)	85.75	45.76	45.52	44.95	44.45	41.05	**128.16**
	ATTA-1	83.36	50.05	49.90	49.02	48.75	45.64	263.34
	ATTA-10	84.43	54.65	53.74	54.25	54.01	50.79	1425.79
	SPGAT (ours)	**88.90**	**55.94**	**55.80**	**54.96**	**54.71**	**52.16**	269.79
CIFAR-100	PGD-10	60.29	26.84	26.44	26.44	26.25	22.48	1234.11
	FREE (m=8)	60.11	26.79	22.66	25.69	25.60	21.19	**127.59**
	ATTA-1	59.07	21.58	22.82	21.14	20.92	17.56	262.68
	ATTA-10	55.09	23.23	23.00	22.85	22.73	19.88	1429.49
	SPGAT (ours)	**61.94**	**27.45**	**27.09**	**28.33**	**27.90**	**23.95**	270.41

All statistics are evaluated against PGD/CW attacks with 20/50 iterations and a random restart for $\epsilon = 8/255$. We highlight the best results in bold and the second-best with underline

Table 2 Robust accuracy of WideResNet34 trained with l_∞ of $\epsilon = 8/255$ boundary against unseen attacks

Dataset	Method	Accuracy (%)					
		l_∞		l_2		l_1	
		ϵ=4/255	16/255	150/255	300/255	2000/255	4000/255
CIFAR-10	PGD-10	67.92	21.52	52.49	24.93	67.36	46.99
	FREE (m=8)	64.66	15.94	51.86	26.74	64.15	46.56
	ATTA-1	67.99	16.95	58.85	28.50	**72.47**	**57.97**
	ATTA-10	69.94	22.43	58.15	28.09	70.91	53.89
	SPGAT (ours)	**70.66**	**42.43**	**59.28**	**44.97**	70.99	56.70
CIFAR-100	PGD-10	40.67	9.96	30.69	12.99	42.43	28.24
	FREE (m=8)	42.50	8.74	**34.63**	15.75	**46.32**	**34.82**
	ATTA-1	35.18	13.28	26.77	15.73	39.10	26.32
	ATTA-10	36.17	12.45	25.50	13.65	36.10	23.06
	SPGAT (ours)	**42.70**	**13.44**	32.74	**16.22**	45.72	31.96

For unseen attacks, we use PGD-50 under different sized l_∞ balls and other types of norm ball, e.g., l_2 and l_1. We highlight the best results in bold and the second-best with underline

l_1 adversaries, ATTA-1 achieves the best result on the CIFAR-10 dataset but its performance degrades dramatically on the large-scale CIFAR-100 dataset. Similarly, FREE ($m = 8$) has the best result on the CIFAR-100 dataset but not on the CIFAR-10 dataset. This reveals the instability and limitations of these methods. Contrarily, our proposed SPGAT can have stable and comparable performance on both datasets.

5.2.3 Black-box transfer attacks

To evaluate the effectiveness of our proposed SPGAT in practical defense scenarios, we test the model under black-box transfer attacks, where adversarial examples are generated from a source model and then are transferred to the target model. In this evaluation, adversarial examples are crafted from PreActResNet18 (source model) trained with standard adversarial training, where we consider FGSM, PGD-50, and CW-20 as black-box adversaries. Then we perform a black-box transfer attack evaluation on WideResNet34-10 (target model). The results are provided in Table 3, where we can find that black-box transfer attacks are always substantially weaker than white-box attacks, which is consistent with the observations drawn in Madry et al. (2018). It is interesting to find that with the increase of k in the ATTA method, the robust accuracy tends to decrease. One possible reason is that iterative adversarial examples generated in the source model can have a higher transferability to the iterative adversarial examples generated in the target model, since more redundancy information is contained. Compared to baselines, our method still achieves the best robustness under black-box attacks.

Table 3 Robust accuracy of the WideResNet34-10 model trained on the CIFAR-10 and CIFAR-100 datasets against black-box transfer attacks, with adversarial examples being crafted from PGD-10 pre-trained PreActResNet18

Dataset	Method	Accurary (%)		
		FGSM	PGD-50	CW-20
CIFAR-10	PGD-10	67.59	67.58	78.11
	FREE (m=8)	68.02	66.19	77.49
	ATTA-1	68.85	65.25	77.79
	ATTA-10	66.82	66.27	77.79
	SPGAT (ours)	71.34	70.21	83.03
CIFAR-100	PGD-10	42.71	42.13	56.55
	FREE (m=8)	41.70	40.81	56.51
	ATTA-1	42.19	41.88	54.77
	ATTA-10	40.63	39.79	50.74
	SPGAT (ours)	46.26	45.93	59.55

We choose l_∞ threat model with $\epsilon = 8/255$ for FGSM and PGD-50. Specially, for CW-20, ϵ is fixed to 160/255

Table 4 Performance comparision with the use of PGD attacks with different k in SPGAT

k	Accurary (%)		Training time (min)
	PGD-20	AA	
1	55.94	51.60	900.12
3	54.67	50.92	1990.23
5	53.47	49.34	2653.59
7	53.39	49.15	3560.12
10	53.43	49.20	4903.11

5.3 Ablation study

5.3.1 Effect of using different adversarial attacks in perturbation generation

We use different adversarial attacks in our SPGAT method and show the results in Table 4. Note that $k = 1$ denotes the FGSM attack, and this is equivalent to our method. From the results, we can see that the models trained with more iterative steps are characterized by a small decrease in PGD-20 accuracy together with a much higher computation cost. It indicates that stronger adversarial examples do not necessarily lead to better adversarial robustness. The main reason is that excessive perturbation on data may impede the model robustness since the model is fragile in the early stage of training. In the SPGAT method discussed in the present study, progressive enhancement of the adversarial attack strength, which is more suitable for model training, has been adopted.

5.3.2 Effect of model shifting

One of the important advantages of the proposed SPGAT is that we aggregate the models periodically to improve the stability of the models. To validate the effectiveness of model shifting, we train the model with different aggregation periods T and show the results in Fig. 4. Note that $T = 0$ denotes the adversarial training method without model shifting. With the increase of T, the trained model achieves better performance in both clean data and adversarial data. The performance gains mainly come from the involving of model characteristics from previous epochs. Applying model shifting with negligible computation cost can significantly improve

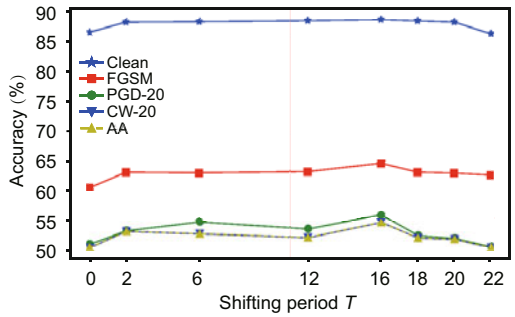

Fig. 4 Clean and adversarial robust accuracy of the model trained on the CIFAR-10 dataset with our proposed SPGAT for different shifting periods T

the generalization ability of the trained models. The robust accuracy reaches its peak at $T = 16$; when the value of T continues to get larger, the clean accuracy maintains almost the same value, but the robust accuracy decreases. This is because the more models are aggregated, the ability for robust information extraction would be smoothed. We set $T = 16$ throughout the experiments to obtain the best performance.

5.3.3 Effect of adversarial perturbation resetting

During the successive adversarial example generation process, we reset the perturbation and let adversarial perturbations be accumulated from the beginning periodically to avoid potential error accumulation. In the experiments, we generally follow Zheng et al. (2020) in setting the reset period C as 10. Here we further investigate the effect of C on the overall model performance. Specifically, we test the model performance using different values of C and report the results in Table 5. Note that $C = 0$ denotes the accumulation of the adversarial perturbations throughout the whole training process. As we can see, if the adversarial perturbations are not reset during the training, the model performance significantly degrades. This is because that only models do not easily changes during training, the perturbation transferability would be decreased as the training goes deeper and the model parameters are largely changed. As the adversarial perturbation resetting is adapted, both the clean and robust accuracies are improved. When the value of C gets larger, the robust accuracy improves before 15 and the clean accuracy decreases after 10. This is because the adversarial examples are stronger when C is larger, thereby leading to a decrease in clean accuracy, and the potential error also accumulates, with the result that the robust accuracy is not improved constantly. We use the value of 10 in the experiments to achieve the trade-off.

Table 5 Clean and adversarial robust accuracy of the model trained on the CIFAR-10 dataset with our proposed SPGAT for different perturbation reset periods C

C	Accuracy (%)				
	Clean	FGSM	PGD-20	CW-20	AA
0	78.64	52.62	41.25	41.03	37.88
5	88.65	60.61	52.73	52.18	47.62
10	88.90	63.25	55.94	54.96	52.16
15	87.39	63.27	55.91	54.69	52.17
20	85.57	63.02	55.99	54.75	51.29
25	83.36	61.31	53.64	53.55	49.92

6 Conclusions

In this paper, we present SPGAT for improving the robustness of CNN models. Specifically, the SPGAT successively generates the adversarial examples with a single-step attack and shift models across the training epochs to enhance the efficiency of adversarial training. More importantly, this strategy greatly improves the generalization ability of models, thus imbuing them with the ability to maintain high accuracy on clean data. In the experiments, our proposed SPGAT demonstrates outstanding performances under various attacks, including white-box attacks, unseen adversaries, and black-box attacks with comparable training time.

Contributors

Wei LIN designed the research and drafted the paper. Wei LIN and Lichuan LIAO revised and finalized the paper.

Conflict of interest

Both authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the first author upon reasonable request.

References

Andriushchenko M, Flammarion N, 2020. Understanding and improving fast adversarial training. Proc 34th Int Conf on Neural Information Processing Systems, Article 1346.

Athalye A, Carlini N, Wagner D, 2018. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. Proc 35th Int Conf on Machine Learning, p.274-283.

Baluja S, Fischer I, 2018. Learning to attack: adversarial transformation networks. Proc 32nd AAAI Conf on Artificial Intelligence, p.2687-2695. https://doi.org/10.1609/aaai.v32i1.11672

Buckman J, Roy A, Raffel C, et al., 2018. Thermometer encoding: one hot way to resist adversarial examples. Proc Int Conf on Learning Representations.

Cai QZ, Liu C, Song D, 2018. Curriculum adversarial training. Proc 27th Int Joint Conf on Artificial Intelligence, p.3740-3747.

Carlini N, Katz G, Barrett C, et al., 2017. Provably minimally-distorted adversarial examples. https://arxiv.org/abs/1709.10207

Chen B, Yin JL, Chen SK, et al., 2023. An adaptive model ensemble adversarial attack for boosting adversarial transferability. http://export.arxiv.org/abs/2308.02897

Chen PY, Zhang H, Sharma Y, et al., 2017. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. Proc 10th ACM Workshop on Artificial Intelligence and Security, p.15-26. https://doi.org/10.1145/3128572.3140448

Cheng YH, Lu F, Zhang XC, 2018. Appearance-based gaze estimation via evaluation-guided asymmetric regression. Proc 15th European Conf on Computer Vision, p.105-121. https://doi.org/10.1007/978-3-030-01264-9_7

Croce F, Hein M, 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. https://arxiv.org/abs/2003.01690v1

Ding KY, Liu XL, Niu WN, et al., 2021. A low-query black-box adversarial attack based on transferability. *Knowl-Based Syst*, 226:107102. https://doi.org/10.1016/j.knosys.2021.107102

Doan BG, Abbasnejad E, Ranasinghe DC, 2020. Februus: input purification defense against Trojan attacks on deep neural network systems. Proc Annual Computer Security Applications Conf, p.897-912. https://doi.org/10.1145/3427228.3427264

Dong YP, Liao FZ, Pang TY, et al., 2018. Boosting adversarial attacks with momentum. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9185-9193. https://doi.org/10.1109/CVPR.2018.00957

Eykholt K, Evtimov I, Fernandes E, et al., 2018. Robust physical-world attacks on deep learning visual classification. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.1625-1634. https://doi.org/10.1109/CVPR.2018.00175

Finlayson SG, Bowers JD, Ito J, et al., 2019. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287-1289. https://doi.org/10.1126/science.aaw4399

Goldblum M, Fowl L, Feizi S, et al., 2020. Adversarially robust distillation. Proc 34th AAAI Conference on Artificial Intelligence, p.3996-4003. https://doi.org/10.1609/aaai.v34i04.5816

Goodfellow IJ, Shlens J, Szegedy C, 2015. Explaining and harnessing adversarial examples. Proc 3rd Int Conf on Learning Representations.

Guo C, Rana M, Cisse M, et al., 2017. Countering adversarial images using input transformations. Proc 6th Int Conf on Learning Representations.

Hu YY, Sun SL, 2021. RL-VAEGAN: adversarial defense for reinforcement learning agents via style transfer. *Knowl-Based Syst*, 221:106967. https://doi.org/10.1016/j.knosys.2021.106967

Huang B, Wang Y, Wang W, 2019. Model-agnostic adversarial detection by random perturbations. Proc 28th Int Joint Conf on Artificial Intelligence, p.4689-4696.

Izmailov P, Podoprikhin D, Garipov T, et al., 2018. Averaging weights leads to wider optima and better generalization. Proc 34th Conf on Uncertainty in Artificial Intelligence, p.876-885.

Kariyappa S, Qureshi MK, 2019. Improving adversarial robustness of ensembles with diversity training. https://arxiv.org/abs/1901.09981

Krizhevsky A, Hinton G, 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report, Computer Science Department, University of Toronto, Canada.

Kurakin A, Goodfellow IJ, Bengio S, 2017. Adversarial examples in the physical world. Proc 5th Int Conf on Learning Representations.

Lecun Y, Bottou L, Bengio Y, et al., 1998. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

Li B, Wang SQ, Jana S, et al., 2021. Towards understanding fast adversarial training. https://arxiv.org/abs/2006.03089v1

Liu L, Du Y, Wang Y, et al., 2022. LRP2A: layer-wise relevance propagation based adversarial attacking for graph neural networks. *Knowl-Based Syst*, 256:109830.

Madaan D, Shin J, Hwang SJ, 2020. Adversarial neural pruning with latent vulnerability suppression. Proc 37th Int Conf on Machine Learning, Article 610.

Madry A, Makelov A, Schmidt L, et al., 2018. Towards deep learning models resistant to adversarial attacks. Proc 6th Int Conf on Learning Representations.

Moosavi-Dezfooli SM, Fawzi A, Frossard P, 2016. DeepFool: a simple and accurate method to fool deep neural networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2574-2582. https://doi.org/10.1109/CVPR.2016.282

Pang TY, Xu K, Du C, et al., 2019. Improving adversarial robustness via promoting ensemble diversity. https://arxiv.org/abs/1901.08846

Papernot N, McDaniel P, Wu X, et al., 2016a. Distillation as a defense to adversarial perturbations against deep neural networks. Proc IEEE Symp on Security and Privacy, p.582-597. https://doi.org/10.1109/SP.2016.41

Papernot N, McDaniel P, Jha S, et al., 2016b. The limitations of deep learning in adversarial settings. Proc IEEE European Symp on Security and Privacy, p.372-387. https://doi.org/10.1109/EuroSP.2016.36

Shafahi A, Najibi M, Ghiasi A, et al., 2019. Adversarial training for free! Proc 33rd Int Conf on Neural Information Processing Systems, Article 302.

Szegedy C, Zaremba W, Sutskever I, et al., 2014. Intriguing properties of neural networks. Proc Int Conf on Learning Representations.

Tramer F, Kurakin A, Papernot N, et al., 2018. Ensemble adversarial training: attacks and defenses. Proc 6th Int Conf on Learning Representations.

Vivek BS, Babu RV, 2020. Single-step adversarial training with dropout scheduling. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.947-956. https://doi.org/10.1109/CVPR42600.2020.00103

Wang ZB, Guo HC, Zhang ZF, et al., 2021. Feature importance-aware transferable adversarial attacks. Proc IEEE/CVF Int Conf on Computer Vision, p.7639-7648. https://doi.org/10.1109/ICCV48922.2021.00754

Wei XX, Liang SY, Chen N, et al., 2019. Transferable adversarial attacks for image and video object detection. Proc 28th Int Joint Conf on Artificial Intelligence, p.954-960.

Wong E, Rice L, Kolter JZ, 2020. Fast is better than free: revisiting adversarial training. Proc 8th Int Conf on Learning Representations.

Yamamura K, Sato H, Tateiwa N, et al., 2022. Diversified adversarial attacks based on conjugate gradient method. Proc 39th Int Conf on Machine Learning, p.24872-24894.

Yang HR, Zhang JY, Dong HL, et al., 2020. DVERGE: diversifying vulnerabilities for enhanced robust generation of ensembles. Proc 34th Int Conf on Neural Information Processing Systems, Article 462.

Zhang JB, Qian WH, Nie RC, et al., 2022. LP-BFGS attack: an adversarial attack based on the Hessian with limited pixels. https://arxiv.org/abs/2210.15446

Zhang JF, Xu XL, Han B, et al., 2020. Attacks which do not kill training make adversarial learning stronger. Proc 27th Int Conf on Machine Learning, Article 1046.

Zheng HZ, Zhang ZQ, Gu JC, et al., 2020. Efficient adversarial training with transferable adversarial examples. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.1178-1187. https://doi.org/10.1109/CVPR42600.2020.00126

Zhu C, Huang WR, Li HD, et al., 2019. Transferable clean-label poisoning attacks on deep neural nets. Proc 36th Int Conf on Machine Learning, p.7614-7623.