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

Pau de Jorge *
University of Oxford
NAVER LABS Europe

Adel Bibi
University of Oxford

Riccardo Volpi
NAVER LABS Europe

Amartya Sanyal
ETH Zürich
ETH AI Center

Philip H. S. Torr
University of Oxford

Grégory Rogez
NAVER LABS Europe

Puneet K. Dokania
University of Oxford
Five AI Ltd.

Abstract

Recently, Wong et al. [35] 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. Code can be found in `https://github.com/pdejorge/N-FGSM`

1 Introduction

Deep neural networks have achieved remarkable performance on a variety of tasks [12, 27, 7]. However, it is well known that they are vulnerable to small worst-case perturbations around the input data – commonly referred to as *adversarial examples* [31]. The existence of such adversarial examples poses a security threat to deploying models in sensitive environments [2, 4]. This has motivated a large body of work towards improving the *adversarial robustness* of neural networks [11, 22, 32, 25, 5].

The most popular family of methods for learning robust neural networks is based on the concept of *adversarial training* [11, 20]. 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 [34] and hence, different *adversarial attacks* have been suggested that approximate them. In their seminal work, Goodfellow et al. [11] 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. [32] showed that these models are still vulnerable to multi-step attacks, namely those allowed to perform multiple gradient ascent steps. Given their better (robust)

*Correspondence to `pau@robots.ox.ac.uk`

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

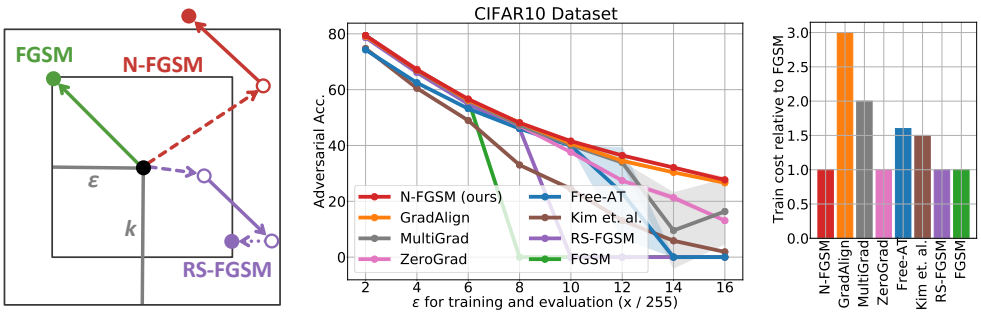


Figure 1: **Left:** Visualization of FGSM [11], RS-FGSM [35] and N-FGSM (ours) attacks. While RS-FGSM is limited to noise in the $\epsilon - l_\infty$ ball, N-FGSM draws noise from an arbitrary $k - l_\infty$ ball. Moreover, N-FGSM does not clip the perturbation around the clean sample. **Middle:** Comparison 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 Q for details.

performance, multi-step attacks such as *Projected Gradient Descent* (PGD) [20] 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 [35, 26, 33]. In particular, Wong et al. [35] studied 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 [37, 18, 20]. 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. [11] first proposed FGSM; Tramèr et al. [32] 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. [26] 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. [35] 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 [10, 15, 33, 23, 19].

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 [14], 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 [20]. 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. [11] leveraged this to propose FGSM, where the adversarial perturbation follows the direction of the sign of the gradient. Tramèr et al. [32] 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. [35] 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. [32]. 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.

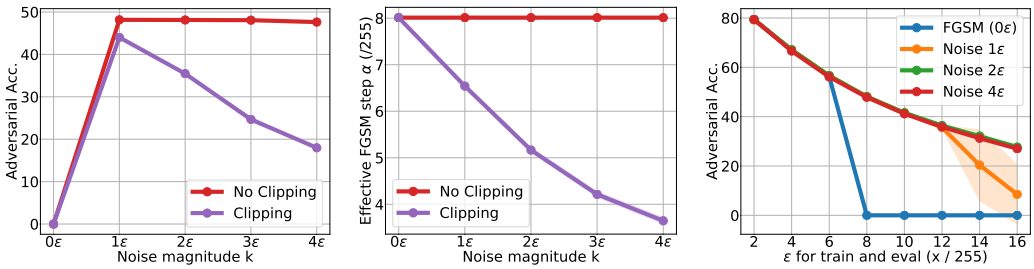


Figure 2: **Left:** Ablation of clipping vs not clipping around the clean sample x for $\epsilon = 8/255$. Clipping leads to a significant drop in robustness which increases with the strength of the noise augmentations. **Middle:** Analysis of the effective FGSM step size after clipping. We observe that 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.

4 Noise and FGSM

Previous methods that combined noise with FGSM, *e.g.,* R+FGSM [32] and RS-FGSM [35], have considered the noise as a *random step* integrated within the attack. Since it is a common practice 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 per-

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)*

turbation 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[2] $\alpha_{\text{effective}} = {\|\delta_{\text{full}} - \delta_{\text{random}}\|_2}/{\|x\|_2}$ 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 N). 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. [15] 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 14 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. This aligns with previous work that theoretically link noise augmentations with a regularizer that encourages Lipschitzness [3].

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. [35], we evaluate adversarial robustness on CIFAR-10/100 [16] and SVHN [21] with PGD-50-10 attacks, using both PreactResNet18 [13] and WideResNet28-10 [36]. Evaluations with AutoAttack[6] are also in Appendix G.

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

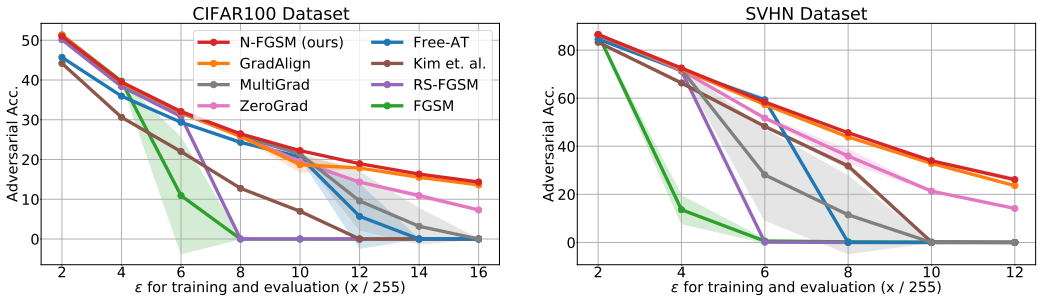


Figure 3: Comparison of single-step methods on CIFAR-100 (left) and SVHN (right) with PreactResNet18 over different perturbation radius (ϵ is divided by 255). Our method, N-FGSM, can match or surpass prior art results while *reducing the cost by a* $3\times$ *factor*. Adversarial accuracy is based on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

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 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. [35]. We apply GradAlign with hyperparameters reported in the official repository[3]. ZeroGrad and Kim et al. [15] 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. [35]. 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. [15] 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.

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

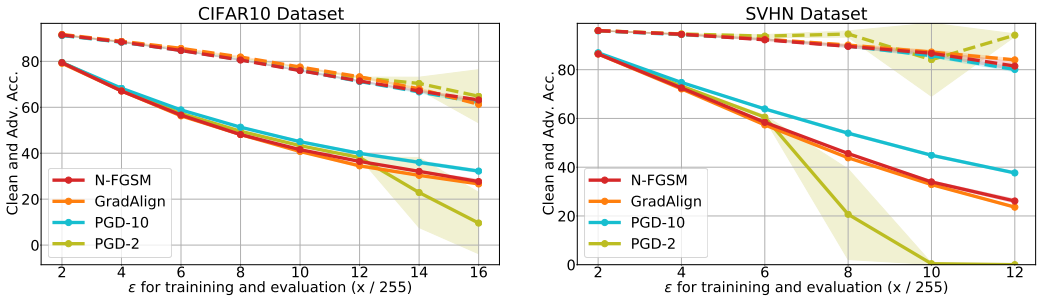


Figure 4: Comparison of N-FGSM and GradAlign with multi-step methods on CIFAR-10 (Left) and SVHN (Right) with PreactResNet18 over different perturbation radii (ϵ is divided by 255). Despite all methods achieving comparable clean accuracy (dashed lines), there is a gap in robust accuracy between PGD-10 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.

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 12 in Appendix for more examples). We observe that, during early stages in training all, methods generate consistent and interpretable δ. However, after CO, FGSM and RS-FGSM generate δ that are harder to interpret, similarly to 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[4] 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 N.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

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

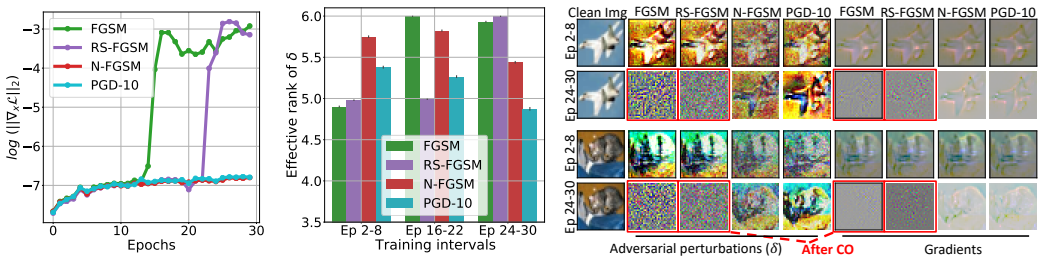


Figure 5: **Left:** Mean ℓ_2 norm of per-sample-gradients across all test set samples. After CO, both FGSM and RS-FGSM gradients increase by several orders of magnitude. **Middle:** Effective rank of perturbations (δ) across three training intervals *Ep 2-8* before CO for all methods; *Ep 16-22*: after 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.

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.

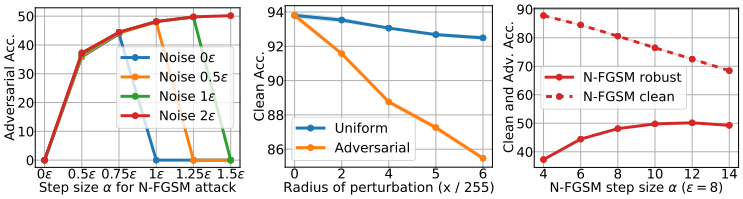
Increasing α requires adjusting the noise magnitude. As observed in Figure 6 (A), increasing α for N-FGSM leads to CO if the noise magnitude is not large enough. For example, while a noise magnitude $k = 1\epsilon$ and an adversarial step size $\alpha = 1.25\epsilon$ yield a robust accuracy of 49.68%, 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 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 [35]. 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 14 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 16).

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 2 in Appendix.

7 Additional Ablations

Hyperparameter selection. While FGSM relies on a fixed step-size (*i.e.,* $\alpha = \epsilon$), Wong et al. [35] 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-



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 [35] and "long" training schedule described in [24]. 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.

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.

Long vs fast training schedules. Throughout our experiments, we used the RS-FGSM training setting introduced in [35]. However, Rice et al. [24] suggest that a longer training schedule coupled with early stopping may lead to a boost in performance. Kim et al. [15] and Li et al. [19] 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 [24] 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 [36] 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.

Experiments on Imagenet. We present results on the Imagenet dataset [17] in Table 1. Due to the high computational demands of Imagenet training and testing we focus on the main baselines of comparable cost to FGSM. Namely FGSM, RS-FGSM and N-FGSM. We observe that FGSM presents CO for $\epsilon = 6/255$ while neither RS-FGSM nor N-FGSM present CO. However, N-FGSM has better robustness. For instance, at $\epsilon = 6/255$ N-FGSM obtains PGD50-10 accuracy of 17.12% while RS-FGSM yields 16.5% and FGSM 0.08% (due to CO). Thus, N-FGSM also avoids CO in ImageNet, improving robustness over same-cost baselines. For experimental details refer to Appendix L.

Combining N-FGSM with additional regularizers. Recent works [29, 30] have been proposed to improve performance of single step methods at moderate perturbation radii $\epsilon = 8/255$. However, we observe that with the default settings (which use a version of RS-FGSM with Bernoulli noise) they lead to CO for larger ϵ. Then we compare them with N-FGSM + Regularizer where we apply

Table 1: Clean accuracy (top) and PGD50-10 accuracy (bottom) of N-FGSM and other same-cost baselines on Imagenet dataset. We observe that FGSM presents CO for $\epsilon = 6/255$ while both RS-FGSM and N-FGSM avoid CO. N-FGSM has consistently better robustness than baselines.

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$
FGSM	54.72	48.50	48.55
	38.21	25.86	0.08
RS-FGSM	56.29	50.81	47.67
	36.86	25.12	16.49
N-FGSM	54.39	47.56	47.70
	38.07	26.28	17.12

their proposed regularizers to N-FGSM. If we apply GAT or NuAT regularizers to N-FGSM then we do not observe CO and usually a boost in performance. For instance, at $\epsilon = 10/255$, GAT has a robust acc (with PGD50-10) of 43.34 $\pm$ 0.23 while N-FGSM+GAT regularizer obtains 44.97 $\pm$ 0.07, in comparison plain N-FGSM has 41.56 $\pm$ 0.16. This is extremely compelling as it suggests N-FGSM can be combined with other regularizers designed to improve FGSM performance and mutually benefit each other. Full results of the comparison are presented in Table 6 in Appendix K

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.

Acknowledgments and Disclosure of Funding

We thank Guillermo Ortiz-Jiménez for the fruitful discussions and feedback. This work is supported by the UKRI grant: Turing AI Fellowship EP/W002981/1 and EPSRC/MURI grant: EP/N019474/1. We would also like to thank the Royal Academy of Engineering and FiveAI. A. Sanyal acknowledges support from the ETH AI Center postdoctoral fellowship. Pau de Jorge is fully funded by NAVER LABS Europe.

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] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

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

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

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

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

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems (NeurIPS)*, 2012.

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

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

[20] 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.

[21] 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.

[22] 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.

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

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

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

[26] 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.

[27] 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.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[29] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances in Neural Information Processing Systems*, 33:20297–20308, 2020.

[30] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Towards efficient and effective adversarial training. *Advances in Neural Information Processing Systems*, 34:11821–11833, 2021.

[31] 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.

[32] 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.

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

[34] 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.

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

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

[37] 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] We include the URL to our code in the abstract.

(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 R.

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? [Yes] We include the URL to our code in the abstract.

(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]

A Additional plots for PreActResNet18 experiments

In the main paper we compare N-FGSM with other single-step methods and multi-step methods separately and remove clean accuracies for better visualization. In this section we present the curves for all methods with both the clean and robust accuracy. The tendency in the three datasets is for N-FGSM PGD-50-10 accuracy to be slightly above that of GradAlign, while the opposite happens to the clean accuracy. We also observe that clean accuracy becomes significantly more noisy when CO happens. Exact numbers for all the curves are in Appendix S.

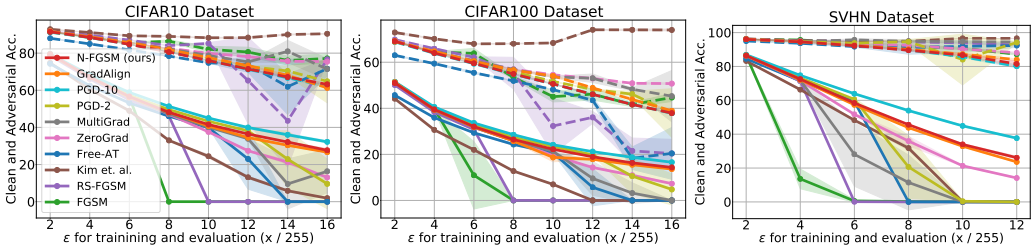


Figure 7: Comparison of all methods on CIFAR-10, CIFAR-100 and SVHN with PreactResNet18 over different perturbation radius (ϵ is divided by 255). We plot both the robust (solid line) and the clean (dashed line) accuracy for each method. Our method, N-FGSM, is able to match or surpass the state-of-the-art single-step method GradAlign 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 all plots.

B Experiments with WideResNet28-10 architecture

In this section we present the plots of our experiments with WideResNet28-10. We report the results in two figures. In Figure 8 we compare all single-step methods and we do not plot the clean accuracy for better visualization. In Figure 9 we plot all methods, including multi-step methods, and report the clean accuracy as well with dashed lines. Since we observed that our baseline, RandAlpha, outperformed [15] in all settings for PreActResNet18, we only report RandAlpha for WideResNet. As mentioned in the main paper, we observe that CO seems to be more difficult to prevent for WideResNet. In particular, for GradAlign we observed the regularizer hyperparameter settings proposed by [1] for CIFAR-10 (searched for a PreActResNet18) worked well. However, those parameters led to CO for $6 \leq \epsilon \leq 12$ in CIFAR-100. Since $\epsilon = 14,\ 16$ did not show CO, we increased the GradAlign regularizer hyperparameter λ for CIFAR-100 so that each $6 \leq \epsilon \leq 12$ would have the default value corresponding to $\epsilon + 2$, for instance, λ for $\epsilon = 6$ would be the default λ in [1] for $\epsilon = 8$.

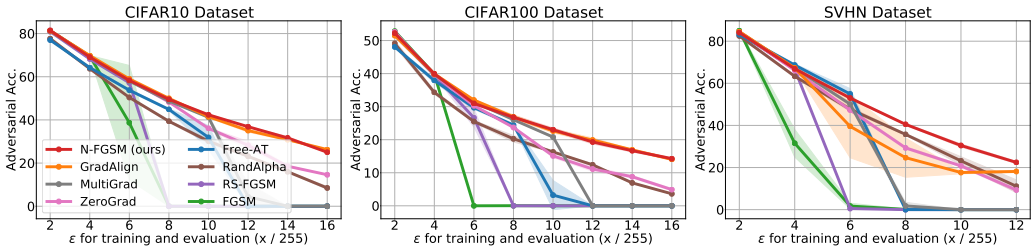


Figure 8: Comparison of single-step methods on CIFAR-10, CIFAR-100 and SVHN with WideResNet28-10 over different perturbation radius (ϵ is divided by 255). Our method, N-FGSM, is able to match or surpass the state-of-the-art single-step method GradAlign while *reducing the cost by a* $3\times$ *factor*. Moreover, we could not find any competitive hyperparameter setting for GradAlign for $\epsilon \geq 6$ in SVHN dataset. Adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds. Legend is shared among all plots.

For SVHN we observed that the default values for λ led to models close to a constant classifier for $\epsilon \geq 6$. We tried to increase the lambda for those ϵ values to 1.25λ but observed the same result. Since

the model did not show typical CO but rather it seemed as it was underfitting, we tried to reduce the step-size to $\alpha = 0.75\epsilon$ and also both decreasing α and increasing λ. When reducing the step size we obtain accuracies above those of a constant classifier for some radii, however, some or all seeds converge to a constant classifier for each setting, hence the large standard deviations. For N-FGSM, the default configuration of 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 as we tried for GradAlign works better than increasing the noise. We use $\alpha = 8$ for both ϵ radii. Exact numbers for all the curves are in Appendix S

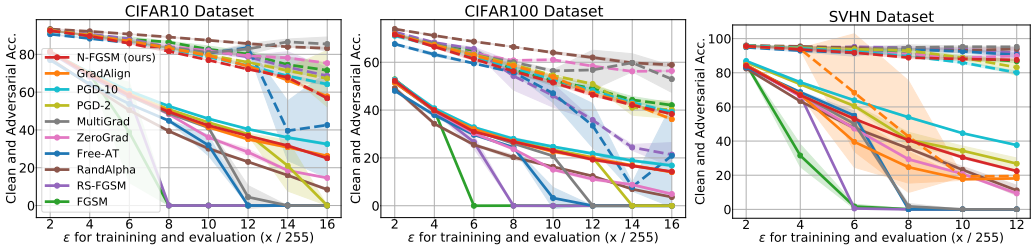


Figure 9: Comparison of all methods on CIFAR-10, CIFAR-100 and SVHN with WideResNet28-10 over different perturbation radius (ϵ is divided by 255). We plot both the robust (solid line) and the clean (dashed line) accuracy for each method. Legend is shared among all plots.

C Increasing adversarial perturbations during training

As mentioned in the main paper, N-FGSM perturbations have $\ell_\infty-$norm larger than ϵ, see Appendix N. In Section 6 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 *i.e.,* $\{\epsilon = 8/255, \epsilon = 12/255, \epsilon = 16/255\}$ while testing with a fixed $\epsilon = 8/255$ on CIFAR-10. Results are presented in Table 2. We observe that increasing ϵ_{train} seems to lead to a decrease in robustness for most methods, *e.g.,* PGD-50-10 accuracy for RS-FGSM goes 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, highest increase is for GradAlign which goes from 48.14 ± 0.15 to 50.6 ± 0.45, however, 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 all baselines perform best without increasing the training ϵ. All ablation results are presented in Table 2.

Table 2: Ablation of the PGD-50-10 accuracy for single-step methods when increasing the ϵ_{train}. All models are evaluated with PGD-50-10 attack and $\epsilon_{\text{test}} = 8/255$. Note that considering the trade-off between clean and robust accuracy, all methods perform best when training with the same epsilon to be applied at test time.

	$\epsilon_{\text{train}} = 1\epsilon_{\text{test}}$		$\epsilon_{\text{train}} = 1.5\epsilon_{\text{test}}$		$\epsilon_{\text{train}} = 2\epsilon_{\text{test}}$		
Method	**Clean acc.**	**PGD acc.**	**Clean acc.**	**PGD acc.**	**Clean acc.**	**PGD acc.**	**Rel. Cost**
GradAlign	81.9 ± 0.22	48.14 ± 0.15	73.29 ± 0.23	50.6 ± 0.45	61.3 ± 0.15	46.67 ± 0.29	3
MultiGrad	82.33 ± 0.14	47.29 ± 0.07	75.28 ± 0.2	50.0 ± 0.79	71.42 ± 5.63	0.0 ± 0.0	2
AT Free	78.41 ± 0.18	46.03 ± 0.36	73.91 ± 4.19	32.4 ± 22.91	71.64 ± 3.89	0.0 ± 0.0	1.6
Kim et. al.	89.02 ± 0.1	33.01 ± 0.09	88.35 ± 0.31	27.36 ± 0.31	90.45 ± 0.08	9.28 ± 0.12	1.5
FGSM	86.41 ± 0.7	0.0 ± 0.0	80.6 ± 2.59	0.0 ± 0.0	77.14 ± 2.46	0.0 ± 0.0	1
RS-FGSM	84.05 ± 0.13	46.08 ± 0.18	65.22 ± 23.23	0.0 ± 0.0	76.66 ± 0.38	0.0 ± 0.0	1
ZeroGrad	82.62 ± 0.05	47.08 ± 0.1	78.11 ± 0.2	46.43 ± 0.37	75.42 ± 0.13	45.63 ± 0.39	1
N-FGSM	80.58 ± 0.22	48.12 ± 0.07	71.46 ± 0.14	50.23 ± 0.31	63.18 ± 0.49	46.46 ± 0.1	1

D Longer training schedule

In our experiments, we have followed the "fast" training schedule introduced by [35]. However, [24] suggest that a longer training schedule coupled with early stopping may lead to a boost in performance. We also use the long training schedule for N-FGSM and observe that it does not lead to CO. In Table 3 we compare the performance of N-FGSM and GradAlign for the long training schedule. We observe that GradAlign does not seem to benefit from the long training schedule. On the other hand, although N-FGSM seems to obtain a slight increase in performance, the "fast" schedule provides comparable performance. It is worth mentioning that for GradAlign, the default regularizer hyperparameter for $\epsilon = 8/255$ and CIFAR-10 ($\lambda = 0.2$) does not prevent CO. We do a hyperparameter search and keep the value with the largest final robust accuracy ($\lambda = 0.632$).

Table 3: Comparison of "long" [24] and "fast" [35] training schedules for N-FGSM and GradAlign. GradAlign does not seem to benefit from the long training schedule. Although N-FGSM seems to obtain a slight increase in performance, the "fast" schedule provides comparable performance.

N-FGSM		**Grad Align**	
Clean Acc	**Robust Acc**	**Clean Acc**	**Robust Acc**
Long schedule: Final model			
83.18 $\pm$ **0.11**	36.56 $\pm$ 0.26	**84.13** $\pm$ **0.24**	36.17 $\pm$ 0.19
Long schedule: Best model			
80.8 $\pm$ 0.36	**48.48** $\pm$ **0.27**	81.57 $\pm$ 0.44	47.86 $\pm$ 0.1
fast schedule: Final model			
80.58 $\pm$ 0.22	48.12 $\pm$ 0.07	81.9 $\pm$ 0.22	**48.14** $\pm$ **0.15**

In Table 3 we observe that the performance of the final model is lower than that of an early stopped method. This could be expected due to the phenomena of robust overfitting described in [24]. However, as a sanity check to make sure that this is not due to a hidden CO during the long schedule which somehow the model recovers from we plot the full training history in Figure 10. There we can observe that *for N-FGSM there is no CO during training*.We also show FGSM (which is well known has CO for $\epsilon = 8/255$) for comparison.

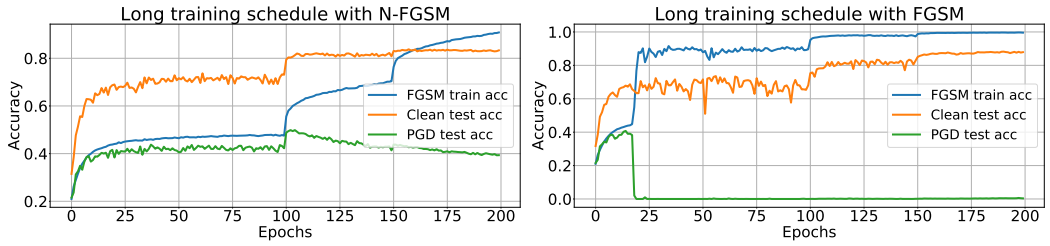


Figure 10: Training and test accuracy during the long training schedule proposed in [24]. We observe that N-FGSM (left) does not present CO at any point during training, however suffers from robust overfitting as described in [24] which suggested selecting the best validated model as a simple and yet effective way to improve robustness. On the other hand FGSM (right) suffers from CO where the robustness drops suddenly to 0 and does not recover.

E Randomized Alpha

Kim et al. [15] evaluate intermediate points along the RS-FGSM direction in order to pick the "optimal" perturbation size. However, we find that increasing the number of intermediate evaluated points does not necessarily lead to increased adversarial accuracy. Moreover, for large perturbations we could not prevent CO even with twice the number of evaluations tested by [15]. This motivates

us to test a very simple baseline where instead of evaluating intermediate steps, the RS-FGSM perturbation size is randomly selected as: $\delta = t \cdot \delta_{\text{RS-FGSM}}$ where $t \sim \mathcal{U}[0,1]^d$. Interestingly, as reported in Figure 11, we find that this very simple baseline, dubbed *RandAlpha*, is able to avoid CO for all values of ϵ and outperforms [15] on CIFAR-10, CIFAR-100 and SVHN. This is aligned with our main finding that combining noise with adversarial attacks is indeed a powerful tool that should be explored more thoroughly before developing more expensive solutions.

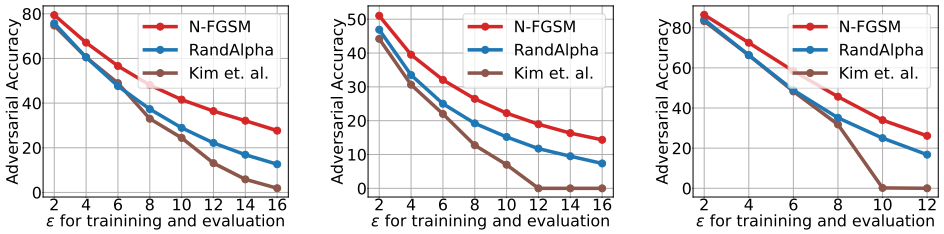


Figure 11: Comparison of [15] with RandomAlpha, our baseline where we multiply the RS-FGSM perturbation by a scalar uniformly sampled in $[0,1]$. We present results on CIFAR-10 (Left), CIFAR-100 (Middle) and SVHN (Right) with PreActResNet18.

F Further visualizations of adversarial perturbations and gradients

In this section we present an extension of Figure 5 with further examples. As observed in the main paper, early in training adversarial perturbations (δ) and gradients are consistent across epochs, however, after CO they become hard to interpret. Note that although we label rows as either pre-CO or post-CO we only observe CO for FGSM and RS-FGSM. Both PGD-10 and N-FGSM obtain robust models as shown in detail in the paper.

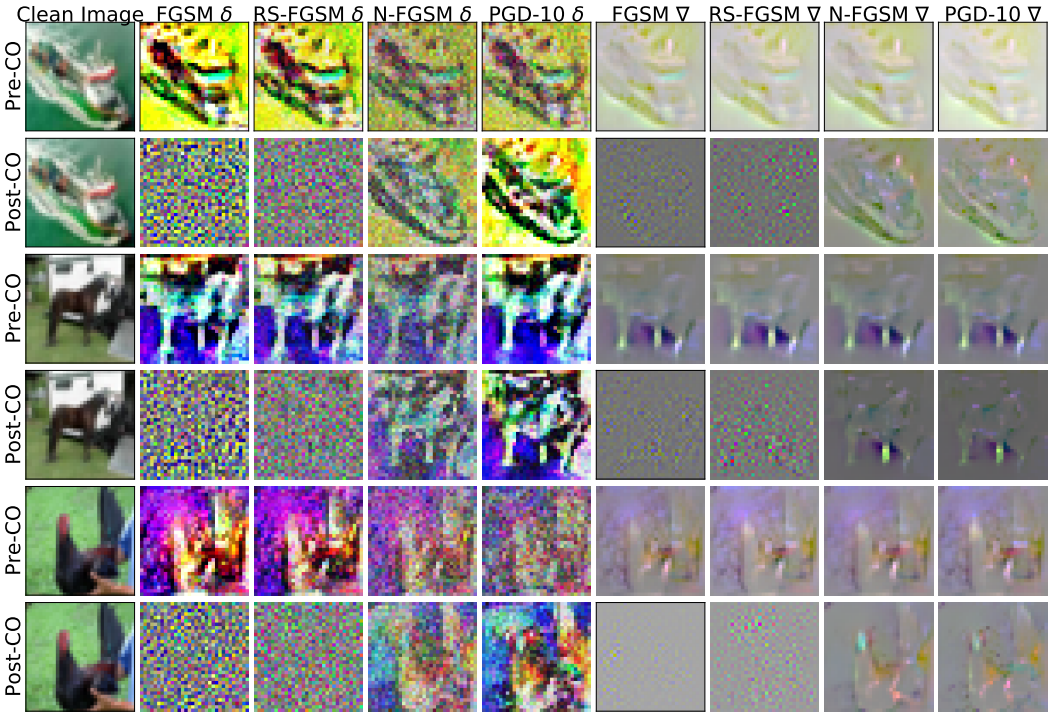


Figure 12: Visualization of adversarial perturbations (δ's) and gradients averaged across several epochs before CO (pre-CO) and after (post-CO). Note that only FGSM and RS-FGSM present CO, PGD-10 and N-FGSM do not. Post-CO, FGSM and RS-FGSM obtain δ's that are hard to interpret, idem for their gradients.

G Robust evaluations with autoattack

Table 4: Clean (top) and robust accuracy (bottom) for CIFAR-10 and PreacResNet18 evaluated with autoattack (AA) [6]. We find the same trend as with PGD50-10.

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
FGSM	91.52 $\pm$ 0.08	88.59 $\pm$ 0.08	85.17 $\pm$ 0.03	86.62 $\pm$ 0.08	83.35 $\pm$ 2.03	78.51 $\pm$ 3.3	77.31 $\pm$ 1.9	75.88 $\pm$ 1.49
	78.99 $\pm$ 0.19	65.99 $\pm$ 0.24	54.0 $\pm$ 0.32	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
GradAlign	91.48 $\pm$ 0.08	88.55 $\pm$ 0.18	85.23 $\pm$ 0.22	81.69 $\pm$ 0.1	77.73 $\pm$ 0.18	73.46 $\pm$ 0.16	67.87 $\pm$ 0.5	61.66 $\pm$ 0.32
	79.09 $\pm$ 0.21	65.65 $\pm$ 0.13	53.99 $\pm$ 0.2	44.11 $\pm$ 0.34	35.72 $\pm$ 0.34	28.66 $\pm$ 0.15	22.93 $\pm$ 0.33	18.4 $\pm$ 0.28
N-FGSM	91.44 $\pm$ 0.09	88.36 $\pm$ 0.04	84.56 $\pm$ 0.12	80.36 $\pm$ 0.03	75.81 $\pm$ 0.22	71.03 $\pm$ 0.16	66.49 $\pm$ 0.36	62.86 $\pm$ 0.88
	78.99 $\pm$ 0.17	66.06 $\pm$ 0.25	53.94 $\pm$ 0.3	44.36 $\pm$ 0.26	36.73 $\pm$ 0.27	30.45 $\pm$ 0.2	25.08 $\pm$ 0.15	19.0 $\pm$ 1.08

Following previous work, [1, 11] we have evaluated robustness with PGD50-10, i.e. PGD with 50 iterations and 10 restarts. However, for the sake of completeness, we also present results of robust accuracy evaluated with autoattack [6]. In Table 4 we evaluate models adversarially trained with our proposed method N-FGSM, the baseline FGSM and GradAlign. We observe the same pattern as with the PGD50-10 attack, therefore we are conviced that our results are general.

H Catastrophic Overfitting outside the ResNet family

Previous work focusing on CO has only used architectures from the ResNet family. In Table 5 we present results for adversarial training with a VGG-16 architecture [28]. Similarly to other studied models we observe that FGSM leads to CO while N-FGSM is able to prevent it. However, it seems that FGSM presents CO for slighly larger ϵ radii, indicating that the architecture might play a role in CO. We consider investigating this further a promising direction of future work.

Table 5: Clean (top) and robust accuracy (bottom) for CIFAR-10 and VGG-16 [28] evaluated with PGD50-10. We also observe CO for VGG architecture when trained with FGSM, moreover, N-FGSM is able to prevent CO. Interestingly, for VGG CO happens for slighly large ϵ values indicating that the architecture might play a role in CO.

	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
FGSM	85.04 $\pm$ 0.1	79.34 $\pm$ 0.11	73.39 $\pm$ 0.0	82.6 $\pm$ 0.0	83.04 $\pm$ 0.0	81.4 $\pm$ 0.0	80.41 $\pm$ 0.21
	62.94 $\pm$ 0.07	52.72 $\pm$ 0.12	44.0 $\pm$ 0.02	0.07 $\pm$ 0.0	0.8 $\pm$ 0.0	0.25 $\pm$ 0.0	0.31 $\pm$ 0.15
N-FGSM	84.53 $\pm$ 0.0	79.42 $\pm$ 0.0	72.01 $\pm$ 0.28	66.81 $\pm$ 0.54	61.19 $\pm$ 0.0	56.97 $\pm$ 0.0	53.1 $\pm$ 1.19
	63.32 $\pm$ 0.0	53.0 $\pm$ 0.0	44.3 $\pm$ 0.09	38.25 $\pm$ 0.1	33.36 $\pm$ 0.0	29.23 $\pm$ 0.0	25.72 $\pm$ 0.22

I Further increasing the attack radii

Following previous work [1] we have studied ϵ attack radii up to $epsilon = 16/255$. Indeed, the performance at these radius is already significantly degraded and thus it would not be very practical for most applications. However, to show that N-FGSM can prevent CO at even larger radii we test two additional radii, $\epsilon = 20/255$ and $\epsilon = 24/255$. In both cases N-FGSM is able to prevent CO. For $\epsilon = 20/255$ we obtain a clean accuracy of 51.63 $\pm$ 0.38 and robust of 20.62 $\pm$ 0.37 while for $\epsilon = 24/255$ we obtain a clean accuracy of 40.16 $\pm$ 0.96 and robust of 15.3 $\pm$ 1.49. We argue that it is of little interest to try even larger perturbations unless more effective methods to improve both the clean and robust performance are found.

J Testing other norms

Following previous work, we have focused on the ℓ_∞ threat model. Although this is where works studying CO have mainly focused, we observe that CO is also present in other norms such as ℓ_1 and ℓ_2. Moreover, in both cases we observe that N-FGSM is able to prevent CO. Interestingly, the range of norms in which we observe CO is usually much higher than normally tested for these norms which would explain why the ℓ_∞ norm has been the main focus of study in related works.

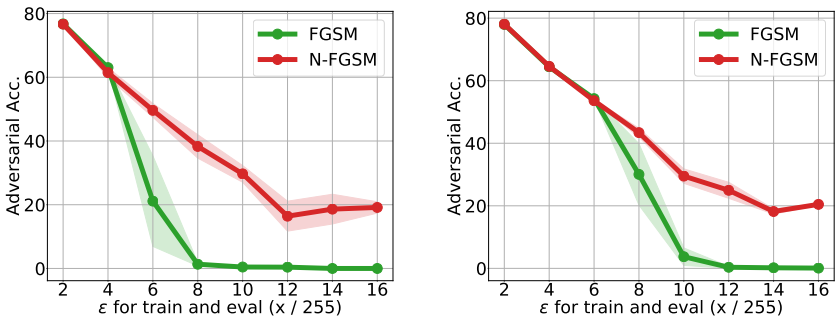


Figure 13: Robust accuracy after training with FGSM or N-FGSM using ℓ_1 (left) and ℓ_2 (right) perturbations. As observed for ℓ_∞ perturbations FGSM leads to CO, while N-FGSM is able to prevent it. Note that the strength of the perturbations is indicated to be equivalent to ℓ_∞ perturbations where all pixels have maximum magnitude i.e. $\epsilon = 8/255$ indicates perturbations were restricted to an ℓ_p norm of a vector where all components are in $\{-\epsilon, +\epsilon\}$. Which would correspond to an ℓ_1 norm of $n\epsilon$ and an ℓ_2 norm of $\epsilon\sqrt{n}$ where n indicates the dimensionality of the input.

K Combining N-FGSM with additional regularizers

In this section we present the results from Section 7 where we combine N-FGSM with additional regularizers [29, 30] that were proposed for single-step adversarial training to boost the performance. First, we try the proposed methods with the default settings (which use a version of RS-FGSM with Bernoulli noise) and observe they lead to CO for larger ϵ. Then we compare them with N-FGSM + Regularizer where we apply their proposed regularizers to N-FGSM. If we apply GAT or NuAT regularizers to N-FGSM then we do not observe CO and usually a boost in performance. Results are presented in Table 6.

Table 6: Clean accuracy (top) and PGD50-10 accuracy (bottom) of N-FGSM with additional regularizers introduced in GAT [29] and NuAT [30]. Both GAT and NuAT present CO with their default training method. If we apply their proposed regularizers to N-FGSM we can avoid CO while achieving a boost in performance.

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
GAT	88.79 $\pm$ 0.15	84.35 $\pm$ 0.11	80.16 $\pm$ 0.15	76.75 $\pm$ 0.38	73.71 $\pm$ 0.12	80.44 $\pm$ 5.08	83.9 $\pm$ 1.0	82.17 $\pm$ 2.47
	80.04 $\pm$ 0.06	68.51 $\pm$ 0.08	59.16 $\pm$ 0.24	50.98 $\pm$ 0.12	43.34 $\pm$ 0.23	14.93 $\pm$ 9.26	2.33 $\pm$ 0.58	1.25 $\pm$ 0.51
N-FGSM+GAT	89.1 $\pm$ 0.08	84.84 $\pm$ 0.05	81.38 $\pm$ 0.07	78.28 $\pm$ 0.04	75.66 $\pm$ 0.35	73.56 $\pm$ 0.23	70.84 $\pm$ 0.51	65.48 $\pm$ 0.96
	79.96 $\pm$ 0.21	69.5 $\pm$ 0.18	60.06 $\pm$ 0.09	51.8 $\pm$ 0.34	44.97 $\pm$ 0.07	38.71 $\pm$ 0.16	32.71 $\pm$ 0.11	27.87 $\pm$ 0.35
NuAT	87.81 $\pm$ 0.24	82.9 $\pm$ 0.18	78.06 $\pm$ 0.2	73.22 $\pm$ 0.34	71.08 $\pm$ 4.87	74.38 $\pm$ 7.32	78.5 $\pm$ 1.54	80.1 $\pm$ 1.08
	79.49 $\pm$ 0.03	67.77 $\pm$ 0.13	57.93 $\pm$ 0.17	50.1 $\pm$ 0.33	34.35 $\pm$ 9.0	17.54 $\pm$ 8.82	6.6 $\pm$ 0.77	3.29 $\pm$ 0.87
NGFSM+NuAT	87.92 $\pm$ 0.0	83.54 $\pm$ 0.0	78.86 $\pm$ 0.25	74.61 $\pm$ 0.34	70.37 $\pm$ 0.12	65.56 $\pm$ 0.19	60.76 $\pm$ 0.74	52.79 $\pm$ 0.66
	79.52 $\pm$ 0.0	68.36 $\pm$ 0.0	58.88 $\pm$ 0.16	51.12 $\pm$ 0.2	44.62 $\pm$ 0.38	38.24 $\pm$ 0.38	32.85 $\pm$ 0.58	29.19 $\pm$ 0.35
N-FGSM	91.48 $\pm$ 0.17	88.44 $\pm$ 0.09	84.72 $\pm$ 0.04	80.58 $\pm$ 0.22	75.98 $\pm$ 0.1	71.46 $\pm$ 0.14	67.11 $\pm$ 0.37	63.18 $\pm$ 0.49
	79.43 $\pm$ 0.21	67.09 $\pm$ 0.31	56.62 $\pm$ 0.26	48.12 $\pm$ 0.07	41.56 $\pm$ 0.16	36.43 $\pm$ 0.16	32.11 $\pm$ 0.2	27.67 $\pm$ 0.93

L Imagenet experimental details

For our experiments on Imagenet we mainly follow the settings from [35]. However, for simplicity we did not do image resizing which requires storing two additional Imagenet datasets. More importantly, we found that the learning rate schedule suggested in [35] was not optimal for N-FGSM. The schedule suggested in [35] follows three different stages in which the learning increases or decreases linearly for some iterations. In particular in the first stage, the learning rate has an initial warm-up where it increases linearly from 0.0 to 0.4 during the first epoch and then decreases linearly to 0.04 during the next 5 epochs. As a lucky coincidence when debugging, we modified this initial stage such that we preserved the initial increase to 0.4 for the first epoch, but then we directly jumped to a learning rate of 0.04 which remained constant for the next 5 epochs. For phase 2 and 3 both schedules remained the same. First decreasing from 0.04 to 0.004 for epoch 6 to 12 and finally from 0.004 to 0.0004 for epoch 12 to 15. This small change made N-FGSM improve both in clean and robust accuracy for $\epsilon = 4/255,\ 6/255$ and these are the numbers reported. This indicates that further tuning the learning rate schedule might be an effective way to improve performance and even help prevent CO, however, due to the computational demands of ImageNet adversarial training we leave it for future work. To be thorough we also trained RS-FGSM and FGSM with the modified schedule and found that neither of them benefit from it. Regarding N-FGSM hyperparameters, for $\epsilon = 2/255$ we used $\alpha = 2/255$ and $k = 1$; for $\epsilon = 4/255$ we used $\alpha = 4/255$ and $k = 1$; and for $\epsilon = 6/255$ we also used $\alpha = 4/255$ and $k = 1$.

M Visualization of the loss surface

In this section we present a visualization of the loss surface. We adapted the code from [15] to analyse the shape of the loss surface at the end of training for different methods. [15] reported that after adversarial training CO, the loss surface would become non-linear. In particular, they found that the FGSM perturbation seems to be misguided by local maxima very close to the clean image that result in ineffective attacks. We note this was already reported by [32] which proposed to perform a

random step to *escape* those maxima. We argue that adding noise to the random step, when properly implemented, actually prevents those maxima to appear in the first place.

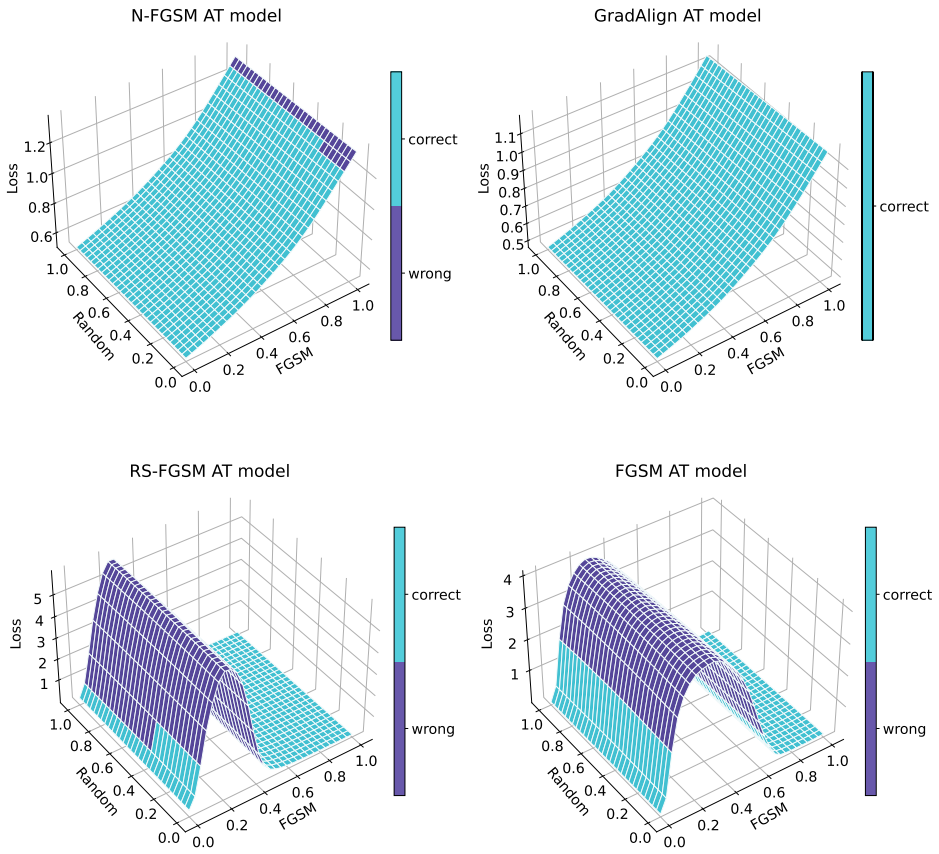


Figure 14: Visualization of the loss surface for models trained using different methods. Given a clean sample from the test set in coordinate $(0, 0)$, we compute the FGSM perturbation and evaluate the loss on the subspace generated by the FGSM perturbation direction and a random direction. That is, we evaluate $x_{\text{clean}} + t_1 \cdot \delta_{\text{FGSM}} + t_2 \cdot \delta_{\text{random}}$, where $t_1,\ t_2 \in [0, 1]$. Note that FGSM and RS-FGSM both have CO and the final models present a highly non-linear loss surface, on the other hand, both N-FGSM and GradAlign produce final models with a very linear loss surface which is key to obtain meaningful perturbations.

N Magnitude of N-FGSM perturbations

Lemma N.1 (Expected perturbation)**.** *Consider the N-FGSM perturbation as defined in Equation (3)*

$$\delta_{\textit{N-FGSM}} = \eta + \alpha \cdot \textit{sign}\left(\nabla_x \ell(f(x+\eta), y)\right), \quad \textit{where} \;\; \eta \sim \Omega.$$

Let the distribution Ω be the uniform distribution $\mathcal{U}\left([-k\epsilon, k\epsilon]^d\right)$ and $\alpha > 0$. Then,

$$\mathbb{E}_\eta\left[\|\delta_{\textit{N-FGSM}}\|_2^2\right] = d\left(\frac{k^2\epsilon^2}{3} + \alpha^2\right) \quad \textit{and} \quad \mathbb{E}_\eta\left[\|\delta_{\textit{N-FGSM}}\|_2\right] \leq \sqrt{d\left(\frac{k^2\epsilon^2}{3} + \alpha^2\right)}$$

Proof. By Jensen's inequality, we have

$$\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2\right] \leq \sqrt{\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2^2\right]}$$

Then let us consider the term $\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2^2\right]$ and use the shorthand $\nabla(\eta)_i = \left(\nabla_x \ell(f(x+\eta), y)\right)_i$.

$$\begin{aligned}
\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2^2\right] =& \mathbb{E}_\eta \|\eta + \alpha \cdot \text{sign}\left(\nabla_x \ell(f(x+\eta), y)\right)\|_2^2 \\
=& \mathbb{E}_\eta\left[\sum_{i=1}^d \left(\eta_i + \alpha \cdot \text{sign}(\nabla(\eta)_i)\right)^2\right] \\
=& \sum_{i=1}^d \mathbb{E}_\eta\left[\left(\eta_i + \alpha \cdot \text{sign}(\nabla(\eta)_i)\right)^2\right] \\
=& \sum_{i=1}^d \mathbb{E}_\eta\left[\left(\eta_i + \alpha \cdot \text{sign}(\nabla(\eta)_i)\right)^2 | \text{sign}(\nabla(\eta)_i) = 1\right] \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = 1\right] \\
&+ \sum_{i=1}^d \mathbb{E}_\eta\left[\left(\eta_i + \alpha \cdot \text{sign}(\nabla(\eta)_i)\right)^2 | \text{sign}(\nabla(\eta)_i) = -1\right] \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = -1\right] \\
=& \sum_{i=1}^d \frac{1}{2k\epsilon} \int_{-k\epsilon}^{k\epsilon} \left(\eta_i + \alpha\right)^2 d\eta_i \cdot \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = 1\right] \\
&+ \frac{1}{2k\epsilon} \sum_{i=1}^d \int_{-k\epsilon}^{k\epsilon} \left(\eta_i - \alpha\right)^2 d\eta_i \cdot \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = -1\right] \\
=& \sum_{i=1}^d \frac{1}{2k\epsilon} \int_{\alpha-k\epsilon}^{\alpha+k\epsilon} z^2 dz \cdot \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = 1\right] \\
&+ \frac{1}{2k\epsilon} \sum_{i=1}^d \int_{-\alpha-k\epsilon}^{-\alpha+k\epsilon} z^2 dz \cdot \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = -1\right] \\
=& \sum_{i=1}^d \frac{1}{2k\epsilon} \int_{\alpha-k\epsilon}^{\alpha+k\epsilon} z^2 dz \cdot \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = 1\right] \\
&+ \frac{1}{2k\epsilon} \sum_{i=1}^d \int_{\alpha-k\epsilon}^{\alpha+k\epsilon} z^2 dz \cdot \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = -1\right] \\
=& \frac{1}{2k\epsilon} \int_{\alpha-k\epsilon}^{\alpha+k\epsilon} z^2 dz \sum_{i=1}^d \left(\mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = 1\right] + \mathbb{P}_\eta\left[\text{sign}(\nabla(\eta)_i) = -1\right]\right) \\
=& \frac{d}{6k\epsilon}\left[(\alpha + k\epsilon)^3 - (\alpha - k\epsilon)^3\right] = \frac{dk^2\epsilon^2}{3} + d\alpha^2
\end{aligned}$$

Therefore,

$$\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2\right] \leq \sqrt{d\left(\frac{k^2\epsilon^2}{3}+\alpha^2\right)}.$$

□

Theorem N.2. *Let $\delta_{N\text{-}FGSM}$ be our proposed single-step method defined by Equation (3), δ_{FGSM} be the FGSM method [11] and $\delta_{RS\text{-}FGSM}$ be the RS-FGSM method [35]. Then, with default hyperparameter values and for any $\epsilon > 0$, we have that*

$$\mathbb{E}_\eta\left[\|\delta_{N\text{-}FGSM}\|_2^2\right] > \mathbb{E}_\eta\left[\|\delta_{FGSM}\|_2^2\right] > \mathbb{E}_\eta\left[\|\delta_{RS\text{-}FGSM}\|_2^2\right].$$

Proof. From Lemma N.1 we have that

$$\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2^2\right] = d\left(\frac{k^2\epsilon^2}{3}+\alpha^2\right).$$

On the other hand, [1] showed that

$$\mathbb{E}_\eta\left[\|\delta_{\text{RS-FGSM}}\|_2^2\right] = d\left(-\frac{1}{6\epsilon}\alpha^3+\frac{1}{2}\alpha^2+\frac{1}{3}\epsilon^2\right).$$

Finally, we note that

$$\mathbb{E}_\eta\left[\|\delta_{\text{FGSM}}\|_2^2\right] = \|\delta_{\text{FGSM}}\|_2^2 = d\epsilon^2.$$

The default hyperparameters for N-FGSM are $k=2,\ \ \alpha=\epsilon$ and RS-FGSM uses $\alpha = 5\epsilon/4$. With these hyperparameters and any $\epsilon > 0$ we have

$$\mathbb{E}_\eta\left[\|\delta_{\text{N-FGSM}}\|_2^2\right] = \frac{7}{3}d\epsilon^2 > \mathbb{E}_\eta\left[\|\delta_{\text{FGSM}}\|_2^2\right] = d\epsilon^2 > \mathbb{E}_\eta\left[\|\delta_{\text{RS-FGSM}}\|_2^2\right] = \frac{101}{128}d\epsilon^2$$

□

In Lemma N.1 we compute the expected value of the squared ℓ_2 norm of N-FGSM perturbations and by Jensen's inequality we obtain an upper bound for the expected ℓ_2 norm of N-FGSM perturbations. However, obtaining the exact expected magnitude is more complex. To compliment our analytic results, we approximate the ℓ_2 norm of FGSM, RS-FGSM and N-FGSM via Monte Carlo sampling. Results are presented in Figure 15. We observe that the empirical estimations are very close to the analytical upper bounds and that indeed, N-FGSM has a magnitude significantly above that of FGSM or RS-FGSM.

O N-FGSM with Gaussian noise

In the main paper we have only explored noise sources coming from a Uniform distribution. Since we are measuring robustness against $l_\infty-$ attacks, the Uniform distribution is a natural choice because the random perturbations will be bounded to the l_∞ ball defined by the span of the distribution. However, for the sake of completeness, we also explore the performance of augmenting the samples from a Gaussian distribution where we choose its standard deviation to match that of the uniform distribution. In Table 7 we present a comparison of the clean (top) and PGD-50-10 (bottom) accuracy for different values of α and noise magnitude with $\epsilon = {}^{8}/_{255}$. Recall that by default we use Uniform distribution $\mathcal{U}[-k, k]$, therefore hyperparameter k sets the noise magnitude.

Increasing the FGSM step size without increasing the amount of noise leads to CO. Note results for $k = 0.5\epsilon$. More importantly, results are very similar when the two noise distributions share the same standard deviation. Thus, using Gaussian instead of Uniform noise does not seem to alter the results. Although this might be expected, we remark that the Gaussian is an unbounded noise distribution and the common practice in adversarial training is to always restrict the norm of the perturbations.

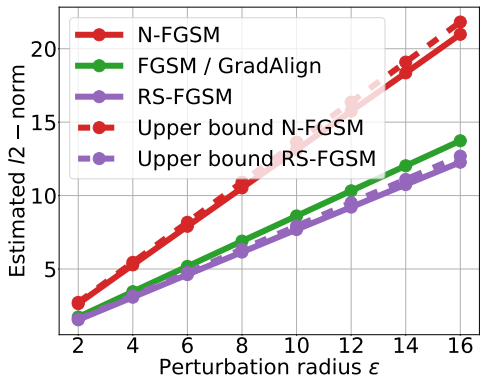


Figure 15: Monte Carlo estimations of the expected l_2-norm of perturbations from different methods and corresponding analytical upper bounds. As mentioned in [1], we observe that RS-FGSM perturbations have lower l_2 norm than FGSM. However, N-FGSM perturbations have a significantly higher l_2-norm than both RS-FGSM and FGSM. This seems to indicate that the role of random step is not simply to lower the l_2 norm as previously suggested [1].

Table 7: Comparison of the clean (top) and PGD-50-10 (bottom) accuracy across different values of step-size α and noise magnitude for the Uniform and Gaussian distributions with $\epsilon = 8/255$. For every value of k, we use a Gaussian with matching standard deviation. We observe that when we match the standard deviation, both distribution perform similarly.

	Uniform Noise			**Gaussian Noise**		
	$\alpha = 6/255\ (0.75\epsilon)$	$\alpha = 8/255\ (1\epsilon)$	$\alpha = 10/255\ (1.25\epsilon)$	$\alpha = 6/255\ (0.75\epsilon)$	$\alpha = 8/255\ (1\epsilon)$	$\alpha = 10/255\ (1.25\epsilon)$
$k = 0.5\epsilon$	85.52 $\pm$ 0.23	81.54 $\pm$ 0.19	82.81 $\pm$ 1.11	85.27 $\pm$ 0.11	81.71 $\pm$ 0.27	83.34 $\pm$ 1.48
	44.14 $\pm$ 0.24	47.93 $\pm$ 0.11	0.0 $\pm$ 0.0	44.23 $\pm$ 0.17	47.98 $\pm$ 0.14	0.0 $\pm$ 0.0
$k = 1\epsilon$	85.03 $\pm$ 0.09	81.57 $\pm$ 0.07	77.32 $\pm$ 0.14	85.01 $\pm$ 0.17	81.35 $\pm$ 0.14	77.22 $\pm$ 0.32
	44.44 $\pm$ 0.13	48.16 $\pm$ 0.21	49.68 $\pm$ 0.25	44.41 $\pm$ 0.04	48.21 $\pm$ 0.11	49.83 $\pm$ 0.1
$k = 2\epsilon$	84.49 $\pm$ 0.1	80.58 $\pm$ 0.22	76.49 $\pm$ 0.14	84.35 $\pm$ 0.24	80.44 $\pm$ 0.31	76.33 $\pm$ 0.37
	44.44 $\pm$ 0.15	48.12 $\pm$ 0.07	49.77 $\pm$ 0.37	44.59 $\pm$ 0.22	48.34 $\pm$ 0.1	49.77 $\pm$ 0.23

P Training with noise augmented samples

Gilmer et al. [9] and Fawzi et al. [8] report a close link between robustness to adversarial attacks and robustness to random noise. Actually, [9] report that training with noise-augmented samples can improve adversarial accuracy and vice-versa. We note that N-FGSM can actually be seen as a combination of noise-augmentation and adversarial attacks. Here we perform an ablation where we train models with samples augmented with uniform noise $\mathcal{U}[-k, k]$ and then test the PGD-50-10 accuracy. We observe, that indeed random noise can increase the robustness to worst-case perturbations for small $\epsilon - l_\infty$ balls. However, as we increase ϵ, noise augmentation is no longer very effective. With N-FGSM, we apply a weak attack to these noise-augmented samples and this seems to be enough to make them effective for adversarial training.

Q Comparison of adversarial training cost

In this section we describe how we compute the relative training cost for single-step methods shown in Figure 1 (right). We approximate the cost based on the number of forward/backward passes each method uses, disregarding the cost of other additional operations such as adding a random step for RS-FGSM or N-FGSM. We understand these operations have a negligible cost compared to a full forward or backward pass.

FGSM: FGSM is the cheapest of all methods since it only uses one forward/backward to compute the attack and an additional forward/backward to compute the weight update. Hence, Cost FGSM = 2 F/B.

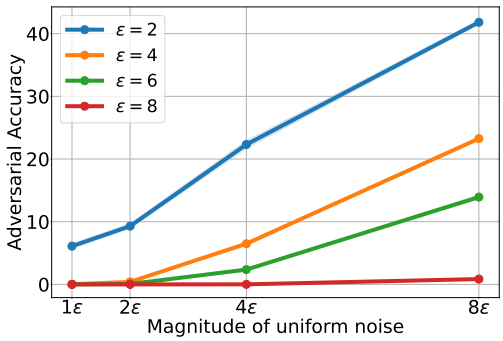


Figure 16: Training with uniform noise augmented samples improves adversarial accuracy for small perturbations but is not effective to protect against larger l_∞ radius ϵ. This motivates us to further augment the noisy samples with FGSM. All experiments are averaged over 3 runs.

RS-FGSM: As previously mentioned, we do not take into account the cost of random steps or clipping, hence we consider RS-FGSM to have the same cost as standard FGSM. Cost RS-FGSM = 2 F/B.

N-FGSM: Idem as before, cost of N-FGSM = 2 F/B.

ZeroGrad: For ZeroGrad they need to do an additional sorting operation to find the smallest gradient components. This could be potentially expensive, however, since the size of the input image is several orders of magnitude smaller than that of the network, we also ignore this cost. Cost ZeroGrad = 2 F/B.

MultiGrad: MultiGrad computes 3 random steps and evaluates the gradient in all of them. Therefore, it needs to do 3 F/B to compute the attack and an additional one to update the weights. Cost MultiGrad = 4 F/B.

[15]: [15] compute the RS-FGSM perturbation and evaluate the model on c points along this direction. Therefore, they will spend 1F/B on the RS-FGSM attack, $c - 1$ F on the evaluations since the clean image has already been evaluated; and 1 F/B for the weight update. In our plot, we used $c = 3$ since it was the most chosen setting. [15] assume the cost of a forward is similar to that of a backward pass, following this assumption, cost of [15] is 1 F/B + 2 F + 1 F/B = 3 F/B

Free-AT: [26] re-use the gradient from the previous backward pass to compute the FGSM perturbation of the current iteration. Hence, the cost of their training is only 1 F/B per iteration. However, [35] observed they needed a longer training schedule to produce comparable results. Therefore, the total training cost per iteration (1 F/B) is scaled by 96 in the case of Free-AT, while it is only scaled by 30 for other methods. Relative cost Free = (96 · 1 F/B) / (30 · 2 F/B).

GradAlign: Finally, GradAlign uses FGSM with a regularizer. However, this regularizer needs to compute second-order derivatives via double backpropagation, which does not have the same cost as regular backpropagation. [1] report that the cost of using GradAlign regularizer increased the cost of FGSM by 3.

R Infrastructure details and GPU hours

All our training runs have been conducted on either NVIDIA GPU V-100 or P-100 from an internal cluster. The total compute for the results presented in this work is roughly 2500 hours.

S Detailed results for Section 5.1 and Section 7

In this section we present the tables with the exact numbers used in plots comparing adversarial training methods. For each method and $\epsilon - l_\infty$ radius, the top number is the clean accuracy while the bottom number is the PGD-50-10 accuracy. We separate single-step from multi-step methods with a double line.

PreActResNet18 – CIFAR-10 Dataset

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
N-FGSM	91.48 ± 0.17	88.44 ± 0.09	84.72 ± 0.04	80.58 ± 0.22	75.98 ± 0.1	71.46 ± 0.14	67.11 ± 0.37	63.18 ± 0.49
	79.43 ± 0.21	**67.09 ± 0.31**	**56.62 ± 0.26**	**48.12 ± 0.07**	**41.56 ± 0.16**	**36.43 ± 0.16**	**32.11 ± 0.2**	**27.67 ± 0.93**
Grad Align	91.73 ± 0.04	88.76 ± 0.0	85.67 ± 0.02	81.9 ± 0.22	77.54 ± 0.06	73.29 ± 0.23	68.01 ± 0.32	61.3 ± 0.15
	79.16 ± 0.03	**67.13 ± 0.26**	**56.27 ± 0.31**	**48.14 ± 0.15**	40.75 ± 0.28	34.51 ± 0.63	30.36 ± 0.27	**26.64 ± 0.27**
FGSM	91.6 ± 0.1	88.77 ± 0.04	85.58 ± 0.11	86.41 ± 0.7	82.08 ± 1.62	80.6 ± 2.59	76.04 ± 2.37	77.14 ± 2.46
	79.35 ± 0.06	67.11 ± 0.09	56.33 ± 0.41	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS-FGSM	92.09 ± 0.05	89.69 ± 0.01	87.0 ± 0.12	84.05 ± 0.13	85.21 ± 0.51	65.22 ± 23.23	43.59 ± 25.01	76.66 ± 0.38
	78.64 ± 0.08	66.12 ± 0.22	54.87 ± 0.22	46.08 ± 0.18	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Kim et. al.	92.85 ± 0.11	91.1 ± 0.04	89.34 ± 0.05	89.02 ± 0.1	88.27 ± 0.14	88.35 ± 0.31	90.01 ± 0.25	90.45 ± 0.08
	74.74 ± 0.35	60.51 ± 0.4	48.95 ± 0.45	33.01 ± 0.09	24.43 ± 0.84	13.11 ± 0.63	5.86 ± 0.57	1.88 ± 0.05
AT Free	87.99 ± 0.16	84.98 ± 0.13	81.77 ± 0.11	78.41 ± 0.18	74.79 ± 0.22	73.91 ± 4.19	61.92 ± 14.94	71.64 ± 3.89
	74.27 ± 0.33	62.47 ± 0.25	53.18 ± 0.15	46.03 ± 0.36	39.87 ± 0.07	22.99 ± 16.26	0.0 ± 0.0	0.0 ± 0.0
ZeroGrad	91.71 ± 0.08	88.8 ± 0.11	85.71 ± 0.1	82.62 ± 0.05	79.91 ± 0.12	78.11 ± 0.2	75.66 ± 0.46	75.42 ± 0.13
	79.36 ± 0.05	**67.32 ± 0.02**	56.14 ± 0.21	47.08 ± 0.1	37.58 ± 0.2	27.41 ± 0.27	21.29 ± 0.97	13.06 ± 0.22
MultiGrad	91.57 ± 0.16	88.74 ± 0.12	85.75 ± 0.05	82.33 ± 0.14	78.73 ± 0.16	75.28 ± 0.2	80.94 ± 5.94	71.42 ± 5.63
	79.34 ± 0.02	66.81 ± 0.02	56.02 ± 0.3	47.29 ± 0.07	40.11 ± 0.24	33.87 ± 0.17	9.55 ± 13.5	16.35 ± 11.57
PGD-2	91.4 ± 0.07	88.46 ± 0.13	85.14 ± 0.13	81.41 ± 0.05	77.18 ± 0.15	72.9 ± 0.26	70.39 ± 2.71	64.81 ± 11.58
	79.55 ± 0.15	67.62 ± 0.03	57.39 ± 0.13	49.58 ± 0.08	43.3 ± 0.11	38.13 ± 0.15	22.89 ± 15.26	9.6 ± 13.37
PGD-10	91.25 ± 0.04	88.34 ± 0.11	84.79 ± 0.11	80.71 ± 0.14	76.13 ± 0.35	71.24 ± 0.3	66.7 ± 0.39	62.11 ± 0.62
	79.47 ± 0.13	**68.29 ± 0.24**	**58.85 ± 0.18**	**51.33 ± 0.31**	**45.02 ± 0.49**	**39.93 ± 0.5**	**36.02 ± 0.67**	**32.22 ± 0.64**

PreActResNet18 – CIFAR-100 Dataset

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
N-FGSM	69.12 ± 0.27	64.0 ± 0.06	59.53 ± 0.02	54.9 ± 0.2	50.6 ± 0.16	46.06 ± 0.14	41.67 ± 0.25	37.91 ± 0.11
	51.02 ± 0.34	**39.5 ± 0.12**	**32.06 ± 0.37**	**26.46 ± 0.22**	**22.23 ± 0.17**	**18.95 ± 0.15**	**16.33 ± 0.15**	**14.34 ± 0.07**
Grad Align	68.96 ± 0.15	64.71 ± 0.16	60.42 ± 0.23	56.53 ± 0.31	54.06 ± 0.44	48.87 ± 0.32	43.84 ± 0.14	38.93 ± 0.21
	51.31 ± 0.12	**39.37 ± 0.25**	**31.91 ± 0.28**	25.8 ± 0.14	18.7 ± 1.92	17.86 ± 0.04	15.51 ± 0.16	13.62 ± 0.19
FGSM	69.01 ± 0.13	64.47 ± 0.15	63.85 ± 2.18	53.42 ± 0.65	45.06 ± 2.29	46.14 ± 2.58	41.66 ± 0.88	44.68 ± 1.74
	51.3 ± 0.19	39.7 ± 0.16	10.93 ± 14.64	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS-FGSM	69.83 ± 0.29	65.9 ± 0.36	62.15 ± 0.23	55.26 ± 6.86	32.33 ± 12.12	36.07 ± 2.59	21.52 ± 5.56	20.38 ± 6.15
	50.13 ± 0.32	38.36 ± 0.19	30.82 ± 0.08	0.01 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Kim et. al.	72.92 ± 0.41	70.16 ± 0.07	67.98 ± 0.19	68.07 ± 0.1	68.37 ± 0.21	74.09 ± 0.06	74.06 ± 0.34	74.01 ± 0.36
	44.19 ± 0.25	30.63 ± 0.28	22.0 ± 0.02	12.75 ± 0.21	6.98 ± 0.23	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
AT Free	63.01 ± 0.19	59.41 ± 0.27	55.43 ± 0.37	51.91 ± 0.08	48.11 ± 0.09	43.48 ± 1.25	18.33 ± 4.86	20.43 ± 11.25
	45.7 ± 0.33	35.95 ± 0.09	29.37 ± 0.21	24.32 ± 0.4	20.64 ± 0.22	5.71 ± 8.05	0.0 ± 0.0	0.0 ± 0.0
ZeroGrad	69.35 ± 0.36	64.59 ± 0.32	60.69 ± 0.09	56.94 ± 0.13	54.55 ± 0.17	52.97 ± 0.34	50.87 ± 0.26	50.73 ± 0.3
	51.1 ± 0.09	**39.38 ± 0.15**	31.72 ± 0.21	25.87 ± 0.09	19.49 ± 0.08	14.32 ± 0.08	10.92 ± 0.59	7.3 ± 0.16
MultiGrad	69.01 ± 0.16	64.44 ± 0.11	60.65 ± 0.26	56.84 ± 0.2	53.62 ± 0.25	53.05 ± 1.85	48.28 ± 0.66	45.28 ± 11.14
	51.15 ± 0.03	39.16 ± 0.03	31.73 ± 0.09	25.96 ± 0.11	21.37 ± 0.16	9.57 ± 7.32	3.2 ± 4.49	0.0 ± 0.0
PGD-2	69.18 ± 0.1	64.32 ± 0.14	60.21 ± 0.13	55.8 ± 0.16	51.68 ± 0.1	48.2 ± 0.1	46.14 ± 1.24	37.97 ± 10.52
	51.36 ± 0.03	40.06 ± 0.14	32.99 ± 0.24	27.38 ± 0.16	23.39 ± 0.19	19.83 ± 0.29	10.55 ± 7.51	4.79 ± 6.75
PGD-10	68.83 ± 0.07	63.87 ± 0.09	59.37 ± 0.07	54.79 ± 0.38	50.53 ± 0.15	46.05 ± 0.21	41.76 ± 0.07	37.81 ± 0.14
	51.51 ± 0.27	**40.59 ± 0.36**	**33.65 ± 0.02**	**28.55 ± 0.27**	**24.17 ± 0.12**	**21.2 ± 0.12**	**18.72 ± 0.06**	**16.59 ± 0.16**

PreActResNet18 – SVHN Dataset

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$
N-FGSM	96.01 ± 0.04	94.54 ± 0.15	92.25 ± 0.33	89.56 ± 0.49	86.74 ± 0.86	81.48 ± 1.64
	86.44 ± 0.1	**72.53 ± 0.19**	58.42 ± 0.14	**45.63 ± 0.11**	**33.96 ± 0.49**	**26.13 ± 0.81**
Grad Align	96.02 ± 0.05	94.56 ± 0.21	92.53 ± 0.24	90.1 ± 0.34	87.23 ± 0.75	84.01 ± 0.46
	86.43 ± 0.1	72.12 ± 0.19	57.34 ± 0.24	43.85 ± 0.14	32.87 ± 0.33	23.62 ± 0.41
FGSM	96.04 ± 0.07	95.67 ± 0.07	93.73 ± 0.68	91.74 ± 0.86	90.76 ± 0.63	87.17 ± 0.43
	86.5 ± 0.05	13.61 ± 5.83	0.56 ± 0.72	0.26 ± 0.36	0.07 ± 0.1	0.0 ± 0.0
RS-FGSM	96.18 ± 0.11	95.09 ± 0.09	95.11 ± 0.44	94.46 ± 0.16	93.88 ± 0.24	92.74 ± 0.5
	86.16 ± 0.14	71.28 ± 0.4	0.11 ± 0.08	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Kim et. al.	96.35 ± 0.02	95.25 ± 0.08	94.83 ± 0.02	94.88 ± 0.29	96.61 ± 0.09	96.61 ± 0.01
	83.26 ± 0.24	66.32 ± 0.63	48.27 ± 0.52	31.8 ± 1.1	0.18 ± 0.21	0.0 ± 0.0
AT Free	95.01 ± 0.09	93.66 ± 0.12	91.72 ± 0.29	91.29 ± 4.07	91.86 ± 3.66	92.36 ± 1.0
	84.55 ± 0.27	71.61 ± 0.75	**59.31 ± 1.0**	0.01 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ZeroGrad	96.06 ± 0.03	94.81 ± 0.16	93.53 ± 0.26	92.42 ± 1.29	90.34 ± 0.32	88.09 ± 0.4
	86.43 ± 0.1	71.59 ± 0.22	51.72 ± 0.53	35.93 ± 2.73	21.34 ± 0.31	14.14 ± 0.32
MultiGrad	96.01 ± 0.08	94.71 ± 0.17	95.75 ± 0.58	94.86 ± 0.97	94.7 ± 0.12	94.48 ± 0.19
	86.4 ± 0.08	**71.98 ± 0.26**	28.1 ± 18.85	11.49 ± 16.19	0.0 ± 0.0	0.0 ± 0.0
PGD-2	96.03 ± 0.14	94.66 ± 0.1	93.77 ± 0.61	94.63 ± 1.29	84.09 ± 14.99	94.16 ± 0.54
	86.72 ± 0.06	73.29 ± 0.29	60.53 ± 0.73	20.68 ± 18.56	0.41 ± 0.29	0.02 ± 0.03
PGD-10	95.92 ± 0.08	94.37 ± 0.13	92.46 ± 0.25	89.67 ± 0.34	85.75 ± 0.65	80.08 ± 0.93
	86.94 ± 0.14	**74.76 ± 0.19**	**63.9 ± 0.48**	**53.95 ± 0.55**	**44.91 ± 0.45**	**37.65 ± 0.53**

WideResNet28-10 – CIFAR-10 Dataset

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
N-FGSM	92.51 ± 0.11	89.65 ± 0.09	85.8 ± 0.23	81.59 ± 0.32	76.92 ± 0.04	72.13 ± 0.15	67.82 ± 0.43	56.73 ± 0.42
	81.43 ± 0.3	69.11 ± 0.24	58.29 ± 0.14	49.53 ± 0.25	**42.37 ± 0.36**	**36.85 ± 0.2**	**31.66 ± 0.6**	25.01 ± 0.23
Grad Align	92.59 ± 0.05	89.95 ± 0.3	86.98 ± 0.06	83.19 ± 0.26	79.35 ± 0.26	73.79 ± 0.72	66.38 ± 0.53	57.75 ± 0.75
	81.33 ± 0.4	**69.81 ± 0.47**	**59.0 ± 0.13**	**50.0 ± 0.05**	41.48 ± 0.51	35.06 ± 0.74	30.83 ± 0.39	**26.26 ± 0.13**
FGSM	92.65 ± 0.17	90.06 ± 0.18	87.99 ± 1.3	86.46 ± 0.45	82.67 ± 1.78	80.14 ± 1.2	74.54 ± 4.01	71.56 ± 3.78
	81.38 ± 0.22	**69.59 ± 0.25**	38.69 ± 26.54	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS-FGSM	92.85 ± 0.1	90.73 ± 0.2	88.24 ± 0.19	83.64 ± 1.74	82.1 ± 1.45	78.62 ± 0.7	73.25 ± 8.16	68.64 ± 4.3
	80.9 ± 0.13	68.23 ± 0.17	57.21 ± 0.17	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RandAlpha	93.37 ± 0.22	92.17 ± 0.21	90.71 ± 0.14	89.16 ± 0.19	87.44 ± 0.31	85.69 ± 0.28	83.98 ± 0.24	83.23 ± 0.46
	77.67 ± 0.66	63.73 ± 0.31	50.4 ± 0.14	39.37 ± 0.42	30.13 ± 0.9	23.13 ± 0.33	16.0 ± 0.22	8.47 ± 0.66
AT Free	90.66 ± 0.25	88.37 ± 0.15	86.11 ± 0.29	83.5 ± 0.27	80.52 ± 0.32	83.59 ± 1.35	39.58 ± 15.8	42.59 ± 27.96
	77.0 ± 0.27	64.25 ± 0.33	53.76 ± 0.48	44.85 ± 0.39	31.87 ± 5.53	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ZeroGrad	92.62 ± 0.11	90.17 ± 0.05	86.98 ± 0.28	84.25 ± 0.28	81.72 ± 0.29	79.24 ± 0.82	78.14 ± 0.46	75.34 ± 0.12
	81.42 ± 0.28	69.28 ± 0.29	58.4 ± 0.14	48.29 ± 0.16	36.08 ± 0.29	28.24 ± 1.79	18.54 ± 0.31	14.6 ± 0.12
MultiGrad	92.64 ± 0.1	90.18 ± 0.13	87.11 ± 0.36	83.87 ± 0.46	80.89 ± 0.14	82.88 ± 2.85	86.6 ± 1.52	85.46 ± 3.73
	81.19 ± 0.28	69.3 ± 0.2	57.98 ± 0.08	48.74 ± 0.09	41.22 ± 0.57	4.46 ± 6.09	0.0 ± 0.0	0.0 ± 0.0
PGD-2	92.69 ± 0.14	90.18 ± 0.19	86.87 ± 0.18	83.31 ± 0.16	79.61 ± 0.47	75.81 ± 0.24	71.41 ± 1.38	67.2 ± 14.94
	81.54 ± 0.18	69.87 ± 0.26	59.4 ± 0.19	50.88 ± 0.16	43.94 ± 0.24	37.77 ± 0.57	21.06 ± 13.39	0.0 ± 0.0
PGD-10	92.24 ± 0.31	89.65 ± 0.33	86.91 ± 0.51	82.82 ± 0.7	78.63 ± 0.66	74.0 ± 0.67	68.6 ± 0.58	64.17 ± 0.72
	81.18 ± 0.57	**70.34 ± 0.26**	**60.59 ± 0.21**	**52.58 ± 0.2**	**45.92 ± 0.38**	**40.44 ± 0.17**	**35.98 ± 0.56**	**32.5 ± 0.61**

WideResNet28-10 – CIFAR-100 Dataset

	$\epsilon = 2/255$	$\epsilon = 5/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$	$\epsilon = 14/255$	$\epsilon = 16/255$
N-FGSM	71.56 ± 0.13	66.49 ± 0.46	61.38 ± 0.68	56.23 ± 0.59	51.54 ± 0.63	46.43 ± 0.61	42.11 ± 0.32	38.34 ± 0.47
	52.23 ± 0.33	**39.93 ± 0.37**	30.97 ± 0.21	**26.77 ± 0.65**	**23.03 ± 0.54**	**19.3 ± 0.59**	**16.67 ± 0.4**	**14.27 ± 0.33**
Grad Align	71.68 ± 0.33	67.09 ± 0.19	62.86 ± 0.1	58.55 ± 0.41	53.85 ± 0.73	46.94 ± 0.86	42.63 ± 0.5	36.17 ± 0.45
	51.5 ± 0.45	**39.9 ± 0.42**	**32.0 ± 0.22**	**26.9 ± 0.62**	**22.63 ± 0.62**	**19.9 ± 0.65**	**16.93 ± 0.12**	**14.03 ± 0.24**
FGSM	71.92 ± 0.33	67.34 ± 0.36	64.72 ± 1.12	56.87 ± 1.24	52.31 ± 2.11	48.99 ± 1.17	44.27 ± 1.4	42.05 ± 1.03
	52.83 ± 0.37	**39.83 ± 0.31**	0.0 ± 0.0	0.03 ± 0.05	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS-FGSM	72.65 ± 0.28	68.26 ± 0.2	65.58 ± 0.69	54.25 ± 5.85	46.08 ± 4.87	35.84 ± 0.17	24.4 ± 1.25	21.37 ± 5.04
	51.63 ± 0.52	39.57 ± 0.09	26.63 ± 2.8	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RandAlpha	73.9 ± 0.15	71.17 ± 0.12	68.65 ± 0.22	66.42 ± 0.13	64.05 ± 0.5	61.99 ± 0.6	59.74 ± 0.57	58.9 ± 0.78
	49.13 ± 0.91	34.3 ± 0.54	25.5 ± 0.33	20.27 ± 0.98	16.3 ± 0.14	12.4 ± 0.29	6.93 ± 0.19	3.63 ± 0.12
AT Free	67.62 ± 0.24	63.27 ± 0.72	59.53 ± 0.31	55.77 ± 0.28	47.02 ± 3.83	33.52 ± 9.24	7.87 ± 1.78	20.92 ± 21.48
	48.07 ± 0.31	37.93 ± 0.69	29.7 ± 0.51	24.43 ± 0.37	3.23 ± 4.43	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ZeroGrad	71.68 ± 0.07	67.2 ± 0.14	63.69 ± 0.14	60.77 ± 0.26	61.05 ± 0.38	58.39 ± 0.16	56.19 ± 0.11	56.38 ± 0.18
	52.63 ± 0.61	39.57 ± 0.33	30.27 ± 0.54	23.7 ± 0.08	15.1 ± 0.49	11.13 ± 0.68	8.8 ± 0.36	4.9 ± 0.36
MultiGrad	71.8 ± 0.15	67.73 ± 0.48	63.24 ± 0.33	60.05 ± 0.79	56.39 ± 0.49	56.79 ± 8.27	59.8 ± 3.77	52.96 ± 5.58
	51.9 ± 0.29	39.7 ± 0.37	31.5 ± 0.62	26.03 ± 0.09	20.8 ± 0.29	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
PGD-2	71.62 ± 0.15	67.25 ± 0.43	63.18 ± 0.36	59.02 ± 0.4	54.47 ± 0.45	50.91 ± 0.35	41.03 ± 3.18	40.13 ± 3.66
	51.73 ± 0.48	**40.27 ± 0.7**	32.23 ± 0.19	27.13 ± 0.37	23.43 ± 0.31	20.23 ± 0.39	0.03 ± 0.05	0.0 ± 0.0
PGD-10	71.11 ± 0.62	66.9 ± 0.57	62.05 ± 0.47	57.64 ± 0.81	52.84 ± 0.88	48.14 ± 0.73	43.14 ± 0.87	39.2 ± 0.62
	52.5 ± 0.59	**40.73 ± 0.56**	**32.8 ± 0.29**	**27.97 ± 0.59**	**24.7 ± 0.36**	**21.8 ± 0.57**	**18.87 ± 0.6**	**16.8 ± 0.57**

WideResNet28-10 – SVHN Dataset

	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 6/255$	$\epsilon = 8/255$	$\epsilon = 10/255$	$\epsilon = 12/255$
N-FGSM	95.64 ± 0.09	93.66 ± 0.41	91.77 ± 0.42	88.89 ± 0.58	88.07 ± 0.59	87.52 ± 0.49
	84.1 ± 0.73	66.9 ± 0.86	**53.0 ± 0.36**	**40.5 ± 0.37**	**30.47 ± 0.76**	**22.43 ± 0.53**
Grad Align	95.41 ± 0.06	93.9 ± 0.48	68.36 ± 34.49	42.62 ± 32.73	19.3 ± 0.21	19.53 ± 0.08
	84.57 ± 0.56	67.27 ± 0.54	39.53 ± 14.89	24.7 ± 9.34	17.63 ± 0.62	18.13 ± 0.52
FGSM	95.83 ± 0.1	95.0 ± 0.24	94.23 ± 0.79	91.11 ± 1.36	88.83 ± 1.71	86.74 ± 0.7
	85.03 ± 0.37	31.53 ± 6.57	1.7 ± 1.36	0.13 ± 0.19	0.0 ± 0.0	0.0 ± 0.0
RS-FGSM	95.81 ± 0.25	94.53 ± 0.4	95.23 ± 0.26	94.68 ± 0.62	93.9 ± 0.52	91.64 ± 2.98
	83.8 ± 0.43	66.67 ± 0.65	0.53 ± 0.26	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RandAlpha	96.02 ± 0.23	95.47 ± 0.18	94.69 ± 0.26	93.72 ± 0.44	93.08 ± 1.45	93.96 ± 0.68
	82.5 ± 0.45	63.33 ± 0.53	47.7 ± 0.99	35.73 ± 0.34	23.17 ± 1.97	11.1 ± 3.05
AT Free	94.85 ± 0.39	92.95 ± 0.65	91.62 ± 1.93	93.74 ± 0.69	92.47 ± 0.97	90.5 ± 1.41
	83.13 ± 0.17	**68.67 ± 0.53**	**54.93 ± 2.58**	0.03 ± 0.05	0.0 ± 0.0	0.0 ± 0.0
ZeroGrad	95.78 ± 0.21	94.06 ± 0.52	92.13 ± 0.98	91.04 ± 0.4	88.85 ± 0.92	89.8 ± 1.36
	84.47 ± 0.83	66.1 ± 0.37	47.3 ± 0.62	29.33 ± 0.56	20.77 ± 0.63	9.33 ± 0.76
MultiGrad	95.63 ± 0.16	94.27 ± 0.38	93.64 ± 1.21	94.83 ± 1.55	95.26 ± 0.34	95.22 ± 0.15
	84.37 ± 0.59	67.27 ± 0.31	50.1 ± 0.9	1.77 ± 1.72	0.0 ± 0.0	0.0 ± 0.0
PGD-2	95.88 ± 0.35	94.66 ± 0.1	93.77 ± 0.61	92.99 ± 1.11	88.81 ± 0.93	83.17 ± 4.78
	86.25 ± 0.7	73.29 ± 0.25	60.53 ± 0.72	40.77 ± 4.39	34.33 ± 2.76	26.8 ± 3.31
PGD-10	95.92 ± 0.08	94.36 ± 0.13	92.46 ± 0.25	89.67 ± 0.34	85.98 ± 0.59	80.08 ± 0.93
	86.94 ± 0.13	**74.46 ± 0.54**	**63.87 ± 0.49**	**53.95 ± 0.55**	**44.59 ± 0.14**	**37.64 ± 0.49**