Intelligent Systems with Applications 19 (2023) 200258

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

ZeroGrad: Costless conscious remedies for catastrophic overfitting in the FGSM adversarial training

Zeinab Golgooni, Mehrdad Saberi [1], Masih Eskandar [1], Mohammad Hossein Rohban *

Department of Computer Engineering, Sharif University of Technology, Azadi Avenue, Tehran, Iran

ARTICLE INFO

Keywords:
Deep learning
Fast adversarial training
Adversarial robustness
Catastrophic overfitting
FGSM training

ABSTRACT

Vulnerability of deep neural networks to small adversarial examples has recently attracted a lot of attention. As a result, making models robust to small adversarial noises has been sought in many safety critical applications. Adversarial training through iterative projected gradient descent (PGD) has been established as one of the mainstream ideas to achieve this goal. However, PGD is computationally demanding and often prohibitive in the case of large datasets and models. For this reason, the single-step PGD, also known as the Fast Gradient Sign Method (FGSM), has recently gained interest in the field. Unfortunately, FGSM-training leads to a phenomenon called "catastrophic overfitting," which is a sudden drop in the test adversarial accuracy under the PGD attack. In this paper, we propose new methods to prevent this failure mode of the FGSM-based attacks with almost no extra computational cost. The proposed methods are also backed up with theoretical insights into the causes of the catastrophic overfitting. Our intuition is that small input gradients play a key role in this phenomenon. The signs of such gradients are quite unstable and fragile from an epoch to the next, making the signed gradient method discontinuous along the training process. These instabilities introduce large weight updates by the stochastic gradient descent, and hence potentially cause overfitting. To mitigate this issue, we propose to simply identify such gradients and make them zero prior to taking the sign in the FGSM attack calculation that is used in the training. This remedy makes the training perturbations stable, while almost preserving the adversarial property of such perturbations. The idea while being simple and efficient, achieves competitive adversarial accuracy on various datasets and can be used as an affordable method to train robust deep neural networks[2].

1. Introduction

Despite deep neural networks impressive success in many real-world problems, their instability under test-time adversarial noises is the major issue against their use in the safety critical applications Filipovich and Kovalev (2023), Vitorino et al. (2023), Tian et al. (2021), Liu et al. (2023), Xu et al. (2023). Training the model based on the adversarial samples in each mini-batch, which is known as "Adversarial training" (AT) Madry et al. (2018), has been empirically established as a general and effective approach to remedy this issue. Multi-step gradient-based maximization of the loss function with respect to the norm-bounded perturbations is often used to craft the adversarial samples in the AT. This method is known as the Projected Gradient Descent (PGD) attack and results in a reasonable model generalization under many strong norm bounded attacks Carlini and Wagner (2017). The PGD iterative process, however, makes crafting the adversarial perturbations very slow, and sometimes infeasible in case of large datasets and models. Fast Gradient Sign Method (FGSM) Goodfellow et al. (2015), which is the single-step PGD, is much faster but results in a poor model generalization under the stronger test-time multi-step PGD attack.

Recently, few attempts have been made to address the generalization issues of the FGSM Wong et al. (2020). "Fast," introduced FGSM-RS, an extended version of FGSM in which a random uniform noise is added to the input sample prior to applying FGSM. Although this method can noticeably improve vanilla FGSM-trained network adversarial accuracy, there is still a large gap with the PGD-training performance, while also suffering from a mysterious phenomenon called "catastrophic overfitting" Wong et al. (2020). This refers to a sudden

* Corresponding author.
E-mail addresses: golgooni@ce.sharif.edu (Z. Golgooni), mehrdads@ce.sharif.edu (M. Saberi), gnomy@ce.sharif.edu (M. Eskandar), rohban@sharif.edu (M.H. Rohban).

[1] Equal contribution.

[2] Code is available at https://github.com/rohban-lab/catastrophic_overfitting.

https://doi.org/10.1016/j.iswa.2023.200258
Received 21 November 2022; Received in revised form 10 June 2023; Accepted 12 July 2023
Available online 18 July 2023

and drastic drop of the adversarial test accuracy, under a stronger PGD attack, at a single epoch, although the training robust accuracy under the FGSM attack continues to go up. That is, a large gap between FGSM and PGD losses can be seen after this epoch. Although early-stopping could be used to obtain a version of the model that is not overfitted, the test accuracy of clean samples are sacrificed in such a remedy. This is mainly because catastrophic overfitting happens way too early in the training. Therefore, for the FGSM training to be competitive with PGD training, one has to address catastrophic overfitting effectively.

Following this track, understanding of the catastrophic overfitting, and improving the FGSM training has recently gained attention Li et al. (2020), Kim et al. (2021), Vivek and Babu (2020b, 2020a), Andriushchenko and Flammarion (2020), de Jorge et al. (2022), Kang and Moosavi-Dezfooli (2021). Currently, GradAlign Andriushchenko and Flammarion (2020) stands out among all these solutions, and explanations for the catastrophic overfitting issue of the FGSM. This method regularizes the model weights during training to make the input gradients more aligned with each other within the set of allowed perturbed inputs. The primary insight for this remedy is that the FGSM training does best when the loss behaves linearly as a function of the input. Although being effective in solving the issue, the computational cost of the regularization step is considerable, contrary to the fundamental goal of the single-step methods.

In this paper, we propose an uncomplicated effective method based on FGSM that prevents catastrophic overfitting and achieves comparable results to GradAlign but with negligible extra training computational cost. Comparison of the results of different methods Fig. 1a shows that by incurring almost no computational overhead, our proposed method, called ZeroGrad, achieves significantly better adversarial accuracy in the CIFAR-10 dataset.

In crafting the FGSM attacks, we suggest zeroing out the input gradient elements whose absolute values are below a certain threshold, before feeding the input gradient into the sign function in FGSM. In Fig. 1b, We illustrate an extreme case where the input gradient, denoted by ∇_x, is close to zero along the vertical axis, and as a result of the sign discontinuity, the FGSM attack (X'_{FGSM}) lies distant away from the input gradient. This could cause attack instability if sign of this small component of the input gradient changes in two consecutive epochs (compare top and bottom plots). On the other hand, our proposed attack, X'_{zeroGrad} stays constant in such cases. We will discuss the relationship between such instability and catastrophic overfitting from an empirical perspective in the next sections. We observe that such simple remedies prevent catastrophic overfitting and result in competitive robust accuracies.

In summary, we hypothesize and support that tiny fragile input gradients play a key role in catastrophic overfitting. Our intuition is that due to the discontinuity of the "sign" function in FGSM, elements of the input gradient that are close to zero could cause unexpected drastic change in the attack and result in large weight updates in two consecutive epochs. This tends to take place later in the training, as it has been empirically observed that adversarially robust models yield sparser input gradients, resulting in many close to zero gradient elements later in the training. In addition, we notice that based on our theoretical insight about large sudden weight updates, regularizing second derivative of the loss, which is implicitly done in GradAlign, could help to avoid overfitting. This also highlights the generality of our insights.

Our contributions are summarized below:

- Proposing a computationally inexpensive method to avoid catastrophic overfitting;
- A more comprehensive explanation of catastrophic overfitting through associating the weight update with the input gradient magnitude;
- Competitive adversarial accuracy on various datasets under no significant train-time overhead.

In this paper, after a review of the related work, we first introduce our proposed method, ZeroGrad. Next, we provide our explanation about causes of catastrophic overftitting with theoretical insights and also numerical observations. In the experiments section, we report our evaluation of methods on various datasets. Finally, we discuss some additional points including issues about hyperparameters in section of ablation study. Supplemantary materials including further analysis of hyperparameters and suggested alternative approaches are provided in appendices.

2. Related work

Wong et al. (2020) has brought back hope to have affordable and efficient methods of adversarial training based on a single-step attack, called FGSM-RS. It also introduced the special failure mode, catastrophic overfitting, as a critical obstacle to achieve results comparable to the PGD-training. As shown in Fig. 2, during training of FGSM-RS, the adversarial accuracy drastically drops to near zero. Also, the input gradients of images are changed significantly and seem more scattered. In other words, the training fails to achieve a robust acceptable model. Some subsequent works try to understand catastrophic overfitting, and improve the FGSM training.

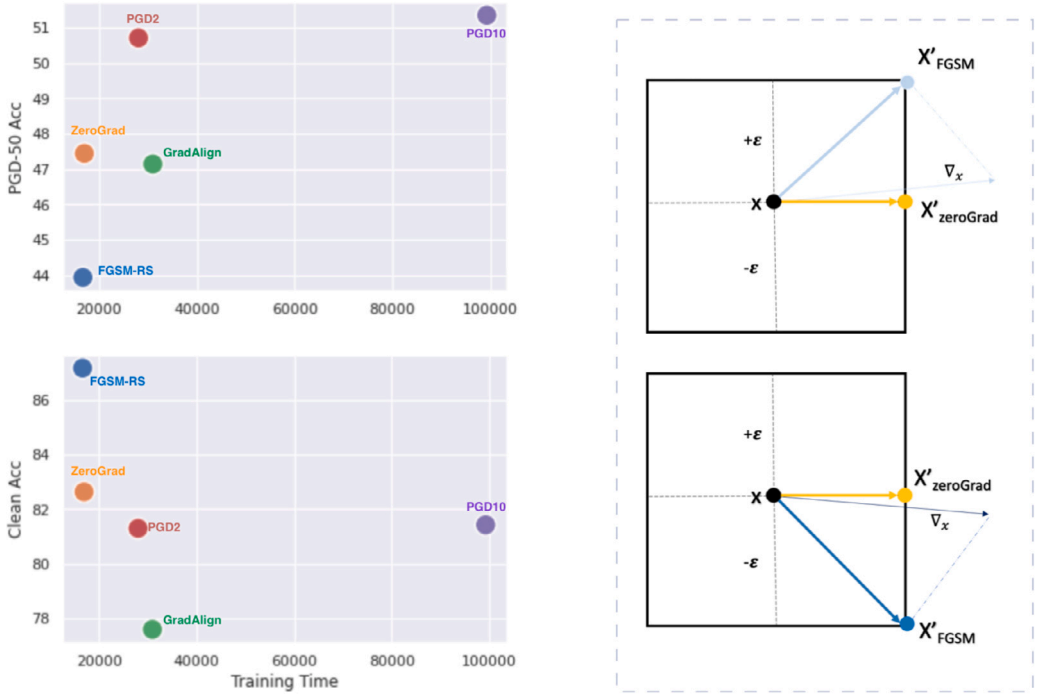
In the beginning, this phenomenon was introduced as a failure mode of FGSM training Wong et al. (2020), but it later became clear that FGSM-RS and generally many other single-step adversarial training methods also have this vulnerability Andriushchenko and Flammarion (2020). At a first look, this event can be seen as overfitting to the FGSM attack due to the attack weakness Wong et al. (2020), but the suddenness of this failure warranted the need for deeper explanations.

Li et al. (2020) noticed that by inspecting the test robust accuracy at the mini-batch resolution, catastrophic overfitting happens even in successful runs of FGSM training, where a sudden accuracy drop is not observed at the end of each epoch, and the final model is robust. They hypothesized that the initial random noise in FGSM-RS serves as a way to recover from such drops in the model robustness, but because of the noise randomness, this recovery may fail with a non-zero probability. According to this hypothesis, they proposed to use PGD training as a better recovery mechanism whenever overfitting happens. They were able to mitigate the issue with a 2-3 times training slowdown compared to the FGSM training. This method does not take a step toward explaining why the overfitting happens. Another downside is that the attacks like PGD are multi-step and cannot be parallelized to run as fast as single-step attacks on current computational units.

As another solution, Vivek and Babu (2020a) proposed to regularize the difference in the network logits for the FGSM and iterative-FGSM perturbed inputs, and force the difference to be close to zero in a few mini-batches. They have empirically shown that such a regularization could prevent gradient masking.

On the other hand, Kim et al. (2021) takes another perspective and points out that due to FGSM perturbations always lying on the boundaries of the ℓ_∞ ball, the network loss faces the so-called "boundary distortion." This distortion makes the loss surface highly curved, and consequently results in the model becoming non-robust against smaller perturbations. Therefore, they proposed to evaluate the loss at various multiples of the FGSM direction, where multiples are between 0 and 1 and use the smallest multiple that results in misclassification, in the training attacks. Their proposed method, CKPT, main hyperparameter adjusts the number of intermediate points to check; Larger values cause longer training time.

de Jorge et al. (2022) revisits the role of noise in FGSM-based adversarial training. They proposed a method called NFGSM, to prevent CO by increasing the amount of noise and also not limiting the perturbation to be smaller equal than ϵ. Although the idea seems simple it is not supported with any considerable theoretic justification Unfortunately, it is not supported with any considerable justification. Unfortunately, they



(a) (b)

Fig. 1. a) Performance and complexity overhead of methods. Clean and robust accuracy of methods with respect to training time are shown in the left and right plots respectively. Each method was evaluated by training a WideResNet network with a width factor of 10 for 51 epochs using a piecewise learning rate schedule with a learning rate drop at epoch 50. We made an exception for GradAlign and instead trained for 30 epochs using a cyclic learning rate schedule since the method seems to perform significantly better under those circumstances. **b)** Visualization of our proposed method, ZeroGrad, v.s. FGSM for two simplified 2-dimensional examples. Due to discontinuity of "sign" function, FGSM is vulnerable to small elements of the gradient. Gradients of these two samples are very similar but a tiny difference leads to a completely different adversarial example by FGSM, which can cause large weight update in one step. ZeroGrad tries to solve this instability by zeroing out tiny fragile gradients for crafting adversarial examples.

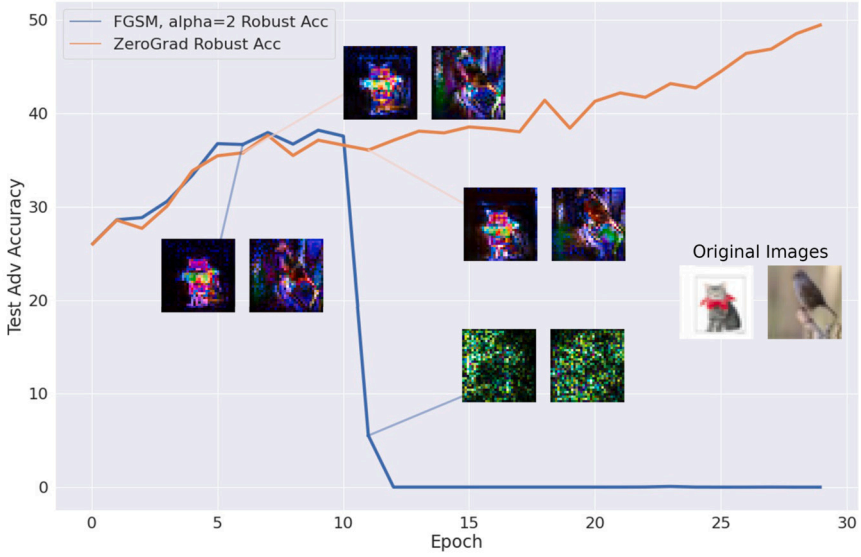


Fig. 2. Training diagram of the adversarial training using ZeroGrad (Q = 0.35) and FGSM-RS ($\alpha = 2$) along with input gradients of two images on epochs 7 and 12, respectively (before and after the catastrophic overfitting). The input gradient changes dramatically during FGSM-RS training, and looks less spatially compact and interpretable to the the human eye after the catastrophic overfitting occurs and the robust accuracy plummets. The input gradient during training with ZeroGrad does not change drastically and robust accuracy continues to rise, as the catastrophic overfitting is prevented.

do not suggest any explanation for the role of randomness in FGSM-based methods.

Among the most insightful attempts, GradAlign Andriushchenko and Flammarion (2020) hypothesized that large variations of the input gradient around a sample cause the FGSM direction to be noisy and irrelevant, and consequently results in poor attack quality and overfitting. They backed up their hypothesis by theoretically showing that the angle between local input gradients is upper-bounded prior to the training for an appropriate Gaussian weight initialization. This partly explains why the FGSM training does well early in the training, and it takes a few epochs before catastrophic overfitting occurs. However, this explanation does not fully characterize this phenomenon, e.g. why such drops in the robust generalization occur so quickly. Based on this explanation, they proposed to regularize the optimization by constraining such variations to be small. This proved to be effective in a lot of cases where FGSM fails. However, the mentioned regularization requires a sequential procedure called "double-backpropagation," and increases the training time by a factor of 2-3 compared to the vanilla FGSM-training.

In Fig. 1a, we provide the comparison of the training times of different methods. Due to the fact that the impracticality of adversarial training arises more in large-scale problems with complex models, here, we report the results of training a WideRestNet network. It is clear that the PGD-10, as the representative of multi-step methods, can achieve better adversarial accuracy but with a much higher computational cost. Although FGSM-RS has improved the vanilla FGSM, there is still a large gap between its accuracy and that of the PGD-10. Furthermore, this method suffers from catastrophic overfitting, in which case the training has to be stopped prematurely that sacrifices the clean and adversarial accuracies. As shown in the figure, successful remedies for the FGSM training, such as GradAlign, Andriushchenko and Flammarion (2020) improve the adversarial accuracy at the cost of increasing the training time compared to FGSM-RS significantly. However, our proposed method, ZeroGrad, achieves the around the same adversarial accuracy of GradAlign, but with almost no extra computational cost compared to the FGSM-RS.

To summarize, some of current works have suggested variations of training by adding more randomness or similar modifications to prevent the failure modes of the FGSM training, but the rationale is not quite clear in such works, and new failure modes, such as catastrophic overfitting, arise. One has to note that this new issue cannot be effectively dealt with using simple solutions such as early stopping, as the adversarial and clean accuracies do not often reach close to those of the PGD training. Some other researches have built on top of FGSM, and have suggested some remedies for the catastrohpic overfitting. However, despite improving the adversarial accuracy, such methods result in significant extra computational costs that contradicts the goal of fast training in the FGSM training.

3. Proposed method

As stateted previously, Fast Wong et al. (2020) showed that combining FGSM adversarial training with random initialization can result in a simple method that is just effective as the PGD-based adversarial training with a low training cost. However, identification of the catastrophic overfitting as a failure mode, which is also the main vulnerability of the FGSM adversarial training, highlights the need for comprehensive understanding of this failure mode, and developing new methods that could solve this issue and simultaneously achieve acceptable robustness with a low cost.

We suggest that there are certain elements in the input gradient, whose magnitudes are susceptible to change their sign, which cause general training instability that could lead to a massive sudden drop in the robust accuracy during training. This susceptibility could for example stem from such elements being close to zero. We call these critical elements, tiny fragile gradients. Our proposed method, avoids occurrence of catastrophic overfitting by explicitly zeroing these tiny gradients. In Fig. 1b, our proposed method ZeroGrad is schematically illustrated and compared against the FGSM training. Discontinuity of the "sign" function that is part of the FGSM, makes this method unstable. Gradients of two given samples in Fig. 1b are very similar but a tiny difference in one dimension leads to a completely different FGSM adversarial example. These small elements can also cause large weight updates in a training iteration as a result of dramatic changes in the training adversarial samples. ZeroGrad tries to solve this instability by zeroing out tiny fragile gradients in crafting adversarial examples.

Algorithm 1 ZeroGrad.

Input: number of epochs T, maximum perturbation ϵ, quantile value q, step size α, dataset of size M, network f_θ
Output: Robust network f_θ
for $t = 1$ **to** T **do**
  for $i = 1$ **to** M **do**
    $\delta \sim \text{Uniform}([-\epsilon, \epsilon]^d)$
    $\nabla_\delta \leftarrow \nabla_\delta \ell(f_\theta(x_i + \delta), y_i)$
    $\nabla_\delta[\nabla_\delta < \text{quantile}(|\nabla_\delta|, q)] \leftarrow 0$
    $\delta \leftarrow \delta + \alpha \cdot \text{sgn}(\nabla_\delta)$
    $\delta \leftarrow \max(\min(\delta, \epsilon), -\epsilon)$
    // Update network parameters with some optimizer
    $\theta \leftarrow \theta - \nabla_\theta \ell(f_\theta(x_i + \delta), y_i)$
  end for
end for

Next, we describe our proposed method in more details, and possible solutions to identify the fragile gradients.

3.1. ZeroGrad

Let $f(x, W)$ represent a classification hypothesis (e.g. a deep neural network) with adjustable parameters W, input x, and $g(x, W) := \ell(f(x, W), y)$ be the loss function, with y as the ground-truth label that is used in training (e.g. the cross-entropy loss). Further, let δ be the perturbation that is designed based on FGSM. We propose to take $\bar{\delta} := \delta \odot \mathbb{I}(|\nabla_x g| \geq q)$, where $\mathbb{I}(.)$ is the indicator function, $\odot$ is the element-wise product, and t is a threshold. To make selection of t easier, we propose to adapt t to the sample and use the lower q-quantile of the absolute value of the input gradient elements for each sample. We note that as long as t is small, the loss function would not change too much compared to the original FGSM attack:

$$\begin{aligned} g(x + \bar{\delta}) &\approx g(x) + \bar{\delta}^\top \nabla_x g \\ &= g(x) + \delta^\top \nabla_x g - \epsilon \sum_{|[\nabla_x g]_i| \leq t} |[\nabla_x g]_i| \\ &\geq \underbrace{g(x) + \delta^\top \nabla_x g}_{\approx g(x+\delta)} - \epsilon d q t, \end{aligned} \tag{1}$$

where d is the input dimension, and t is assumed to be the quantile threshold as described earlier. Therefore, zeroing the gradient elements does not totally harm the adversarial property of the new perturbation $\bar{\delta}$ as long as the threshold t is small. By tuning the threshold and zeroing out the lower gradients, we are able to avoid catastrophic overfitting altogether without sacrificing too much accuracy as shown in section 5. The pseudocode for this method is shown in Algorithm 1.

4. Causes of catastrophic overfitting

In this section, we provide some theoretical insights into why catastrophic overfitting happens, and why it is sudden and is often observed in later epochs of training. We then back up our hypothesis with some numerical results.

4.1. Theoretical insights

In making a model adversarially robust, the following optimization should ideally be solved:

$$\min_{W} \mathbb{E}_{x,y} \max_{\delta \in \Delta} \ell(f(x+\delta, W), y). \tag{2}$$

Let $\eta_i = \epsilon \operatorname{sgn} \nabla_x g(x_i, W)$ be the FGSM attack on the data point x_i. So $x_i' = x_i + \alpha \eta_i$ represents the FGSM-based adversarial example, where α is the FGSM step size.

We consider a variant of FGSM training, in which the inner maximization of the Eq. (2), is approximated by the FGSM attack, and the expectation is estimated by the empirical mean. This leads to the following optimizing:

$$\min_{W} \sum_i g(x_i', W) \tag{3}$$

A necessary condition for the optimizer of this loss in the local minimum is the following:

$$\begin{aligned} &\nabla_W \sum_i g(x_i', W) \\ &\quad = \sum_i \alpha \left\{ \nabla_W x_i' \right\} \Big|_{(x_i, W)} \cdot \left\{ \nabla_x g \right\} \Big|_{(x_i', W)} \\ &\qquad + \nabla_W g \Big|_{(x_i', W)} \\ &\quad = \mathbf{0} \end{aligned} \tag{4}$$

But note that,

$$\nabla_W x_i' = \nabla_W \eta_i = \epsilon \left\{ \nabla_{W,x} g \right\} \cdot \left\{ \operatorname{diag}(\delta(\nabla_x g)) \right\} \Big|_{(x_i, W)}, \tag{5}$$

where $\delta(.)$ is the Dirac delta function. Assume that for the i-th training sample, there exists some element j such that:

- (A1): $[\nabla_x g]_j \big|_{(x_i, W)} = 0$,
- (A2): $[\nabla_x g]_j \big|_{(x_i', W)} \neq 0$, and
- (A3) : $[\nabla_{W,x} g]^{(j)} \big|_{(x_i, W)} \neq \mathbf{0}$,

where $[a]_j$ and $[B]^{(j)}$ denote the j-th element of the vector a, and the j-th column of the matrix B, respectively. Also, '.' stands for the inner product operator. Under (A1), the j-th column of $\nabla_W \eta_i$, denoted as $[\nabla_W \eta_i]^{(j)}$, would be an ∞ multiple of a non-zero vector ($[\nabla_{W,x} g]^{(j)}$), according to the Eq. (5), assumption (A3), and due to the delta function being infinite at zero. In addition, note that the first term in left-hand side of Eq. (4) is a weighted summation of column vectors, each with the size of weights:

$$\left\{ \nabla_W \eta_i \right\} \Big|_{(x_i, W)} \cdot \left\{ \nabla_x g \right\} \Big|_{(x_i', W)} = \sum_k [\nabla_W \eta_i]^{(k)} [\nabla_x g]_k \tag{6}$$

Therefore, as $[\nabla_W \eta_i]^{(j)}$ is an ∞ multiple of a non-zero vector and $[\nabla_x g]_k \neq 0$, according to (A2), the summation in Eq. (6) would contain a multiple of ∞. The term $\nabla_W g$ is the weight update that is made during FGSM adversarial training, and according to the Eq. (4) is negative of the weighted sum in Eq. (6). Therefore, once getting close to a local minimum in the weight space, the network may experience a huge weight update, which corresponds to the mentioned ∞ multiple of $[\nabla_{W,x} g]^{(j)}$ at x_i.

A simple way to break (A1) and (A2) is to force the input gradients change smoothly, which is implicitly achieved in GradAlign. Properly clipping the gradient updates of the network's weights might also be a way to prevent large weight updates, and hence to avoid catastrophic overfitting. Although we do not investigate gradient clipping in our work, we include some initial results in Appendix C for future research. The other possibility to mitigate the issue is to zero out tiny input gradients so that the training would not encounter large weight updates close to the local optima. Also, motivated by this explanation, we could design the attack based on various random starts and zeroing elements of the attack that are changing across various random starts. This helps to ensure that η_i would probably not change too much after the weight update, that is, $\nabla_W \eta_i$ would probably remain small. We introduce this method in more detail in Appendix C.

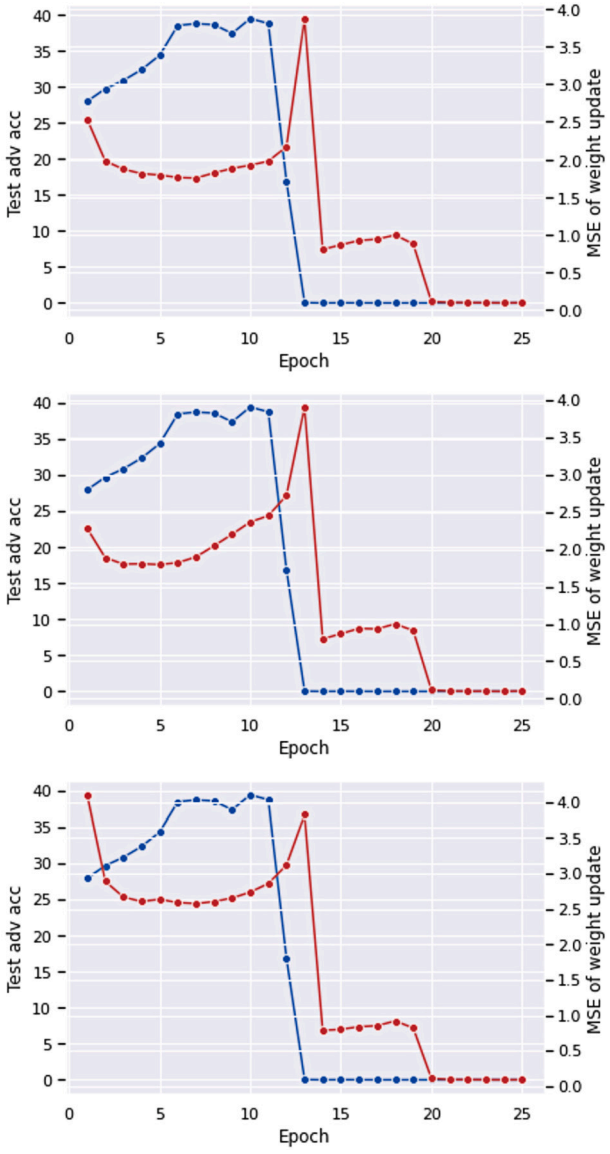


Fig. 3. MSE difference of weights, related to three kernels in the convolution layers of the model, before and after updating the model in each epoch (red), and the test adversarial accuracy of the model (blue). The model is trained on the CIFAR-10 dataset with batch size = 128, ϵ = 8/255, and FGSM-alpha = 2.

We will next empirically validate these insights by some experiments on the CIFAR-10 dataset.

4.2. Numerical results

Motivated by the explained insights, we tried to observe the trend of a few statistics during the training of a Preact ResNet-18 model on CIFAR-10, specifically exactly at the step when the catastrophic overfitting happens. Here, we use the same default setting for training that is provided in Wong et al. (2020). We observe the ℓ_2 norm of difference in weights of the model before and after updating the model in an epoch. Interestingly, a huge significant change in weights is seen exactly at the same epoch that catastrophic overfitting happens. Fig. 3 displays three elements of the weight update of the model, indicating a huge weight update exactly simultaneous to the drop of test adversarial accuracy of the model.

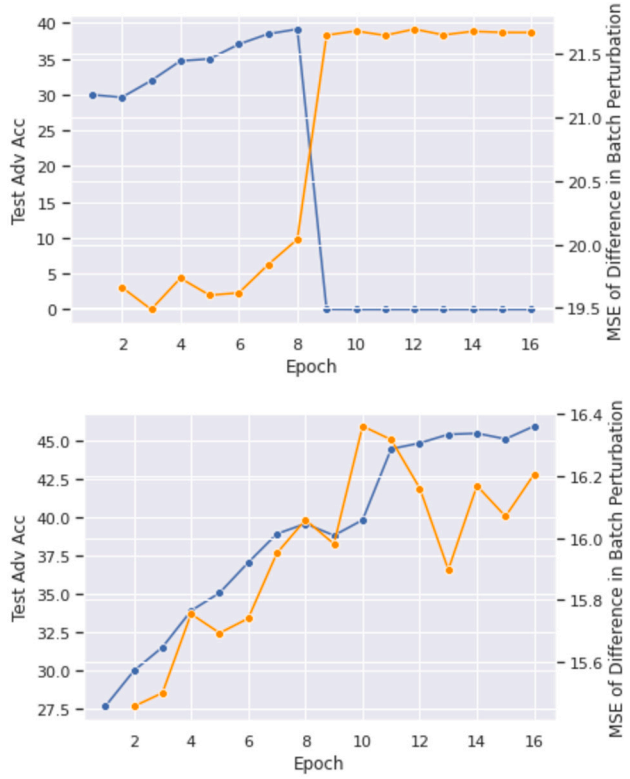


Fig. 4. MSE based on the difference in batch perturbations of each two consecutive epochs (orange), and test adversarial accuracy of the model (blue). The top diagram refers to training by FGSM-RS. The bottom diagram refers to training by the method that is similar to FGSM-RS, but with zeroing 35% lower quantile of absolute gradients. The model is trained on the CIFAR-10 dataset with batch size $= 128$, $\epsilon = 8/255$, and FGSM-alpha $= 2$.

Another observation is the significant change in the FGSM perturbations of a mini-batch before and after catastrophic overfitting, which is shown at top of Fig. 4. The jump in the perturbation difference is in accordance with our hypothesis that once the FGSM-trained model gets close to the convergence, the derivative of FGSM perturbations could increase sharply. Due to the idea that tiny elements of the gradient bring about the drastic change of the model, we try to verify that whether ignoring small gradients in FGSM could change this observation or not. To achieve this, we test the difference between FGSM perturbations of a mini-batch in two consecutive epochs when zeroing tiny elements of the input gradient. Specifically, we explicitly set the lower 35% quantile of the absolute input gradients to zero, and acquired the perturbation difference by this modified gradient. MSE based on the difference of perturbations of the same mini-batch in this setting is shown at the bottom of Fig. 4. Clearly, in this case, the drastic change of perturbations is prevented. This intuition can help us to propose our new single-step training attacks that are potentially safe from the catastrophic overfitting.

5. Experiments

In this section, we demonstrate the effectiveness of our proposed method on CIFAR-10, CIFAR-100, and SVHN datasets. We compare our proposed method, ZeroGrad, with the Fast (FGSM-RS), and GradAlign, based on the standard test accuracy, and robust test accuracy. For each method, the standard deviation and the average of the test accuracies are calculated by training the models using two random seeds. For training with FGSM-RS, the step size α is tuned seperately in every setting, to both prevent overfitting and have its maximum possible value to achieve better adversarial accuracy.

Test-time Attacks and models. To report the robust accuracy of our models on the test data, we attack the models with the PGD adversarial attack with 50 steps, 10 random restarts, and step size $\alpha = 2/255$. We also evaluate our methods based on AutoAttack Croce and Hein (2020) to make sure that our methods do not suffer from the gradient obfuscation.

The network that is used for training is Preact ResNet-18 He et al. (2016). The results for training with WideResNet-34 are also available in the Appendix D.4.1, which have better accuracies compared to training with Preact ResNet-18, but the training is much slower.

Learning rate schedules. There are two types of learning rate schedules that are used for our experiments. The first one is the *cyclical learning rate schedule* Smith (2017), which helps us get a faster convergence with a fewer number of epochs. We set the cyclical learning rate schedule that reach its maximum learning rate when half of the epochs are passed. The other one is the *one-drop learning rate schedule*, which starts with a fixed learning rate, and decreases it by a factor of 10 in the last few epochs. For example, if the model is trained for 52 epochs and the initial learning rate is 0.1, we drop the learning rate to 0.01 in the 50-th epoch. The reason for stopping the training a few epochs after the drop of the learning rate is to prevent the normal overfitting of the model to the training data Rice et al. (2020). If the training continues after the learning rate drops, the robust training loss keeps on decreasing but the robust test loss would also increase. Note that if the training continues after the learning rate drops, catastrophic overfitting does not happen using ZeroGrad with suitable q. But it results in worse robust test accuracy due to the general overfitting in adversarial training, which is different from catastrophic overfitting. The one-drop learning rate is not used in the experiments reported in this section. But results with this learning rate schedule are provided in the Appendix D.2, as it enables the model to achieve better accuracy if it is trained for more epochs.

Setup for our proposed method. For the ZeroGrad method, similar to FGSM-RS, we add a uniformly random noise in $[-\epsilon, \epsilon]$, to the samples prior to the attack. We also use different values of q based on the dataset, size of the network, and ϵ. But to avoid the criticisms of small effective attack norm being the main cause of catastrophic overfitting prevention Andriushchenko and Flammarion (2020), we always use the FGSM step size $\alpha = 2.0$, which is the maximum possible step size for this method. More analysis on hyperparameters of our method are available in next sections.

5.1. CIFAR-10 results

For the CIFAR-10 dataset, we use the cyclic learning rate with a maximum learning rate of 0.2 and 30 epochs to be able to compare the results of our methods to the accuracies that are reported for the previously proposed methods. The results for maximum perturbation size $\epsilon = 8/255$ are shown in Table 1. For this dataset, we also include the results of NFGSM de Jorge et al. (2022) and CKPT Kim et al. (2021) as two other recent proposed single-step methods. It seems that larger perturbation sizes like $\epsilon = 16/255$, are not valid for the CIFAR-10 dataset, and can completely change labels of the perturbed images Tramèr et al. (2020). However, we also investigated our methods on $\epsilon = 16/255$ to see how it performs (see Appendix D.1). Note that our methods can be trained for a higher number of epochs. For example, with appropriate settings, we can train our models for 200 epochs with the piecewise learning rate schedule without encountering catastrophic overfitting. The results for training with more epochs, which lead to better accuracies are available in Appendix D.2.

We also report performance of the models that are trained by our proposed methods against the AutoAttack Croce and Hein (2020) as a standard robustness evaluation tool.

Table 1
Standard and PGD-50 accuracy on the CIFAR-10 dataset with $\epsilon = 8/255$ for different training methods. All models are trained with the cyclical learning rate schedule for 30 epochs.

Method	Standard Acc.	PGD-50 Acc.	Autoattack Acc.
ZeroGrad (q = 0.35)	81.61± 0.24	47.55± 0.05	44.02± 0.52
FGSM-RS ($\alpha = 1.25$)	84.32± 0.08	45.10± 0.56	43.47± 0.26
FGSM-CKPT (c = 3)	**87.7± 0.08**	33.9± 2.3	32.3± 2.2
N-FGSM (k = 2 ϵ)	80.58± 0.22	**48.12± 0.07**	44.36± 0.34
GradAlign	81.00± 0.37	47.58± 0.24	**44.57± 0.03**
PGD-2	82.15± 0.48	48.43± 0.40	46.24± 0.02
PGD-10	81.88± 0.37	**50.04± 0.79**	**47.18± 0.03**

Table 2
Standard and PGD-50 accuracy on the CIFAR-100 dataset with $\epsilon = 8/255$ for different training methods. All models are trained with the cyclical learning rate schedule for 30 epochs.

Method	Standard Acc.	PGD-50 Acc.
ZeroGrad (q = 0.45)	53.70± 0.23	**25.08± 0.07**
FGSM-RS ($\alpha = 1.25$)	49.33± 0.57	0.00± 0.00
FGSM-RS ($\alpha = 1.0$)	**56.94± 0.25**	23.78± 0.41
GradAlign	51.92± 0.18	24.52± 0.10
PGD-2	52.45± 0.12	26.72± 0.02
PGD-10	51.29± 0.30	**26.79± 0.14**

5.2. CIFAR-100 results

To train our models on the CIFAR-100 dataset, we again use the cyclic learning rate with a maximum learning rate of 0.2 and 30 epochs of training. The models are trained with maximum perturbation size $\epsilon = 8/255$. As can be seen in Table 2, our method is able to outperform other models that are trained using single-step adversarial attacks. We keep in mind that although the difference between the robust accuracy of our methods and the GradAlign method might not be much, the main contribution of our methods is that they provide simplicity, speed, and high robust accuracy at the same time.

5.3. SVHN results

For the SVHN dataset we encountered slightly different results compared to other datasets. We observed that the SVHN dataset requires a significantly larger zeroing threshold, i.e. 0.7, for ZeroGrad to successfully prevent catastrophic overfitting. This large value of q, in turn, leads to a weaker robust training and a lower evaluation accuracy as is shown in Table 3. This weakness is due to the premise that adversarial properties of the ZeroGrad attacks hold if q is small (see Eq. (1)).

The cause of this phenomenon remains unexplained to us. However, we hypothesize that this may be due to the peculiar nature of the SVHN dataset, where a large portion of the image (i.e. the background), is often without any information about the label, and is unrelated to the object that is being classified (i.e. the number). This may cause many small but yet unnecessarily *fragile* or *problematic* gradients to appear, shifting the minimum required zeroing threshold.

This hypothesis is somewhat supported by the mostly unhindered performance of the MultiGrad method on the SVHN dataset. MultiGrad is similar to ZeroGrad in the sense that it attempts to zero out *fragile* gradients, but uses a criterion other than magnitude for finding these *fragile* gradients. The details of the MultiGrad method and our experiments can be found in the Appendix C.2.

5.4. ImageNet results

For the ImageNet dataset, we used the network Resnet-50. We mainly followed the settings from Wong et al. (2020). The 15 epochs of training is divided into three phase, adjusting the hyperparameters including the learning rate, size of image clipping and batch size. Due to the high computational cost of training and testing on the ImageNet, we only train our proposed method and test with PGD-50 with 1 restart. We adopt the results of FGSM from Wong et al. (2020) for better comparison. The standard accuracy and robustness against PGD-50 with one restart, for maximum perturbation sizes $\epsilon = 2/255$ and $\epsilon = 4/255$ is reported in Table 4. We observed that during the training of ZeroGrad, CO did not happen and also can achieve acceptable robustness.

Table 3
Standard and PGD-50 accuracy on the SVHN dataset with $\epsilon = 8/255$ for different training methods.

Method	Standard Acc.	PGD-50 Acc.
ZeroGrad (q = 0.7)	88.36 ± 0.63	39.42 ± 1.92
FGSM-RS ($\alpha = 0.875$)	92.25± 0.01	38.73± 0.07
GradAlign	**92.36± 0.47**	**42.08± 0.25**
PGD-2	**92.68± 0.45**	47.28± 0.26
PGD-10	91.92± 0.40	**52.08± 0.49**

Table 4
Standard and PGD-50-1 accuracy on the ImageNet dataset for different.

Method	Epsilon	Standard Acc.	PGD-50 Acc.
ZeroGrad (q = 0.2)	2/255	60.78	42.95
FGSM	2/255	60.90	43.46
ZeroGrad (q = 0.2)	4/255	54.80	30.69
FGSM	4/255	55.45	30.28

It is worth noting that, we have not tuned our quantile value, q, or other hyperparameters for training ZeroGrad on this dataset and better performance of ZeroGrad may be expected. Actually, due to the fact that ImageNet is highly computational demanding, investigation of ZeroGrad and generally single-step AT on this dataset is left for future works.

5.5. Evaluation based on AutoAttack

AutoAttack Croce and Hein (2020) is a parameter-free adversarial attack, which is an ensemble of a few different attacks. Its goal is to evaluate robustness of models in a reliable manner and identify the defenses that give a wrong impression of robustness. Many earlier proposed defenses resulted in much lower robust accuracy compared to other common attacks that are used for evaluation. To show that training the network with our methods does not cause gradient masking or gradient obfuscation, we evaluate the models based on the AutoAttack. The results show that our method is performing well against this attack, and increases our confidence on robustness of the models that are trained with the propoesed method, ZeroGrad.

5.6. Training time

Without a doubt, one of the main motivations for the FGSM-based attacks is to be fast. Therefore, it is necessary to report this aspect of our proposed method. The running time for one epoch of training with

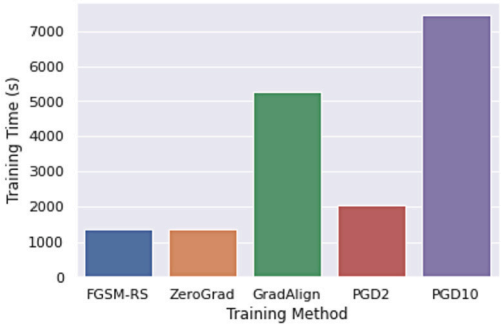


Fig. 5. The training time of different methods for 30 epochs, using half-precision on CIFAR-10 with Preact ResNet-18.

ZeroGrad is almost equal to the FGSM-RS method Wong et al. (2020), since no extra computational overhead is needed in this approach, except for calculating the thresholds. Approaches like Li et al. (2020) can not reduce the training time as much as FGSM-RS Wong et al. (2020), due to the fact that no parallelization is possible while training with multi-step adversarial attacks like PGD. In addition, GradAlign is more than two times slower than FGSM-RS and also cannot be parallelized Andriushchenko and Flammarion (2020).

The half-precision calculation technique Micikevicius et al. (2017), which leads to nearly two times speedup in the FGSM training, can also be used to speed up our methods without having a considerable effect on the standard or robust accuracy of our models. The reported results in our experiments are without using the half-precision technique.

The type of GPUs and other system specifications are inconsistent across papers and does not make much sense to compare the reported ones. In our experiments, by using T4 GPU, the training time of different methods for 30 epochs, using half-precision Micikevicius et al. (2017) on CIFAR-10 with Preact ResNet-18 is shown in Fig. 5. In Fig. 1a comparison of training time for WideResNet network is also available.

6. Ablation studies

In this section, we discuss the effects of hyperparameters that are used in our proposed methods. We mention a rationale and rule of thumb for choosing hyperparameters for a new problem and also introduce a practical technique for adjusting hyperparameters. Finally, we test whether the benefits of the proposed methods originate mainly from lowering the attack ℓ_1 norm or not. Additional experiments about the settings of the suggested method are available in the Appendix.

6.1. Hyperparameters selection

ZeroGrad has a single hyperparameter that needs to be adjusted. This hyperparameter, q, is used to determine the threshold for zeroing the gradients based on the lower q-quantile of the absolute values of the loss input gradient. The suitable choice of q is related to the given problem and dataset. As mentioned in the previous sections, by zeroing, we are ignoring the small gradients that have less importance, i.e. not relevant to the main object of interest in the sample. This appears to be related to the given task, i.e. what percentage of pixels have no/little information about predicting the output. Therefore, the dataset can give us some hints about the percentage of pixels that are informative. Our estimation about the average percentage of such pixels in the images of our dataset provides us a clue to choose the appropriate q.

In this regard, a high percentage of irrelevant and less important parts, is a sign for a larger q in ZeroGrad. For example, SVHN dataset contains real-world images that have few digits as a house number and a usually large portion as the background. Through inspecting the images, we realize that many of the pixels have no considerable information about the class label of the images in this dataset. Confirming this point, we observe in our experiments that for SVHN dataset, larger q is needed to prevent catastrophic overfitting, especially in comparison with CIFAR-10 and CIFAR-100 datasets.

6.2. Adjusting the hyperparameter by rolling back

In addition to the initial estimation of hyperparameters according to the learning problem and the dataset, it is possible to adjust hyperparameters by a rolling back approach. Using a validation dataset for detection of catastrophic overfitting, we can choose a small approximate q_0 in the beginning and start the ZeroGrad training algorithm with q_0, until catastrophic overfitting occurs. We save checkpoints during training to make rolling back possible. The occurrence of this failure mode shows that larger q is needed to keep on training for more epochs in our problem. So once faced by the catastrophic overfitting phenomenon, we stop the training, increase the chosen q hyperparameter and adjust it to q_1, which is larger than q_0. We then continue the training with the ZeroGrad algorithm, resuming from a saved checkpoint at the epoch right before catastrophic overfitting.

To test the rolling back idea, we used CIFAR-10 dataset and successfully trained the model for more than 100 epochs without overfitting, starting from q_0 equal to 0.25 and increasing it to 0.45 during the experiment. By this approach, we can get an appropriate q for a new dataset and continue the training for a desired number of epochs. Actually, one of the main consequences of catastrophic overfitting is stopping the training procedure prematurely, i.e. before achieving the best possible performance using that setting. Although employing remedies like early stopping prevents from resulting in models with 0 robust accuracy, it is not able to let premature models continue the training and achieve their best potential results. Using our approach, we can prolong training until we get the best results.

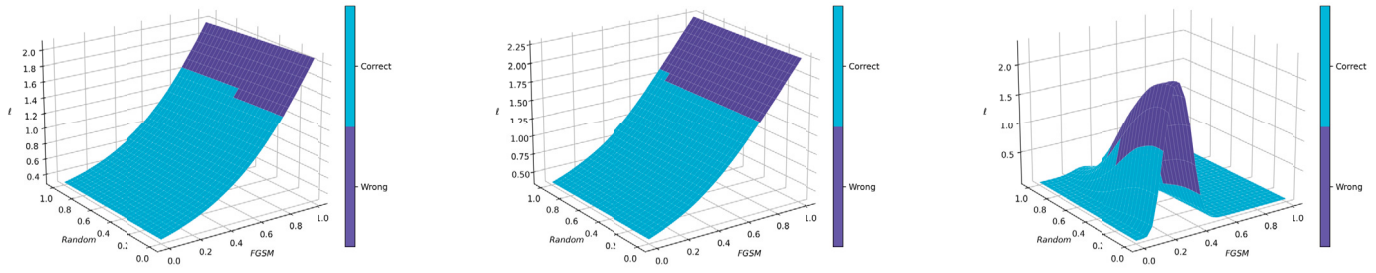
6.3. Decision boundary distortion

One of the possible signs of catastrophic overfitting in a model is believed to be the distortion in the decision boundary Kim et al. (2021). In the case of distortion, the loss surface of the overfitted model is curved and non-linear in the ϵ-ball around the inputs. This results in a highly robust accuracy against single-step attacks that target the borders of the decision boundary, and a low robust accuracy against multi-step attacks such as PGD. To further illustrate that our model does not suffer from a distorted loss surface, we plotted the loss surface in Fig. 6 for a sample image using the code from Kim et al. (2021) and compared it to the other models. The loss for the models trained with ZeroGrad and GradAlign, unlike the model trained with FGSM-RS that suffers from catastrophic overfitting, has a linear behavior which is something that GradAlign achieves using a regularization factor to linearize the loss surface, but ZeroGrad achieves implicitly.

6.4. Mean perturbation size

One may argue that zeroing certain elements of the input gradients, results in lower $\|\delta\|_1$, and similar results could effectively be obtained by lowering the FGSM step size. We note that lowering the perturbation ℓ_1 norm has earlier been shown to be helpful in avoiding catastrophic overfitting Andriushchenko and Flammarion (2020). One should note that lowering the perturbation norm is not an ideal solution to avoid catastrophic overfitting, as it may harm the robust accuracy. To investigate this further, we compare different methods based on their average perturbation ℓ_1 norm.

We compare our proposed techniques with simply reducing the FGSM step size α to decrease the perturbation size during training on the CIFAR-10 dataset. The results that are reported in Table 5, are calculated based on the whole training set samples once at the beginning of the training in the second epoch, and once at the end of the training in 52-nd epoch. All different methods either have a fixed perturbation size during the training, or their perturbation size decreases as the model



(a) ZeroGrad ($\alpha = 2.0,\ Q = 0.35$) (b) GradAlign (c) FGSM-RS ($\alpha = 2.0$)

Fig. 6. The comparison of the loss surfaces in the decision boundaries of a sample image for three different models. The loss is plotted along with the directions of the FGSM attack vector and a random vector.

Table 5
ℓ_1 perturbation norm (multiplied by 255) at the second and the 52-nd epochs during training with one-drop learning rate schedule for 52 epochs. The training is done on the CIFAR-10 dataset with $\epsilon = 8/255$, and FGSM step size of 2 for ZeroGrad.

Method	2nd epoch	52nd epoch
ZeroGrad (q = 0.35)	6.52	6.49
FGSM-RS ($\alpha = 1.0$)	5.96	5.95
FGSM-RS ($\alpha = 1.25$)	6.81	6.78
GradAlign	7.89	7.89
PGD-2	7.21	6.67
PGD-10	7.52	7.16

gets more robust to the adversarial attacks. It can be seen that by reducing the FGSM-RS step size to 1.0, the perturbation norm would be less than that of our proposed method, but its PGD-50 test robust accuracy would be 45.44%, which is less than what our method can achieve, which is 47.55%. Overall, FGSM-RS cannot reach a test robust accuracy close to our methods with any value of the FGSM step size α. So the mechanism by which our method is avoiding catastrophic overfitting is not simply through shrinkage of the adversarial perturbation.

6.5. FGSM step size

FGSM-RS can help to slightly mitigate CO for small enough epsilons by adding an initial random noise. However, this only holds when the step size (α) is limited. FGSM-RS is vulnerable to the large values of step size; For example in training on CIFAR-10, for epsilon = 8/255, with alpha = 16/255, FGSM-RS fails and faces CO. Earlier, we just reported our performance on the largest step-size to make clear that our method is not prone to the step size.

In this section, we report the performance of ZeroGrad with different values for FGSM step size. We trained a Preact ResNet-18 He et al. (2016) with ZeroGrad for 30 epochs and varied the step size from alpha = 8/255 to alpha = 16/255. Results of standard accuracy and PGD-50 accuracy are shown in Fig. 7.

7. Conclusion and future work

A number of recent works tried to understand FGSM and FGSM-RS adversarial training, its unknown phenomenon, "catastrophic Overfitting," and also suggest solutions to prevent its occurrence. However, existing explanations do not provide any acceptable answer to some questions about specific aspects of this phenomenon, such as its suddenness. Furthermore, the previously suggested solutions usually suffer from the increased training time, which contradicts the original aim of such methods to reduce the computational cost of adversarial training. In this work, we aimed to present a more comprehensive explanation for catastrophic overfitting. Our hypothesis highlights the role of tiny and fragile input gradients in the training with the fast gradient sign method. Based on this intuition, we proposed an affordable effective method. Our empirical results show that our method prevents catastrophic overfitting, and achieve competitive adversarial accuracy with no significant train-time overhead. A more comprehensive study of why the large weight update is corrupting adversarial robustness remains a subject of future work. This could potentially result in more useful insights and lead to further advancement of single-step attacks for the adversarial training.

CRediT authorship contribution statement

Zeinab Golgooni, Mehrdad Saberi, Masih Eskandar, and Mohammad Hossein Rohban contributed to the conception, design, analysis, interpretation of results, and drafting of the manuscript. Mehrdad Saberi, and Masih Eskandar also contributed to writing and running codes, and designing and performing the ablation studies. Mohammad Hossein Rohban revised the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A. Analysis on hyperparameters

In this section, we investigate and discuss effects of the main hyperparameter of our proposed method (q in ZeroGrad) in the final results.

A.1. Different settings for the ZeroGrad

Results of the ZeroGrad algorithm with the final selected hyperparameters have previously been reported in the paper. In this section, we report the results of our experiments on CIFAR-10 with different hyperparameter (q). Table 6 shows the results of ZeroGrad with different q. We used a Preact ResNet-18 He et al. (2016), and 30 epochs of training with the cyclic learning rate schedule similar to our other experiments.

Finding the most appropriate value of q for training the model with ZeroGrad can be challenging in some cases. We note that it depends on different aspects of the model and dataset. We noted earlier in the paper that the elements of the gradient that do not contribute to catastrophic overfitting are seemingly the ones that are correlated with the main object of interest, i.e. the foreground. Furthermore, it seems that most of the *fragile* gradient elements are small in magnitude. However, we have

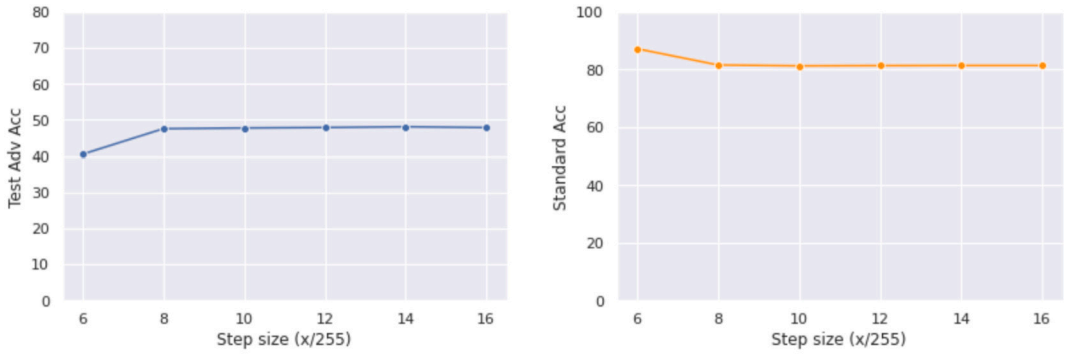


Fig. 7. Standard and robust test performance of ZeroGrad over different step sizes with $\epsilon = 8/255$.

Table 6
Standard and PGD-50 accuracy on the CIFAR-10 dataset with $\epsilon = 8/255$ for different settings of ZeroGrad. All models are trained with the cyclical learning rate schedule for 30 epochs.

q =	0.27	0.3	0.33	0.35	0.36	0.4	0.43	0.46	0.5
Standard Acc	81.47	81.27	81.48	81.68	81.64	81.70	81.85	82.00	82.28
PGD−50 Acc	47.95	48.00	47.70	47.85	47.46	47.31	46.64	46.61	45.98

no reason to belive that all small elements of the gradient are fragile. By zeroing out gradient elements that are not fragile, we may lose some helpful information, and consequently achieve a lower accuracy than what we could get otherwise.

The SVHN dataset seems to require a higher value of q (i.e. about 70%) in order to prevent catastrophic overfitting, and even with the higher q, ZeroGrad training results in less than the desired accuracy. After a specific point in the training, its accuracy decreases in each epoch. For this reason, we used the validation-based early stopping scheme. It is worth mentioning that this was not the case with the MultiGrad training. MultiGrad on SVHN seems to reach similar results to GradAlign and also its accuracy did not seem to diminish significantly over time. Regardless, we reported the results using the early stopping scheme for the sake of consistency, and fair comparison. The peculiarity of this dataset and its relation to the value of q is still largely unexplained to us aside from the initial hypothesis that we mentioned earlier, and could be subject of further research.

For the case of the CIFAR-10 dataset, it seems that the lower the value of q, the better the model seems to perform, provided that q is sufficiently large to avoid catastrophic overfitting. Hence it seems that the *fragile* elements of the gradient are smaller than the rest, i.e. they make up the lower portion of all gradient elements. The results of training on different values of q for CIFAR-10 can be seen in Table 6. We trained the models for 30 epochs with a cyclical learning rate similar to our other experiments.

Appendix B. Detailed results against different attacks

For comprehensive evaluation of robustness, the model should be evaluated with various attacks including strong PGDs and black-box attacks. In this regard, we have evaluated zeroGrad against PGD-50-10, untargeted APGD-CE, targeted APGD-DLR, targeted FAB, Square Attack as a query-efficient black-box attack Croce and Hein (2020). Table 7 shows these results in detail.

Appendix C. Alternative methods

Considering simplicity as one of main goals, ZeroGrad is our main suggested methods. However, based on our intuition there can be other methods for preventing occurence of the catastrophic overfitting. In this section we discuss two possible methods. The first try to avoid large weight updates by clipping and the second one, try to recognize fragile gradients by a different approach.

C.1. Gradient clipping

As stated in Section 4, large gradient updates lead the model to overfit on FGSM and perform poorly on stronger attacks. To avoid these updates, we can clip them such that they do not interfere with the gainful updates that help with the model's training. While increasing the clipping threshold reduces its effect and might not prevent catastrophic overfitting, decreasing this threshold interrupts the model's convergence. As shown in Table 8, with a clipping value of 0.1, the model can reach a high accuracy compared to the existing methods, but the training process is unstable and the model would overfit in some cases. However, we believe that it might be possible to use a more complex clipping method to achieve stable training with competitive accuracies to our proposed methods. Note that the time overhead of using gradient clipping is negligible compared to the total FGSM training time.

C.2. MultiGrad

We try to tackle the problem of zeroing out *problematic* gradient elements by identifying the *fragile* ones in a different way. While it does seem that these fragile gradients are generally of small magnitude, we have no reason to believe that all gradients of small magnitude are fragile. Thus it would seem that in the process of zeroing out all gradients below a certain threshold, we are eliminating perturbations that are not necessarily contributing to catastrophic overfitting, and losing some robustness as a result.

As is the sake of the name, these fragile gradients easily change their sign as we move in the ϵ ball around our sample. Let $\delta_1, \delta_2, \dots, \delta_k$ be k different FGSM-RS perturbations corresponding to k random starting positions. Let A be the set of indices of the input gradient elements where these perturbations *all* agree on the same sign:

$$A = \{i |\ 1 \leq i \leq d,\ [\text{sgn}(\delta_1)]_i = [\text{sgn}(\delta_j)]_i \quad \forall : 1 < j \leq k\} \tag{7}$$

We define $\bar{\delta}'$ as the perturbation that is equal to δ_1 for the elements that are corresponding to the indices of A, and equal to zero for the rest. Details of the algorithm is given in the Algorithm 2. The evaluation results of the MultiGrad algorithm is provided in Table 9.

Table 7
Performance of ZeroGrad for CIFAR10, CIFAR100 and SVHN datasets for $\epsilon = 8/255$ againts different attacks in detail.

Dataset	Clean	PGD-50	C&W	APGD-CE	APGD-T	FAB-T	SQUARE
CIFAR10	81.84	47.85	38.24	47.24	44.05	44.05	44.05
CIFAR100	53.89	25.19	21.14	24.71	20.94	20.93	20.93
SVHN	91.08	36.84	-	34.46	31.49	31.48	31.48

Table 8
Results for FGSM-RS adversarial training combined with gradient clipping on CIFAR-10 dataset with $\alpha = 2\epsilon$ for $\epsilon = 8/255$. Training is done using cyclical learning rate for 30 epochs and with 3 different random seeds.

Clipping Value	Standard Acc.	PGD-50 Acc.	Max PGD-50 Acc.
0.02	46.67 ± 0.28	35.11 ± 0.04	35.16
0.06	69.04 ± 0.56	45.48 ± 0.17	45.67
0.10	78.23 ± 2.08	31.11 ± 22.00	48.51

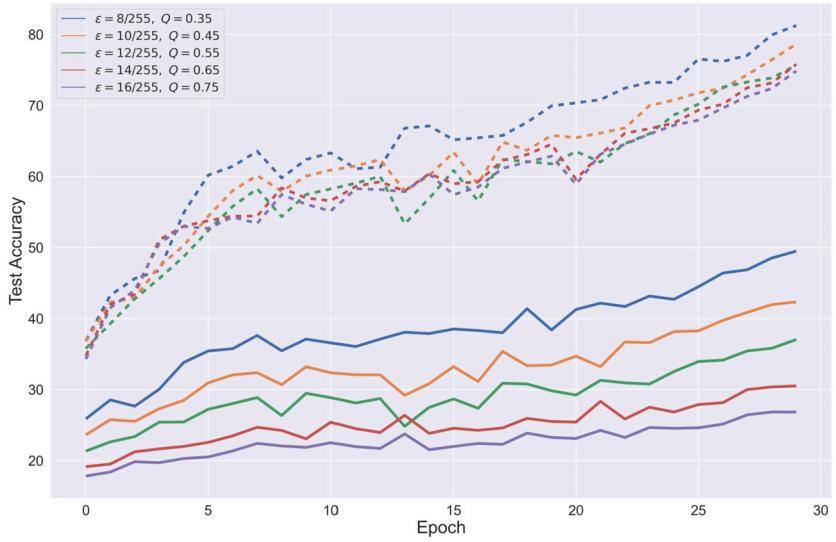


Fig. 8. Training diagram of ZeroGrad for different values of ϵ on CIFAR-10. The dotted lines and straight lines show the standard and robust PGD-10 accuracies respectively.

Algorithm 2 MultiGrad.

Input: number of epochs T, maximum perturbation ϵ, number of samples N, step size α, dataset of size M, network f_θ
Output: Robust network f_θ
for $t = 1$ **to** T **do**
  for $i = 1$ **to** M **do**
    $\nabla_{\delta_{1..N}} \leftarrow \mathbf{0}$
    for $j = 1$ **to** N (in parallel) **do**
      $\delta_j \sim \text{Uniform}([-\epsilon, \epsilon]^d)$
      $\nabla_{\delta_j} \leftarrow \nabla_x \ell(f_\theta(x_i + \delta_j), y_i)$
    end for
    $\omega \leftarrow \frac{1}{N} \sum_{j=1}^{N} \text{sgn}(\nabla_{\delta_j})$
    $\nabla_\delta \leftarrow 0$
    $\nabla_\delta[|\omega| == 1] \leftarrow \nabla_{\delta_1}$
    $\delta \leftarrow \alpha \cdot \text{sgn}(\nabla_\delta)$
    $\delta \leftarrow \max(\min(\delta, \epsilon), -\epsilon)$
    // Update network parameters with some optimizer
    $\theta \leftarrow \theta - \nabla_\theta \ell(f_\theta(x_i + \delta), y_i)$
  end for
end for

Appendix D. Additional experiments

D.1. Training with larger maximum perturbation size ϵ

In this section, we train models using our method for maximum perturbation sizes larger than $\epsilon = 8/255$ on CIFAR-10 dataset. As it can be seen in Fig. 8, ZeroGrad is able to prevent catastrophic overfitting even for $\epsilon = 16/255$. This is while training with FGSM-RS ($\alpha = 1.25$) results in catastrophic overfitting for $\epsilon \geq 10/255$.

Additionally, for $\epsilon = 16/255$, ZeroGrad can be combined with PGD-2 to achieve a stable test accuracy of 37.06 ± 0.24 against PGD-10. This is while training only using PGD-2 is unstable for this value of ϵ and sometimes results in poor robust test accuracy. Note that combining PGD-2 and ZeroGrad means using two steps in ZeroGrad, and this still has a computational time advantage against the GradAlign method.

Table 9
Standard and PGD-50 accuracy on the CIFAR-10 dataset with $\epsilon = 8/255$ for different settings of MultiGrad. All models are trained with the cyclical learning rate schedule for 30 epochs.

Method	Standard Acc.	PGD-50 Acc.
MultiGrad (N = 2)	81.48 ± 0.17	48.21 ± 0.15
MultiGrad (N = 3)	81.45 ± 0.08	48.05 ± 0.09
MultiGrad (N = 4)	82.04 ± 0.18	47.78 ± 0.26

D.2. Training with more epochs

In this section, we train our models with more epochs on the CIFAR-10 dataset. In order to do so, we use the onedrop learning rate schedule.

Table 10
Percentage of gradient elements that differ in the perturbation sign with PGD-10.

Method	Before Ovft.	after Ovft.
FGSM	17.58	48.53
MultiGrad (N = 3)	11.48	12.82
ZeroGrad (q = 0.4)	6.49	28.66

The ZeroGrad method is able to train with $q = 0.35$, and for 52 epochs, and it reaches the robust accuracy of 47.79±0.11% against PGD-50. But if we want to train the model with ZeroGrad for 100 epochs or more, we have to increase q in order to prevent catastrophic overfitting, and we would end up with a worse robust accuracy. For example, if we want to train for 102 epochs with $q = 0.45$, the robust accuracy would be 47.21%, which is less than the training for 52 epochs with $q = 0.35$.

D.3. Similarity to PGD perturbation

To further our understanding, we evaluate the similarity of our methods with the Projected Gradient Descent, and compare it with FGSM. In order to do so, we take a look at the mean percentage of difference in sign of the PGD-10 and other methods perturbations over one epoch. That is, the number of non-zero perturbation features that are of different signs with the PGD perturbation divided by the number of all of the features, and averaged over all mini-batches is calculated. We do so for the model that is trained based on the FGSM (without random starts for any of the methods) and report the values for one epoch before, and one epoch after the catastrophic overfitting occurs. The results are shown in Table 10.

It seems that, overall, our methods are reducing the difference in perturbation with the PGD, which is almost expected for ZeroGrad, because all it does is zeroing out elements. However, this also happens in MultiGrad. Furthermore, after overfitting occurs, the difference increases significantly for both FGSM and ZeroGrad, but not by much for the MultiGrad. This goes to show that MultiGrad is in some ways similar to the PGD-10 as the catastrophic change to the model does not seem to increase their difference as is seen with the other methods.

D.4. Training with different architectures

For more comprehensive evaluation and investigation, we also test our proposed method using more complex architectures, WideResNet and Vision Transformer (ViT).

D.4.1. Training with WideResNet

WideResNet is a network that is more complex than PreActResNet. Here, we use WideResNet-34 with the width factor 10. Training with this network is around 5 times slower than that of the Preact ResNet-18. The accuracy for training with this network is shown in Table 11. As expected, both the standard and robust accuracy of the models are higher than those of the Preact ResNet-18. We note that a larger q is needed to prevent the wider architectures from catastrophic overfitting. We hypothesize that such networks are more vulnerable to get catastrophically overfitted by a large weight update, as they contain more filters.

D.4.2. Training with vision transformers

Vision Transformers (ViTs) are novel architectures, resulted from adapting transformers for computer vision tasks Dosovitskiy et al. (2020). They achieve significant performance on several datasets. Recently, some researches try to analyze these architectures from the lens of robustness. Some recent works argued that ViTs are naturally more robust than CNNs Bhojanapalli et al. (2021), Shao et al. (2021). Currently, this comparison seems a controversial topic, due to the different settings for testing models Bai et al. (2021). However, improving the robustness of ViTs by using different techniques including adversarial training is an active line of research.

Table 11
Standard and PGD-50 accuracy with WideResNet-34 on CIFAR-10 datasets. The training is done with the onedrop learning rate schedule and for 52 epochs.

Method	Standard Acc.	PGD-50 Acc.
ZeroGrad (q = 0.5)	82.63	**47.44**
FGSM-RS	**87.17**	43.94
GradAlign	77.58	47.14
PGD2	81.29	50.70
PGD10	81.42	**51.34**

Table 12
Standard and PGD-50 accuracy on the CIFAR-10 dataset with $\epsilon = 8/255$ for different training methods with ViTs.

Method	Standard	PGD-10	PGD-50
ZeroGrad (q = 0.7)	90.74	41.8	38.09
FGSM-RS ($\alpha = 1.25$)	-	-	33.06
GradAlign	88.97	47.65	44.38
PGD-2	-	-	45.00

Shao et al. (2021) is one of the current works that try to train the ViTs in an adversarial manner. We use their code for testing ZeroGrad and other methods- FGSM, FGSM-RS, GradAlign, and PGD2- to observe their behavior with this architecture and to check if catastrophic overfitting happens or not. In the first part of our experiment, we see no occurrence of catastrophic overfitting neither for simple FGSM. After reviewing their code, we notice that they clip gradient updates of the network's weights. We leave out this action and do the experiments again. Interestingly, we observe that FGSM and FGSM-RS both drastically suffer from catastrophic overfitting. In addition, ZeroGrad successfully prevents its occurrence. Remarkably, all of these observations together, support our intuitions in ViT setting.

The accuracy for training with this architecture type is shown in Table 12. We used ViT-B/16 with 87M parameters, which is pre-trained on the ImageNet dataset. All methods are run with the cyclical learning rate schedule for 30 epochs. There is a gap between the performance of ZeroGrad and GradAlign. However, it can be permissible considering the large gap between their cost of training which is more worthy of note for such a complex network. In conclusion, these experiments not only showed the effectiveness of our method but also confirmed our suggested theoretical intuition.

References

Andriushchenko, M., & Flammarion, N. (2020). Understanding and improving fast adversarial training. In *NeurIPS*.

Bai, Y., Mei, J., Yuille, A. L., & Xie, C. (2021). Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, *34*.

Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., & Veit, A. (2021). Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10231–10241).

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)* (pp. 39–57). IEEE.

Croce, F., & Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning* (pp. 2206–2216). PMLR.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. preprint, arXiv: 2010.11929.

Filipovich, I., & Kovalev, V. (2023). Dependence of the results of adversarial attacks on medical image modality, attack type, and defense methods. In *Diagnostic biomedical signal and image processing applications with deep learning methods* (pp. 179–195). Elsevier.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. CoRR, arXiv:1412.6572 [abs].

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630–645). Springer.

de Jorge, P., Bibi, A., Volpi, R., Sanyal, A., Torr, P. H., Rogez, G., & Dokania, P. K. (2022). Make some noise: Reliable and efficient single-step adversarial training. preprint, arXiv:2202.01181.

Kang, P., & Moosavi-Dezfooli, S. M. (2021). Understanding catastrophic overfitting in adversarial training. preprint, arXiv:2105.02942.

Kim, H., Lee, W., & Lee, J. (2021). Understanding catastrophic overfitting in single-step adversarial training. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8119–8127).

Li, B., Wang, S., Jana, S., & Carin, L. (2020). Towards understanding fast adversarial training. preprint, arXiv:2006.03089.

Liu, M., Zhang, Z., Chen, Y., Ge, J., & Zhao, N. (2023). Adversarial attack and defense on deep learning for air transportation communication jamming. *IEEE Transactions on Intelligent Transportation Systems*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International conference on learning representation*.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. (2017). Mixed precision training. preprint, arXiv:1710.03740.

Rice, L., Wong, E., & Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In *International conference on machine learning* (pp. 8093–8104). PMLR.

Shao, R., Shi, Z., Yi, J., Chen, P.Y., & Hsieh, C.J. (2021). On the adversarial robustness of visual transformers. arXiv e-prints, arXiv–2103.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)* (pp. 464–472). IEEE.

Tian, J., Wang, B., Guo, R., Wang, Z., Cao, K., & Wang, X. (2021). Adversarial attacks and defenses for deep-learning-based unmanned aerial vehicles. *IEEE Internet of Things Journal*, *9*, 22399–22409.

Tramèr, F., Behrmann, J., Carlini, N., Papernot, N., & Jacobsen, J. H. (2020). Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *International conference on machine learning* (pp. 9561–9571). PMLR.

Vitorino, J., Praça, I., & Maia, E. (2023). Towards adversarial realism and robust learning for iot intrusion detection and classification. *Annals of Telecommunications*, 1–12.

Vivek, B., & Babu, R. V. (2020a). Regularizers for single-step adversarial training. preprint, arXiv:2002.00614.

Vivek, B., & Babu, R. V. (2020b). Single-step adversarial training with dropout scheduling. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 947–956). IEEE.

Wong, E., Rice, L., & Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. In *International conference on learning representations*.

Xu, C., Zhang, C., Yang, Y., Yang, H., Bo, Y., Li, D., & Zhang, R. (2023). Accelerate adversarial training with loss guided propagation for robust image classification. *Information Processing & Management*, *60*, Article 103143.