Make Some Noise: Reliable and Efficient Single-Step Adversarial Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recently, Wong et al. [30] showed that adversarial training with single-step FGSM leads to a characteristic failure mode named *Catastrophic Overfitting* (CO), in which a model becomes suddenly vulnerable to multi-step attacks. Experimentally they showed that simply adding a random perturbation prior to FGSM (RS-FGSM) could prevent CO. However, Andriushchenko and Flammarion [1] observed that RS-FGSM still leads to CO for larger perturbations, and proposed a computationally expensive regularizer (GradAlign) to avoid it. In this work, we methodically revisit the role of noise and clipping in single-step adversarial training. Contrary to previous intuitions, we find that using a *stronger noise* around the clean sample combined with *not clipping* is highly effective in avoiding CO for large perturbation radii. We then propose *Noise-FGSM* (N-FGSM) that, while providing the benefits of single-step adversarial training, does not suffer from CO. Empirical analyses on a large suite of experiments show that N-FGSM is able to match or surpass the performance of previous state-of-the-art GradAlign, while achieving $3\times$ speed-up.

1 Introduction

Deep neural networks have achieved remarkable performance on a variety of tasks [11, 25, 6]. However, it is well known that they are vulnerable to small worst-case perturbations around the input data – commonly referred to as *adversarial examples* [26]. The existence of such adversarial examples poses a security threat to deploying models in sensitive environments [2, 3]. This has motivated a large body of work towards improving the *adversarial robustness* of neural networks [10, 20, 27, 23, 4].

The most popular family of methods for learning robust neural networks is based on the concept of *adversarial training* [10, 18]. In a nutshell, adversarial training can be posed as a min-max problem where instead of minimizing some loss over a dataset of *clean* samples, we augment the inputs with worst-case perturbations that are generated online during training. However, obtaining such perturbations is NP-hard [29] and hence, different *adversarial attacks* have been suggested that approximate them. In their seminal work, Goodfellow et al. [10] proposed the *Fast Gradient Sign Method* (FGSM), that generates adversarial attacks by performing a gradient ascent step on the loss function. Yet, while FGSM-based adversarial training provides robustness against single-step FGSM adversaries, Tramèr et al. [27] showed that these models are still vulnerable to multi-step attacks, namely those allowed to perform multiple gradient ascent steps. Given their better (robust) performance, multi-step attacks such as *Projected Gradient Descent* (PGD) [18] have now become the de facto standard for adversarial training.

The main downside of multi-step adversarial training is that the cost of these attacks increases linearly with the number of steps, making their applicability often computationally prohibitive. For this reason, several works have focused on reducing the cost of adversarial training by approximating the worst-case perturbations with single-step attacks [30, 24, 28]. In particular, Wong et al. [30] studied

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

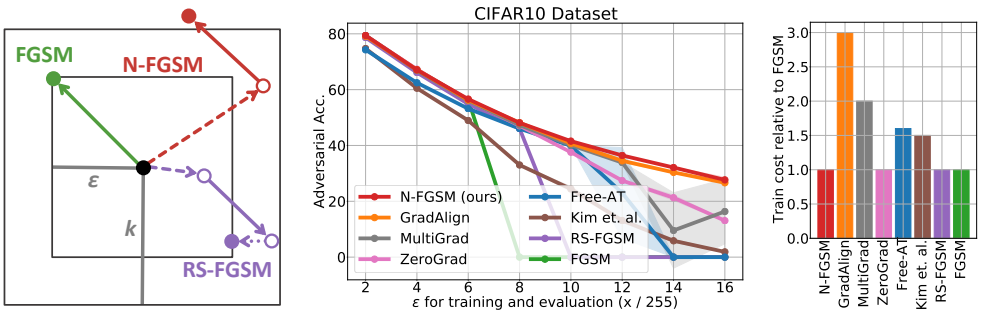


Figure 1: **Left:** Visualization of FGSM [10], RS-FGSM [30] and N-FGSM (ours) attacks. While RS-FGSM is limited to noise in the $\epsilon - l_\infty$ ball, N-FGSM draws noise from an arbitrary $k - l_\infty$ ball. Moreover, N-FGSM does not clip the perturbation around the clean sample. **Middle:** Comparison of single-step methods on CIFAR-10 with PreactResNet18 over different perturbation radii (ϵ is divided by 255). Our method, N-FGSM, can match or surpass state-of-the-art results while *reducing the cost by a* $3\times$ *factor*. Adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds. **Right**: Comparison of training costs relative to FGSM baseline based on the number of Forward-Backward passes, see Appendix L for details.

FGSM adversarial training and discovered that it suffers from a characteristic failure mode, in which a model suddenly becomes vulnerable to multi-step attacks despite remaining robust to single-step attacks. This phenomenon is referred to as *Catastrophic Overfitting* (CO). As a solution, they argued that adding a random perturbation prior to FGSM (RS-FGSM) seemed sufficient to prevent CO and produce robust models. Yet, Andriushchenko and Flammarion [1] recently observed that RS-FGSM still leads to CO as one increases the perturbation radii of the attacks. They suggested a regularizer (GradAlign) that can avoid CO in the settings they considered, but at the expense of computing a double derivative – significantly increasing the computational cost with respect to RS-FGSM.

In this paper, we revisit the idea of including noise in single-step attacks. Differently from previous methods that consider the noise as part of the attack, we propose an adversarial training procedure where the noise is used as a form of *data augmentation*. As we detail in Section 4, this motivates us to introduce two main changes with respect to previous methods: 1) We center adversarial perturbations with respect to noise-augmented samples and therefore, unlike previous RS-FGSM, we do not clip around the clean samples. 2) We use noise perturbations larger than the $\epsilon-$ball, since they are not restricted by the strength of the attack anymore. Our experiments show that performing data augmentation with sufficiently *strong noise* and removing the *clipping step* improves model robustness and prevents CO, even against large perturbation radii. Our new method, termed N-FGSM, matches, or even surpasses, the robust accuracy of the regularized FGSM introduced by Andriushchenko and Flammarion [1] (GradAlign), while *providing a 3× speed-up*.

To corroborate the effectiveness of our solution, we present an experimental survey of recently proposed single-step attacks and empirically demonstrate that N-FGSM trades-off robustness and computational cost better than any other single-step approach, evaluated over a large spectrum of perturbation radii (see Figure 1, middle and right panels), over several datasets (CIFAR-10, CIFAR-100, and SVHN) and architectures (PreActResNet18 and WideResNet28-10). We will release our code to reproduce the experiments.

2 Related Work

Since the discovery of adversarial examples, many defense mechanisms have been proposed, *adversarial training* being one of the most popular and empirically validated. We can categorise adversarial training methods based on how they approximate the perturbations applied to training samples. *Multi-step* approaches approximate an inner maximization problem to find the worst-case perturbation with several gradient ascent steps [32, 16, 18]. While this provides a better approximation, it is also more expensive. At the other end of the spectrum, *single-step* methods only use one gradient step to approximate the worst case perturbation. Goodfellow et al. [10] first proposed FGSM; Tramèr et al. [27] proposed a new variant with an additional random step (R+FGSM), but observed that both methods were

vulnerable to multi-step attacks. Shafahi et al. [24] proposed *Free Adversarial Training* (Free-AT), which successfully reduced the computational cost of training by using a single backward pass to compute both weight update and attack. Motivated by this, Wong et al. [30] explored a variant of R+FGSM, namely RS-FGSM, that uses a less restrictive form of noise and showed this can improve robustness for moderate perturbation radii at the same cost as FGSM. Recently, Andriushchenko and Flammarion [1] proposed the GradAlign regularizer. Combining FGSM with GradAlign results in robust models at even larger perturbation radii. However, GradAlign suffers from a *threefold* increase in the training cost to as compared to FGSM. The need for more efficient solutions has motivated a growing body of work whose goal is the design of computationally lighter single-step methods [9, 14, 28, 21, 17].

In this work, we revisit the idea of combining noise with the FGSM attack. Our method builds upon FGSM and intuitions from R+FGSM and RS-FGSM to combine it with random perturbations, however, we consider the noise step as data augmentation rather than part of the attack. This motivates us to use a stronger noise *without* clipping. As opposed to [13], our thorough study leads to a practically effective approach that yields robustness also against large perturbation radii.

3 Preliminaries on Single-Step Adversarial Training

Given a classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ and a perturbation set $\mathcal{S}$, f_θ is defined as *robust* at $x \in \mathcal{X}$ on the set $\mathcal{S}$ if for all $\delta \in \mathcal{S}$ we have $f_\theta(x+\delta) = f_\theta(x)$. One of the most popular definitions for $\mathcal{S}$ is the $\epsilon - \ell_\infty$ ball, *i.e.,* $\mathcal{S} = \{\delta : \|\delta\|_\infty \leq \epsilon\}$. This is known as the l_∞ threat model which we adopt throughout this work.

To train networks that are robust against ℓ_∞ threat models, adversarial training modifies the classical training procedure of minimizing a loss function over a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1:N}$ of images $x_i \in \mathcal{X}$ and labels $y_i \in \mathcal{Y}$. In particular, adversarial training instead minimizes the worst-case loss over the perturbation set $\mathcal{S}$, *i.e.,* training is on the adversarially perturbed samples $\{(x_i+\delta_i, y_i)\}_{i=1:N}$. Under the l_∞ threat model, we can formalize adversarial training as solving the following problem:

$$\min_\theta \sum_{i=1}^{N} \max_\delta \mathcal{L}(f_\theta(x_i+\delta), y_i) \;\; \text{s.t. } \|\delta\|_\infty \leq \epsilon, \tag{1}$$

where $\mathcal{L}$ is typically the cross-entropy loss. Due to the difficulty of finding the exact inner maximizer, the most common procedure for adversarial training is to approximate the worst-case perturbation through several PGD steps [18]. While PGD has been shown to yield robust models, its cost increases linearly with the number of steps. As a result, several works have focused on reducing the cost of adversarial training by approximating the inner maximization with a single-step.

If the loss function is linear with respect to the input, the inner maximization of Equation (1) will enjoy a closed form solution. Goodfellow et al. [10] leveraged this to propose FGSM, where the adversarial perturbation follows the direction of the sign of the gradient. Tramèr et al. [27] proposed adding a random initialization prior to FGSM. However, both methods were later shown to be vulnerable against multi-step attacks, such as PGD. Contrary to prior intuition, recent work from Wong et al. [30] observed that combining a random step with FGSM can actually lead to a promising robustness performance. In particular, most recent single-step methods approximate the worst-case perturbation solving the inner maximization problem in Equation (1) with the following general form:

$$\delta = \psi\Big(\eta + \alpha \cdot \text{sign}\big(\nabla_{x_i}\mathcal{L}(f_\theta(x_i+\eta), y_i)\big)\Big), \tag{2}$$

where η is drawn from a distribution Ω. For example, when ψ is the projection operator onto the ℓ_∞ ball and Ω is the uniform distribution $[-\epsilon, \epsilon]^d$, where d is the dimension of $\mathcal{X}$, this recovers RS-FGSM. Under a different noise setting where $\Omega = (\epsilon - \alpha) \cdot \text{sign}\left(\mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)\right)$ and by choosing the step size α to be in $[0, \epsilon]$, we recover R+FGSM by Tramèr et al. [27]. This was among the first works to explore the application of noise to FGSM, but did not report improvements over it. If we consider Ω to be deterministically 0 and ψ to be the identity map, we recover FGSM. Finally, if we take FGSM and add a gradient alignment regularizer, this recovers GradAlign.

4 Noise and FGSM

Previous methods that combined noise with FGSM, *e.g.,* R+FGSM [27] and RS-FGSM [30], have considered the noise as a *random step* integrated within the attack. Since it is a common practice

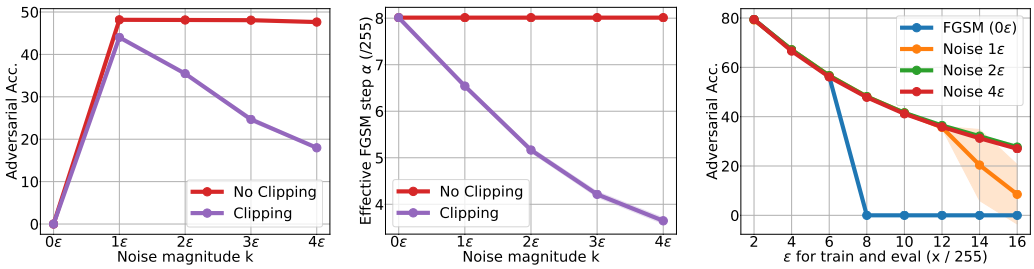


Figure 2: **Left:** Ablation of clipping vs not clipping around the clean sample x for $\epsilon = 8/255$. Clipping leads to a significant drop in robustness which increases with the strength of the noise augmentations. **Middle:** Analysis of the effective FGSM step size after clipping. We observe that clipping leads to a decrease in the effective FGSM step size, thus, adversarial perturbations will be more similar to random noise. **Right**: N-FGSM (ours) when varying the noise magnitude k (ϵ is divided by 255). Increasing the amount of noise is key to avoiding CO. For (left) and (right) plots, adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds.

to restrict adversarial perturbations to the $\epsilon-$ball, we argue that this introduces a trade-off between the magnitude of the random step and that of the attack. For illustration, consider the purple lines corresponding to RS-FGSM in Figure 1 (left). If the initial random step is significantly larger than the $\epsilon-$ball, then the final clipping step will have a noticeable impact on the perturbation, possibly removing a considerable portion of the FGSM step (middle arrow). To prevent this from happening, R+FGSM and RS-FGSM restrict the random step to lie within the $\epsilon-$ball, thereof implicitly entangling the noise magnitude and the attack strength.

Contrary to previous methods, we note that adding noise to the clean sample can be considered as a form of *data augmentation* to be applied independently from the attack. We make two considerations from this perspective 1) When one performs data augmentation during adversarial training, the input after the corresponding transformation is the starting point to compute the adversarial perturbations, therefore, we argue that adversarial attacks should be centered around the noise-augmented samples. This motivates us to *avoid clipping* around the clean sample. 2) We do not need to restrict the noise augmentation to lie inside an $\epsilon-$ball, since its strength is disentangled from that of the attack. Thus, we can use *stronger noise-augmentations* than previous methods.

These modifications lead to a novel adversarial training method that combines noise-based data augmentations with FGSM. We denote it as Noise-FGSM (N-FGSM). Following the notation introduced in Section 3, we define the noise augmented sample as $x_{\text{aug}} = x + \eta$ where η is sampled from a uniform distribution on $[-k, k]^d$ (where we can have $k > \epsilon$). Then the adversarially perturbed samples have the following form:

$$x_{\text{N-FGSM}} = x_{\text{aug}} + \alpha \cdot \text{sign}\big(\nabla_{x_{\text{aug}}} \mathcal{L}(f_\theta(x_{\text{aug}}), y)\big). \tag{3}$$

This construction corresponds to augmenting the clean sample x with the perturbation defined in Equation (2) where ψ is the identity map and Ω is the uniform ditribution spanning $[-k, k]^d$. We detail our full adversarial training procedure in Algorithm 1. In what follows, we analyse the effect of treating the noise as data augmentation as opposed to treating it as a random step within the attack. In particular, we show that clipping around the clean sample x (as done in RS-FGSM) can strongly degrade the robustness of the network. Moreover, we show that as we increase the ϵ radii of adversarial attacks, we need stronger noise perturbations than previously used to prevent CO.

Clipping around clean sample x hinders the effectiveness of perturbations. We analyse two variants, one where perturbations are clipped around the clean sample x (as done in previous methods) and another where no clipping is applied. In Figure 2 (left), we report the robust accuracy using PGD-50-10 (*i.e.,* PGD attack with 50 iterations and 10 restarts) with $\epsilon = 8/255$ and observe that clipping significantly degrades the effectiveness of FGSM training. To understand this drop, consider the following perturbations; (**1**) a baseline perturbation where we only use noise $\delta_{\text{random}} = \psi(\eta)$ and (**2**) a perturbation that combines noise with FGSM $\delta_{\text{full}} = \psi(\eta + \alpha \cdot \text{sign}\big(\nabla_x \mathcal{L}(f_\theta(x+\eta), y)\big))$. Moreover, we consider two cases in which we either define ψ as a clipping operator or as the identity. We define the effective FGSM step size as the magnitude corresponding to the ratio[1] $\alpha_{\text{effective}} = \|\delta_{\text{full}} - \delta_{\text{random}}\|_2 / \|x\|_2$

[1]The denominator $\|x\|_2$ is simply to normalize the ℓ_2-norm and be comparable to the FGSM step size α.

Algorithm 1 N-FGSM adversarial training

Inputs: epochs T, batches M, radius ϵ, step-size α (default: ϵ), noise magnitude k (default: 2ϵ).
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, M$ **do**
    $\eta \sim \text{Uniform}[-k, k]^d$
    $x^i_{\text{aug}} = x^i + \eta$ *// Augment sample with additive noise.*
    $x^i_{\text{N-FGSM}} = x^i_{\text{aug}} + \alpha \cdot \text{sign}\big(\nabla_{x^i_{\text{aug}}} \mathcal{L}(f_\theta(x^i_{\text{aug}}), y^i)\big)$ *// N-FGSM augmented sample.*
    $\nabla_\theta = \nabla_\theta \mathcal{L}(f_\theta(x^i_{\text{N-FGSM}}), y^i)$ *// Compute gradients of model's weights*
    $\theta = \text{optimizer}(\theta, \nabla_\theta)$ *// Standard weight update, (e.g., SGD)*

which measures the contribution of the FGSM step in the final perturbation compared to simply following the noise direction η. In Figure 2 (middle), we observe that the clipping operator reduces the effective magnitude of FGSM, thus, perturbations become more similar to only using random noise. On the other hand, without clipping we always take the full step in the FGSM direction. This highlights the trade-off between noise magnitude and attack strength discussed above.

Larger noise is also necessary to prevent CO. As discussed above, previous work did not investigate the effects of using noise perturbations potentially larger than the attack strength. However, we empirically find that increasing the noise magnitude is key to avoiding CO. In particular, as seen in Figure 2 (right), when no clipping is performed, it is crucial that we augment with larger noise magnitude in order to prevent CO in all settings. We find the noise magnitude of $k = 2\epsilon$ to work well in most of our experiments, however, a more extensive hyperparameter tuning might improve our results further.

Note that these results are contrary to previous intuitions: Andriushchenko and Flammarion [1] suggested that the random step in RS-FGSM is not important per se, arguing that its main role is reducing the ℓ_2 norm of the perturbations, so that the loss remains to be approximately locally linear. In contrast, N-FGSM perturbations are larger on expectation than those of RS-FGSM, while they do not suffer from CO (refer Appendix I). We believe that our findings will lead to a better understanding of the role of noise in avoiding CO in future work. Moreover, in Section 6 we conduct extensive analyses to show that, despite N-FGSM obtains larger perturbations, clean accuracy does not degrade and other methods do not benefit from simply increasing the strength of their attacks.

Why does noise augmentation avoid CO? Andriushchenko and Flammarion [1] found that after CO, the gradients of the loss with respect to the input around clean samples became strongly misaligned, which is a sign of non-linearity. Moreover, Kim et al. [14] showed that the loss surface of models suffering from CO appears distorted, *i.e.,* there is a sharp peak in the loss surface along the FGSM direction, which seems to render FGSM ineffective (observe from Figure 5 how after after CO, visually, FGSM perturbations change drastically). In order to prevent CO, GradAlign explicitly regularizes the loss surface so it remains linear. To investigate further, we plot the loss surface at the end of training for different methods (see Figure 12 in Appendix) and find that, while FGSM or RS-FGSM lead to a distorted loss, N-FGSM obtains a non-distorted loss surface similar to that obtained by GradAlign regularizer. Thus, it seems that adding strong noise-augmentations implicitly regularizes the loss landscape, leading to more effective single-step attacks.

5 Robustness Evaluations and Comparisons

We compare N-FGSM against several adversarial training methods, on a broad range of $\epsilon - l_\infty$ radii. Following Wong et al. [30], we evaluate adversarial robustness on CIFAR-10/100 [15] and SVHN [19] with PGD-50-10 attacks, using both PreactResNet18 [12] and WideResNet28-10 [31]. Evaluations with AutoAttack[5] are also in Appendix G.

5.1 Comparison against Single-Step Methods

We start by comparing N-FGSM against other single-step methods. Note that not all single-step methods are equally expensive, since they may involve more or less computationally demanding operations. For instance, GradAlign uses a regularizer that is considerably expensive, while MultiGrad requires evaluating input gradients on multiple random points. For a comparison of training costs of

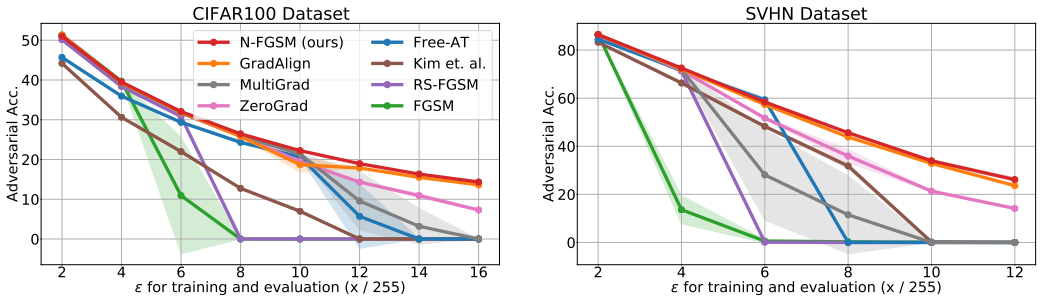


Figure 3: Comparison of single-step methods on CIFAR-100 (left) and SVHN (right) with PreactResNet18 over different perturbation radius (ϵ is divided by 255). Our method, N-FGSM, can match or surpass prior art results while *reducing the cost by a* $3\times$ *factor*. Adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds. Legend is shared among plots.

different single-step methods, we refer the reader to Figure 1 (right). We use RS-FGSM and Free-AT with the settings recommended by Wong et al. [30]. We apply GradAlign with hyperparameters reported in the official repository[2]. ZeroGrad and Kim et al. [14] do not have a recommended set of hyperparameters; for a fair comparison we ablate them and select the ones with highest adversarial accuracy (for every ϵ and dataset). We train on CIFAR-10/100 for 30 epochs and on SVHN for 15 epochs with a cyclic learning rate. Only for Free-AT, we use 96 and 48 epochs for CIFAR-10/100 and SVHN, respectively, to obtain comparable results following Wong et al. [30]. CIFAR-10 results are in Figure 1 (middle), whereas CIFAR-100 and SVHN are in Figure 3.

As observed in Figure 1 and Figure 3, FGSM and RS-FGSM suffer from CO for larger ϵ attacks on all reported datasets. For instance, RS-FGSM fails against attacks with $\epsilon = 8/255$ on CIFAR-10 and CIFAR-100 and against $\epsilon = 6/255$ on SVHN. With appropriate hyperparameters, ZeroGrad is able to consistently avoid CO. However, it obtains sub-par robustness compared to N-FGSM and GradAlign, especially against large ϵ attacks. Neither MultiGrad nor Kim et al. [14] avoid CO in all settings despite being more expensive. Free-AT also suffers from CO on all three datasets as also observed by Andriushchenko and Flammarion [1]. In contrast, N-FGSM avoids CO on all datasets, achieving comparable or superior robustness to GradAlign *while being 3 times faster.*

5.2 Comparison against Multi-Step Attacks

In Section 5.1, we compared the performance of single-step methods and observed that N-FGSM is able to match or surpass the state-of-the-art method, *i.e.,* GradAlign, while reducing the computational cost by a factor of 3. In this section, we compare the performance of N-FGSM against multi-step attacks. In particular, we compare against PGD-2 with $\alpha = \epsilon/2$ and PGD-10 with $\alpha = 2/255$, keeping the same training settings as described in Section 5.1. PGD-x denotes x iterations and no restarts.

In Figure 4, we observe that PGD-2, despite being a multi-step method, still suffers from CO for larger ϵ as opposed to our proposed N-FGSM. On the other hand, despite achieving comparable clean accuracies, there is a gap in adversarial accuracies between PGD-10, and other single-step methods that grows with perturbation size. This can be partially expected since the search space for adversaries grows exponentially with ϵ; and PGD, with more iterations, can explore it more thoroughly. Nevertheless, *computing a PGD-10 attack is* $10\times$ *more expensive to N-FGSM*. An important direction for future work would be addressing this gap and analysing, both theoretically and empirically, whether single-step methods can match the performance of their multi-step counterparts.

5.3 Analysis of Gradients and Adversarial Perturbations

To gain further insights into CO, we visually explore the perturbations generated with FGSM, RS-FGSM, N-FGSM, and PGD-10 attacks. We show that N-FGSM generates perturbations that exhibit behavior similar to PGD-10. In particular, for a given test sample, we average the adversarial perturbations (δ) and gradients across several epochs at the beginning of training (Epoch 2 to 8) and at the end (Epoch 24 to 30) and visualise them in Figure 5 (see also Figure 11 in Appendix for

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

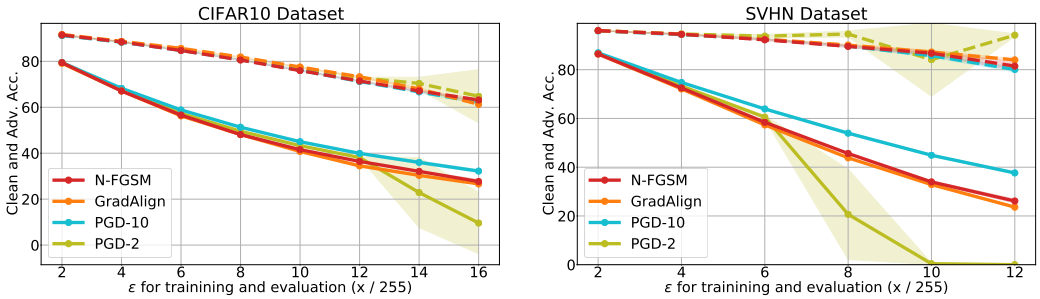


Figure 4: Comparison of N-FGSM and GradAlign with multi-step methods on CIFAR-10 (Left) and SVHN (Right) with PreactResNet18 over different perturbation radii (ϵ is divided by 255). Despite all methods achieving comparable clean accuracy (dashed lines), there is a gap in robust accuracy between PGD-10 and single-step methods. However, note that PGD-10 is $10\times$ more expensive than N-FGSM. Adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds.

more examples). We observe that early in training all methods generate consistent and interpretable δ. However, after CO, FGSM and RS-FGSM generate δ that are harder to interpret, idem for their gradients. On the other hand, we observe that N-FGSM provides consistent and interpretable δ throughout training, similar to those generated by PGD-10. This provides further evidence that N-FGSM enjoys similar properties to the more expensive PGD-10 training.

Figure 5 analyzes the gradients and δ throughout the test set. Aside from loosing interpretability, post-CO the gradient norm increases by several orders of magnitude for FGSM and RS-FGSM while it remains low for N-FGSM and PGD-10. We also compute the effective rank[3] of δ for each example before and after CO to measure the consistency of δ before and after CO. We consider three training intervals, (Epoch 2 to 8): before CO for all methods; (Epoch 16 to 22): after FGSM suffers CO but not RS-FGSM; (Epoch 24 to 30): after both FGSM and RS-FGSM suffer CO. Prior to CO, PGD-10 has a larger effective rank (*i.e.,* the perturbations span a larger subspace) than FGSM and RS-FGSM. N-FGSM has the highest effective rank, arguably due to the higher noise magnitude. Note that RS-FGSM, which has a smaller noise magnitude and clipping, also has a larger effective rank than FGSM, however, the difference is much lower. When either FGSM or RS-FGSM suffer from CO, the effective rank of their δ increases significantly above that of PGD-10 and N-FGSM. This would suggest that δ loose consistency after CO and is aligned with our visualizations in Figure 5. All of these show properties of δ and gradients that are consistent across methods (N-FGSM and PGD) that avoid CO and different from methods like RS-FGSM and FGSM, which suffer from CO.

6 Increasing Adversarial Perturbations

In Section 4, we observed that removing clipping and increasing the noise magnitude were both necessary for the improved performance of N-FGSM. However, as discussed in Theorem I.2 this will result in an increase of the squared norm of the training perturbation $\delta_{\mathrm{N-FGSM}}$ as compared to FGSM. In this section, we perform further ablations to corroborate that it is indeed the increase in noise magnitude – and not the mere increase of the perturbation's magnitude – that helps to stabilize N-FGSM.

Increasing α alone is not sufficient. N-FGSM combines a noise perturbation with an FGSM step. Thus, we can increase the perturbation magnitude by increasing any of the two. This begs the question: Is it sufficient to increase the N-FGSM step-size α to avoid CO without adding any noise? We observe in Figure 6 (A) that training without noise (essentially, FGSM) leads to CO, with robust accuracy equal to zero, even for large values of α. This indicates that it is not an increase in the perturbation norm, but the combination with noise which plays an essential role in circumventing CO for N-FGSM.

Increasing α requires adjusting the noise magnitude. As observed in Figure 6 (A), increasing α for N-FGSM leads to CO if the noise magnitude is not large enough. For example, while a noise magnitude $k = 1\epsilon$ and an adversarial step size $\alpha = 1.25\epsilon$ yield a robust accuracy of 49.68%, increasing α to 1.5ϵ while keeping the same noise magnitude results in CO – with robust accuracy equal to zero. This further suggests that an increase in the adversarial step-size α requires

[3]We compute effective rank as the number of singular vectors required to explain 90% of the variance.

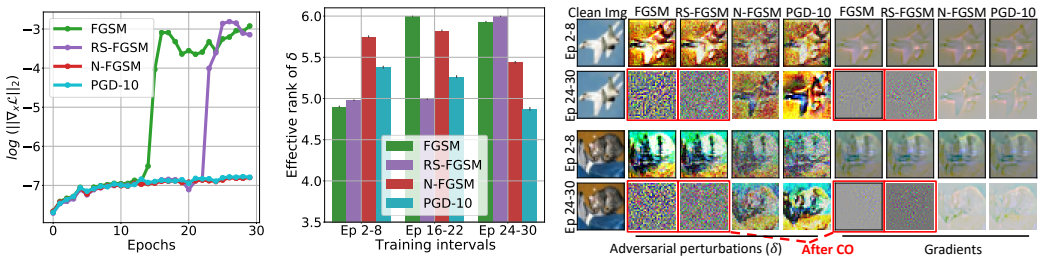


Figure 5: **Left:** Mean ℓ_2 norm of per-sample-gradients across all test set samples. After CO, both FGSM and RS-FGSM gradients increase by several orders of magnitude. **Middle:** Effective rank of perturbations (δ) across three training intervals *Ep 2-8* before CO for all methods; *Ep 16-22*: after FGSM presents CO but not RS-FGSM; *Ep 24-30*: after both FGSM and RS-FGSM had CO. **Right:** Visualization of δ and gradients averaged across several epochs at the beginning (top) and end (bottom) of training. After CO, FGSM and RS-FGSM obtain δ and gradients that are hard to interpret.

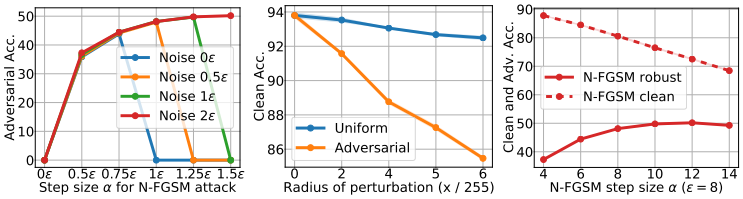
a commensurate increase in the noise magnitude. We find that setting the noise magnitude $k = 2\epsilon$ works well for most settings.

Larger noise perturbations preserve clean accuracy. Increasing the norm of training perturbations by increasing α results in a drop in the clean accuracy (discussed later in Section 7). This has also been observed in prior works [30]. However, we show in Figure 4 that the clean accuracy for N-FGSM is similar to that of GradAlign, despite the magnitude of the perturbations being larger. We ablate the effects of adversarial and noise perturbations on the clean accuracy in Figure 6 (B): we observe that augmenting training samples with noise alone (*i.e.,* $\alpha = 0$) has a much milder effect on the clean accuracy than augmenting in an adversarial direction. In general, increasing noise is more forgiving on the clean accuracy than increasing the adversarial step size. This is not surprising, considering that moving in random directions along the input space has a significantly lower impact on the loss than moving along the FGSM direction (see Figure 12 in the Appendix) and that training with noise alone does not provide any significant robustness against larger attacks (for a more detailed ablation, see Appendix Figure 14).

Other methods do not benefit from larger training ϵ. As previously mentioned, N-FGSM perturbations have $\ell_\infty-$norm larger than ϵ. We have seen that the benefits of N-FGSM can not be reproduced by simply increasing α without increasing the noise. However, for the sake of completeness, we also ablate other single-step baselines by using a larger ϵ during training, while testing with a fixed $\epsilon = {}^{8}/_{255}$ on CIFAR10. We observe that increasing ϵ_{train} seems to lead to a decrease in robustness for most methods; for instance, PGD-50-10 accuracy for RS-FGSM drops from 46.08 ± 0.18 when training with $\epsilon = {}^{8}/_{255}$ to 0.0 ± 0.0 with $\epsilon = {}^{12}/_{255}$. In two cases (GradAlign and MultiGrad), we observe a small increase, with the highest increase being for GradAlign, which improves from 48.14 ± 0.15 to 50.6 ± 0.45; yet, the clean accuracy drops from 81.9 ± 0.22 to 73.29 ± 0.23. This is similar to increasing α for N-FGSM (see Figure 6 (C)). However, this is tied to a significant degradation of clean accuracy. All in all, taking into account both clean and robust accuracy, we conclude that all single-step baselines suffer from either CO or a severe degradation in their clean accuracy when increasing the training ϵ. Full results are presented in Table 1 in Appendix.

7 Additional Ablations

Hyperparameter selection. While FGSM relies on a fixed step-size (*i.e.,* $\alpha = \epsilon$), Wong et al. [30] explored different values of α for RS-FGSM, finding that an increase of the step-size improves the adversarial accuracy – up to a point where CO occurs. We also ablate the value of α for N-FGSM in Figure 6 (C). We find that by increasing the noise magnitude, N-FGSM can use larger α values than RS-FGSM, without suffering from CO. This leads to an increase in the adversarial accuracy at the expense of a decrease in the clean accuracy. In light of this trade-off, we also use $\alpha = \epsilon$ for N-FGSM. Regarding the noise hyperparameter k, we find that $k = 2\epsilon$ works in all but one SVHN experiment ($\epsilon = 12$, in which we set $k = 3\epsilon$). In comparison, GradAlign regularizer hyperparameter or ZeroGrad quantile value need to be modified for every radius with a noticeable shift between CIFAR-10 and SVHN hyperparameters, suggesting they may require additional tuning when applied to novel datasets.



Comparison of Training Schedules

Clean Acc	Robust Acc
Long schedule: Final model	
83.18 ± 0.11	36.56 ± 0.26
Long schedule: Best model	
80.8 ± 0.36	**48.48 ± 0.27**
Fast schedule: Final model	
80.58 ± 0.22	48.12 ± 0.07

Figure 6: Different ablations on N-FGSM parameters and training schedule. From left to right: **A:** Adversarial accuracy when varying step-size α and noise magnitude k ($\epsilon = 8$). Increasing α does not suffice to prevent CO, we must also increase the noise magnitude. **B:** Clean accuracy after training with random or adversarial perturbations. With comparable radius, random perturbations have a much milder effect than adversarial. **C:** Ablation of step size α in N-FGSM $\epsilon = 8, k = 2\epsilon$. As we increase the magnitude of the FGSM perturbation we observe an increase in robustness coupled with a drop on the clean accuracy. **D**: Comparison of the "fast" training schedule from [30] and "long" training schedule described in [22]. N-FGSM shows robust oberfitting but not CO with the long schedule. Adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds.

Long vs fast training schedules. Throughout our experiments, we used the RS-FGSM training setting introduced in [30]. However, Rice et al. [22] suggest that a longer training schedule coupled with early stopping may lead to a boost in performance. Kim et al. [14] and Li et al. [17] report that longer training schedules increase the chances of CO for RS-FGSM and that this limits its performance. We test the longer training schedule with N-FGSM and find that it does not suffer from CO. However, it does suffer from *robust overfitting*, *i.e.,* adversarial accuracy on the training set is larger than on the test set as described in [22] for PGD-10. Notice the difference between the robust accuracy of the final and best models in Figure 6 (D). Interestingly, although we observe a slight increase in performance when using the long training schedule, we find the fast training schedule to be remarkably competitive. See more results in Appendix D, including a comparison to GradAlign.

Experiments with WideResNet28-10. We also compare the performance of all methods on WideResNet28-10 [31] architecture in Figure 8 and Figure 9 in Appendix. As in the experiments with PreActResNet18, N-FGSM obtains the best performance/cost trade-off. We had to increase the regularizer hyperparameter for GradAlign (compared to the settings for PreActResNet18) in order to prevent CO on CIFAR-100 and, to our surprise, *we could not find a competitive hyperparameter setting for GradAlign on the SVHN dataset* for $\epsilon \geq 6$. We tried both increasing the regularizer hyperparameter and decreasing the step size α, but some or all runs led to models close to a constant classifier for each setting. We do not claim that GradAlign will not work, but finding a good configuration might require further tuning. The default configuration for N-FGSM ($\alpha = \epsilon,\ k = 2\epsilon$) works well in all settings except for $\epsilon = 16$ on CIFAR-10 and $\epsilon = 10,\ 12$ on SVHN. For CIFAR-10, we increase the noise magnitude to $k = 4\epsilon$. For SVHN, we find that decreasing α works better than increasing the noise. In both cases, N-FGSM yields nontrivial adversarial accuracy.

8 Conclusion

In this work, we explore the role of noise and clipping in single-step adversarial training. Contrary to previous intuitions, we show that increasing the noise magnitude and removing the $\epsilon - \ell_\infty$ constraint leads to an improvement in adversarial robustness while maintaining a competitive clean accuracy. These findings led us to propose N-FGSM, a simple and effective approach that can match or surpass the performance of GradAlign [1], while achieving a $3\times$ speed-up.

We perform an extensive comparison with other relevant single-step methods, observing that all of them achieve sub-optimal performance and most of them are not able to avoid CO for larger ϵ attacks. Moreover, we also analyze gradients and adversarial perturbations during training and observe that they have a similar behaviour for N-FGSM and PGD-10 as opposed to other methods that present CO such as FGSM and RS-FGSM. However, despite impressive improvements of single-step adversarial training methods, there is still a gap between single-step and multi-step methods such as PGD-10 as we increase the ϵ radius. Therefore, future work should put an emphasis on formally understanding the limitations of single-step adversarial training and explore how, if possible, this gap can be reduced.

References

[1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[2] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 2018.

[3] Adith Boloor, Xin He, Christopher D. Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Simple physical adversarial examples against end-to-end autonomous driving models. *arxiv:1903.05157*, 2019.

[4] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2017.

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020.

[6] Jacob Devlin, Ming-Wei Changm, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, 2019.

[7] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In *International Conference on Machine Learning (ICML)*, 2019.

[9] Zeinab Golgooni, Mehrdad Saberi, Masih Eskandar, and Mohammad Hossein Rohban. Zerograd: Mitigating and explaining catastrophic overfitting in fgsm adversarial training. *arXiv:2103.15476*, 2021.

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[13] Peilin Kang and Seyed-Mohsen Moosavi-Dezfooli. Understanding catastrophic overfitting in adversarial training. *arXiv:2105.02942*, 2021.

[14] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

[17] Bai Li, Shiqi Wang, Suman Jana, and Lawrence Carin. Towards understanding fast adversarial training. *arXiv:2006.03089*, 2020.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems (NeurIPS), Workshops*, 2011.

[20] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE symposium on security and privacy (SP)*, 2016.

[21] Geon Yeong Park and Sang Wan Lee. Reliably fast adversarial training via latent adversarial perturbation. In *International Conference on Learning Representations (ICLR), Workshops*, 2021.

[22] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning (ICML)*, 2020.

[23] Amartya Sanyal, Varun Kanade, and Philip H. S. Torr. Robustness via deep low-rank representations. *arxiv:1804.07090*, 2018.

[24] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Neural Information Processing Systems (NeurIPS)*, 2019.

[25] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[27] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018.

[28] BS Vivek and R Venkatesh Babu. Single-step adversarial training with dropout scheduling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[29] Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning (ICML)*, 2018.

[30] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.

[31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC British Machine Vision Conference (BMVC)*, 2016.

[32] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We provide experimental validation for all our claims.
   (b) Did you describe the limitations of your work? [Yes] Yes, in conclusions and when appropriate along the paper.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not consider research on training robust networks can have negative impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code will be released upon acceptance.
   (b) Did you specify all the training details (*e.g.,* data splits, hyperparameters, how they were chosen)? [Yes] When relevant in the text.

(c) Did you report error bars (*e.g.,* with respect to the random seed after running experiments multiple times)? [Yes] All plots or numbers have standard deviation. Although sometimes too small to be visible.

(d) Did you include the total amount of compute and the type of resources used (*e.g.,* type of GPUs, internal cluster, or cloud provider)? [Yes] Yes, see appendix M.

4. If you are using existing assets (*e.g.,* code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]