TOWARDS FAST AND EFFECTIVE SINGLE-STEP ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, Wong et al. (2020) showed 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. Moreover, they showed adding a random perturbation prior to FGSM (RS-FGSM) seemed to be sufficient to prevent CO. However, Andriushchenko & Flammarion (2020) observed that RS-FGSM still leads to CO for larger perturbations and argue that the only contribution of the random step is to reduce the magnitude of the attacks. They suggest a regularizer (GradAlign) that avoids CO but is significantly more expensive than RS-FGSM. In this work, we methodically revisit the role of noise and clipping in single-step adversarial training. Contrary to previous intuitions, we find that *not clipping* the perturbation around the clean sample and using a *stronger noise* is highly effective in avoiding CO for large perturbation radii, despite leading to an increase in the magnitude of the attacks. Based on these observations, we propose a method called *Noise-FGSM* (N-FGSM), which attacks noise-augmented samples directly using a single-step. Empirical analyses on a large suite of experiments show that N-FGSM is able to match or surpass the performance of GradAlign while achieving a 3x speed-up.

1 INTRODUCTION

Deep neural networks have achieved remarkable performance on a variety of tasks (He et al., 2015; Silver et al., 2016; Devlin et al., 2019). However, it is well known that they are vulnerable to small worst-case perturbations around the input data – commonly referred to as *adversarial examples* (Szegedy et al., 2014). The existence of such adversarial examples poses a security threat to deploying models in sensitive environments (Biggio & Roli, 2018). This has motivated a large body of work towards improving the *adversarial robustness* of neural networks (Goodfellow et al., 2015; Papernot et al., 2016; Tramèr et al., 2018).

The most popular family of solutions to obtain robust neural networks is based on the concept of *adversarial training* (Goodfellow et al., 2015; Madry et al., 2018). 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 (Weng et al., 2018) and hence, different approaches have been suggested to approximate them. They are commonly referred to as *adversarial attacks*. In their seminal work, Goodfellow et al. (2015) proposed the *Fast Gradient Sign Method* (FGSM), that generates adversarial attacks by running one step of gradient ascent on the loss function. However, while FGSM-based adversarial training provides robustness against single-step FGSM adversaries, Madry et al. (2018); Tramèr et al. (2018) showed that these models were still vulnerable to multi-step attacks, namely those allowed to perform multiple gradient ascent steps instead of a single one. Notably, Madry et al. (2018) introduced the multi-step *Projected Gradient Descent* (PGD) attack.

PGD-based attacks have now become the de facto standard for adversarial training; yet, their 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 worst-case perturbations with single-step attacks (Wong et al., 2020; Shafahi et al., 2019; Vivek & Babu, 2020). In particular, Wong et al. (2020) studied FGSM adversarial training and discovered that it suffers from a characteristic failure

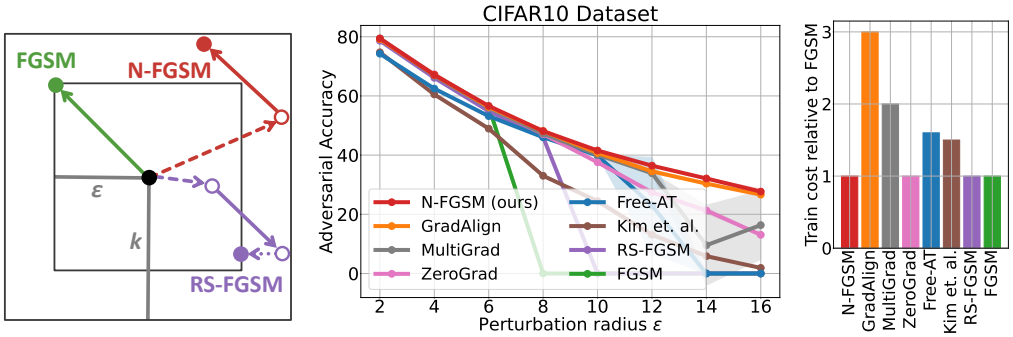


Figure 1: **Left:** Visualization of FGSM (Goodfellow et al., 2015), RS-FGSM (Wong et al., 2020) and N-FGSM (ours) attacks. While RS-FGSM is limited to noise in the $\epsilon - l_\infty$ ball, N-FGSM draws noise from an arbitrary $k - l_\infty$ ball. Moreover, N-FGSM does not clip the perturbation around the clean sample. **Middle:** Comparison of single-step methods on CIFAR-10 with PreactResNet18 over different perturbation radii ϵ. Our method, N-FGSM, is able to match or surpass the 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 I for details.

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*. Moreover, they argued that adding a random perturbation prior to FGSM (RS-FGSM) seemed sufficient to prevent catastrophic overfitting and yield robust models. Recently, Andriushchenko & Flammarion (2020) observed that RS-FGSM still leads to catastrophic overfitting as we increase the perturbation radii. They suggested a regularizer (GradAlign) that, on the one hand avoids catastrophic overfitting in all the settings they considered, but on the other hand requires the computation of a double derivative – which significantly increases the computational cost compared to RS-FGSM. This has motivated other works that aim at achieving the same level of robustness with a lower computational overhead (Golgooni et al., 2021; Kim et al., 2021).

In this paper, we revisit two key components that are common among previous works combining noise and FGSM (Tramèr et al., 2018; Wong et al., 2020): the role of noise, i.e. the *random step*, and the role of the *clipping step*. In Section 4.1, we study how these two components affect model robustness; our experiments suggest that adding noise with a large magnitude in the *random step* and removing the *clipping step* improves model robustness and prevents catastrophic overfitting, even against large perturbation radii. We combine these observations and propose a new method called Noise-FGSM (N-FGSM), an illustration of which is presented in Figure 1 (left). N-FGSM allows to match, or even surpass, the robust accuracy of the regularized FGSM introduced by Andriushchenko & Flammarion (2020), while *providing a* $3\times$ *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 other single-step approaches, 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 reproducing all experiments.

2 Related Work

Since the discovery of adversarial examples, many defense mechanisms have been proposed. *Pre-processing techniques* try to modify the input image to neutralize adversarial attacks (Guo et al., 2018; Buckman et al., 2018; Song et al., 2018). *Adversarial detection* methods focus on detecting and rejecting adversarial attacks (Carlini & Wagner, 2017; Ma et al., 2018; Yang et al., 2020; Tian et al., 2021). *Certifiable defenses* provide theoretical guarantees for the lower bound performance of networks subjected to worst-case adversarial attacks, however, they incur additional costs during inference and, empirically, they yield sub-optimal performance (Cohen et al., 2019; Wong &

Kolter, 2018; Raghunathan et al., 2018; Balunovic & Vechev, 2020). *Adversarial training* methods are based on a special form of data augmentation designed to make the network robust to worst-case perturbations (Zhang et al., 2019; Athalye et al., 2018; Kurakin et al., 2017). However, computing a worst-case perturbation is an NP-hard problem that needs to be solved at every iteration. To minimize the overhead of adversarial training, Goodfellow et al. (2015) proposed FGSM which requires one additional gradient step per iteration. Tramèr et al. (2018) first proposed performing a random step before taking the adversarial step (R+FGSM), but they observed that neither method yields robust models against PGD attacks (Madry et al., 2018). Since then, augmenting the training with PGD attacks has been one of the most popular approaches for robustness, but its cost increases linearly with the number of steps, which presents a severe practical limitation.

To reduce the cost of PGD, Shafahi et al. (2019) proposed *Free Adversarial Training* (Free-AT), that exploits a single back-propagation step to both update the network parameters and compute the attack. Wong et al. (2020) explored a variation of R+FGSM, namely RS-FGSM, and showed it can yield robust networks against multi-step attacks. Andriushchenko & Flammarion (2020) found that RS-FGSM only works for limited perturbation radii and introduced GradAlign – a regularizer to linearize the loss surface. However, optimizing GradAlign *triplicates* the computational cost. This motivated a new series of works that aim at matching the performance of GradAlign without the additional computational overhead (Golgooni et al., 2021; Kim et al., 2021). Other strategies that attempted to improve FGSM included introducing dropout in every layer (Vivek & Babu, 2020) and perturbing intermediate feature maps together with the input (Park & Lee, 2021). Li et al. (2020) suggested combining RS-FGSM and PGD attacks during training, however, the proposed strategy requires a frequent monitoring of the PGD robust accuracy and, in the worst-case, is computationally equivalent to PGD training.

Gilmer et al. (2019); Fawzi et al. (2018) suggested a strong link between robustness to adversarial attacks and to random noise. Motivated by this, we revisit the idea of combining noise and FGSM and propose N-FGSM. Our method is closely related to RS-FGSM, however, we find that using a larger amount of noise *and* removing the constraint that attacks must lie in the $\epsilon - l_\infty$ ball is key to obtaining robust models. We note that Kang & Moosavi-Dezfooli (2021) concurrently studied RS-FGSM without clipping, however, as opposed to our work, they did not investigate and provide insights on the impact of noise, and the learned models were not robust against large perturbations.

3 Preliminaries on single-step Adversarial Training

Given a classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ and a perturbation set $\mathcal{S}$, the classifier f_θ is said to be *robust* at $x \in \mathcal{X}$ under $\mathcal{S}$ if the following holds for all $\delta \in \mathcal{S}$: $f_\theta(x+\delta) = f_\theta(x)$. 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 and is the setting 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. trains on adversarially perturbed samples $\{(x_i+\delta_i, y_i)\}_{i=1:N}$. When using the l_∞ threat model, we can formalize adversarial training as solving the following problem:

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

where $\mathcal{L}$ is typically the cross-entropy loss for image-classification models. 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 (Madry et al., 2018). While this has been shown to yield robust models, it comes at a cost of a linear increase in the computational overhead with the number of PGD steps. As a result, several works have focused on reducing the cost of adversarial training by approximating the inner maximization with single-step attacks.

If we assume that the loss function is locally linear with respect to changes in the input, then the inner maximization of Equation (1) enjoys a closed form solution. Goodfellow et al. (2015) leveraged this result to propose the FGSM method, which takes one step in the direction of the sign of the gradient. Tramèr et al. (2018) proposed adding a random initialization prior to FGSM. However, both

methods were later shown to be vulnerable against multi-step attacks, such as PGD (Madry et al., 2018). Contrary to prior intuition, recent work from Wong et al. (2020) observed that combining a random step with FGSM can actually lead to a promising robustness performance. In particular, we note that most recent single-step methods approximate the worst-case perturbation that results from solving the inner maximization problem in Equation (1) with the following general form:

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

and Ω is the distribution from which we draw noise perturbations. For example, when ψ is the projection operator onto the ℓ_∞ ball and Ω is the uniform distribution in $[-\epsilon, \epsilon]$, this recovers RS-FGSM with the following update:

$$\delta_{\text{RS-FGSM}} = \text{Proj}_{\|\delta\|_\infty \leq \epsilon}\Big(\eta + \alpha \cdot \text{sign}\big(\nabla_{x_i}\mathcal{L}(f(x_i+\eta), y_i)\big)\Big), \quad \text{where} \;\; \eta \sim \mathcal{U}[-\epsilon, \epsilon]^d. \tag{3}$$

On the other hand, with a different noise setting where $\Omega = (\epsilon - \alpha) \cdot \text{sign}\left(\mathcal{N}(\mathbf{0}, \mathbf{I})\right)$ and by choosing the step size α to be in $[0, \epsilon]$ we recover R+FGSM by Tramèr et al. (2018) that initially explored the idea of combining noise with FGSM but reported no improvement over adversarial training with FGSM. If we consider Ω to be deterministically 0 and ψ to be the identity map, we recover the FGSM. Finally, if we adjust the choice of the loss function $\mathcal{L}$ to include a gradient alignment regularizer, this recovers the GradAlign algorithm by Andriushchenko & Flammarion (2020).

4 Noise and FGSM

A common practice when performing adversarial training is to restrict the perturbations used during training to the same $\epsilon - \ell_\infty$ ball that will be considered at test time. The rationale behind it is that increasing the magnitude of perturbations could "unnecessarily" decrease the clean accuracy, since perturbations outside the ball will not be evaluated at test time. For instance, R+FGSM combines the noise step, with magnitude $(\epsilon - \alpha)$, and the FGSM step, with magnitude α, in a convex combination manner, thereof, restricting the perturbation to ϵ. On the other hand, Wong et al. (2020) apply a clipping operation after the FGSM step to the $\epsilon - \ell_\infty$ ball. In the following, we experimentally challenge this common practice and explore the two key components in previous single-step methods that limit the magnitude of the perturbations. In particular, we explore **(i)** the role of the clipping operation, i.e. ψ as a projection to ℓ_∞ ball; and **(ii)** the source and magnitude of noise for the random step, i.e. Ω. We thoroughly revisit the role of both components on the robustness attained in single-step methods. Throughout this work, unless stated otherwise, we consider noise perturbations η sampled from a symmetric Uniform distribution, i.e. $\Omega = \mathcal{U}[-k, k]^d$, where d is the dimension of $\mathcal{X}$ and we refer to k as the "noise magnitude".

4.1 The Role of Noise and Clipping on Robustness in Single-Step Methods

Clipping Reduces the Effectiveness of Single-Step Perturbations. To study the impact clipping has on model robustness, we consider the training of PreActResNet18 (He et al., 2016) on CIFAR-10 (Krizhevsky & Hinton, 2009) with an instance of Equation (2) as a single-step adversarial training. In particular, we consider the case where ψ is a projection to the ℓ_∞ ball of size $\epsilon_{\text{clip}} \in \{\epsilon, 2\epsilon, 3\epsilon, \infty\}$, where ∞ denotes that ψ is an identity function, i.e. no clipping is performed. Moreover, we consider noise sampled from a symmetric uniform distribution where $k \in \{\epsilon, 2\epsilon, 3\epsilon, 4\epsilon\}$. We report in Figure 2 the robust accuracy using PGD-50-10 (i.e. PGD attack with 50 iterations and 10 restarts) with $\epsilon = {}^8/_{255}$. We observe in Figure 2 (left), that for all choices of noise magnitude k, as we expand the clipping ℓ_∞ ball, i.e. we increase ϵ_{clip}, the adversarial robustness improves. We believe that this is due to the fact that more aggressive clipping, i.e. smaller ϵ_{clip}, reduces the strength of the computed single-step perturbations during training. To support this intuition, we report in Figure 2 (middle) the distribution of the loss measured at perturbed points prior to applying the clipping step and after applying the clipping step, with $\epsilon_{\text{clip}} = \epsilon$. Moreover, the negative impact of clipping during training is more prominent as we increase the noise magnitude k. This is to be expected since, for a fixed α, increasing the noise magnitude k will lead to a prevalence of the noise component over the sign gradient direction in Equation (2).

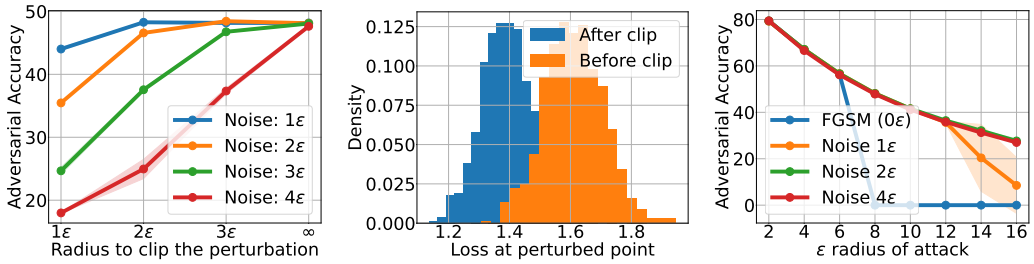


Figure 2: **Left:** N-FGSM + Clipping to different radii (∞ means no clipping). As we constraint the perturbations by reducing the clipping radius, adversarial accuracy drops. This effect is stronger as we increase the noise magnitude. Thus, clipping the perturbations seems to have a negative impact on model robustness. **Middle:** Histogram of the loss value for perturbations before and after clipping to the $\epsilon - \ell_\infty$ ball. There is a clear shift in the distributions, which indicates that clipping reduces the adversarial effect of the perturbations. **Right**: N-FGSM when varying the noise magnitude k. Increasing the amount of noise is key to avoiding catastrophic overfitting. For (left) and (right) plots, adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds.

Thus, overall, we observe that clipping during training has negative impact on the robust accuracy. Despite unclipped perturbations may lie outside the $\epsilon - \ell_\infty$ ball, we could not observe any significant drop in clean accuracy. In fact, we obtain comparable clean accuracy to GradAlign as later reported in Section 5.5.

The role of noise in single-step adversarial training. Neither Tramèr et al. (2018) nor Wong et al. (2020) explored settings with an increased noise magnitude. Moreover, Andriushchenko & Flammarion (2020) argue that noise in RS-FGSM is not important per se, claiming that its main purpose is only reducing the ℓ_2 norm of the final perturbation $\delta_{\text{RS-FGSM}}$ so that the loss is still in the linear regime. However, we empirically find that increasing the noise magnitude is key to avoiding catastrophic overfitting – see Figure 2 (right).

Investigating further, similarly to Kim et al. (2021), we plot the loss surface at the end of training (see Figure 12 in Appendix H) and find that, as observed by Kim et al. (2021), the loss surface of models trained via FGSM or RS-FGSM appears distorted at the end of training, i.e. the loss increases sharply along the FGSM direction, but then it rapidly decreases back to the same loss values associated with the clean sample. This is consistent with the gradient obfuscation observed by Tramèr et al. (2018). However, when training without clipping with an increased noise magnitude, we observe a non-distorted loss surface, i.e. the loss gradually increases along the FGSM direction. Interestingly, we observe a similar effect when training with the GradAlign regularizer. Thus, combining noise with FGSM *seems* to have a regularizing effect that encourages the loss surface to be locally linear in a similar spirit to GradAlign. Note this is based on the empirical analyses and visual inspections that we performed. We do not provide theoretical justification behind this.

Despite the clear benefits of increasing the noise magnitude, we do observe a slight but consistent decrease in robustness as we keep on increasing the noise, which we hypothesize is due to over-regularization. We find $k = 2\epsilon$ to be the sweet spot in most of our experiments; we do not exclude that a more extensive hyperparameter tuning procedure may lead to improved results.

4.2 Our Approach

In the previous section, we provided experimental analyses to show that increasing the noise magnitude and not clipping prevents catastrophic overfitting and improves robustness significantly. Based on these observations, we propose the following, simple and efficient, single-step method for adversarial training that we denote as Noise-FGSM (N-FGSM):

$$\delta_{\text{N-FGSM}} = \eta + \alpha \cdot \text{sign}\big(\nabla_{x_i} \mathcal{L}(f_\theta(x_i + \eta), y_i)\big), \quad \eta \sim \mathcal{U}[-k, k]^d. \tag{4}$$

We detail our full adversarial training procedure in Algorithm 1.

Theoretical insights We now theoretically analyse N-FGSM to understand the role of noise in single-step approaches. According to Andriushchenko & Flammarion (2020), the main contribution

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**
    // Perform N-FGSM adversarial attack
    $\eta \sim \text{Uniform}[-k, k]^d$
    $\delta = \eta + \alpha \cdot \text{sign}\big(\nabla_{x_i} \mathcal{L}(f_\theta(x_i + \eta), y_i)\big)$
    $\nabla_\theta = \nabla_\theta \mathcal{L}(f_\theta(x_i + \delta), y_i)$
    $\theta = \text{optimizer}(\theta, \nabla_\theta)$ *// standard weight update, (e.g. SGD)*
  end for
end for

of noise in single-step adversarial training is to reduce the effective ℓ_2 norm of the perturbation δ. Since N-FGSM does not involve clipping, we find the expected norm squared of N-FGSM perturbations to be larger than RS-FGSM perturbations. In fact, it is even larger than the FGSM perturbations which always lie on the $\epsilon - \ell_\infty$ ball. These observations are formalized in Theorem 1.

Theorem 1. *Let $\delta_{\text{N-FGSM}}$ be our proposed single-step method defined by Equation (4), δ_{FGSM} be the FGSM method (Goodfellow et al., 2015) and $\delta_{\text{RS-FGSM}}$ be the RS-FGSM method (Wong et al., 2020). Then, with default hyperparameter values and for any $\epsilon > 0$, we have that*

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

We present the *Proof* in Appendix D where we also compute an empirical estimation of the expected ℓ_2 norm of different methods via Monte Carlo sampling and find them to align with Theorem 1. Differently from the hypothesis of Andriushchenko & Flammarion (2020), we find that despite that N-FGSM perturbations have larger ℓ_2 norms, they yield robust models even under larger ϵ radii, for which RS-FGSM or FGSM fail to catastrophic overfitting. We believe that these contradictory observations can lead to a better understanding on the role of noise in adversarial training and catastrophic overfitting in future work.

5 Experiments and analyses

We compare N-FGSM against several adversarial training methods, considering a broad range of $\epsilon - l_\infty$ radii. Following Wong et al. (2020); Andriushchenko & Flammarion (2020), we measure adversarial robustness with PGD-50-10 attack (Madry et al., 2018) on CIFAR-10/100 (Krizhevsky & Hinton, 2009) and SVHN (Netzer et al., 2011) datasets.

5.1 Comparison to other 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 (Andriushchenko & Flammarion, 2020) relies on a regularizer that is considerably expensive, while, MultiGrad (Golgooni et al., 2021) requires evaluating the input gradients on multiple random points. For a comparison of training cost over different single-step methods, we refer to Figure 1 (right). Following the standard practice (Wong et al., 2020; Andriushchenko & Flammarion, 2020), we use a PreactResNet18 architecture (He et al., 2016).

We use RS-FGSM and Free-AT with the settings recommended by Wong et al. (2020). We apply GradAlign following the hyperparameters reported in the official repository [1]. Golgooni et al. (2021) recommend applying MultiGrad with $n = 3$ random samples, but do not provide a default hyperparameter setting for what concerns the ZeroGrad variant. Also Kim et al. (2021) do not recommend a set of hyperparameters; for a fair comparison, we ablate them and select the ones that provide the highest adversarial accuracy (for every combination of 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

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

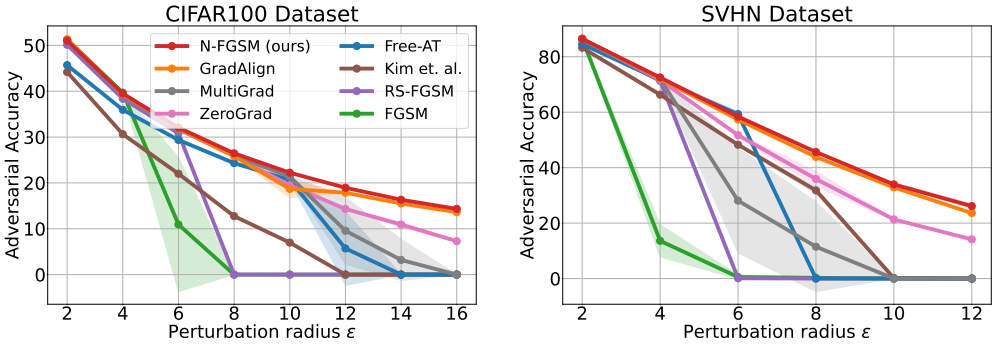


Figure 3: Comparison of single-step methods on CIFAR-100 (left) and SVHN (right) with PreactResNet18 over different perturbation radius ϵ. Our method, N-FGSM, is able to match or surpass the state-of-the-art results while *reducing the cost by a* $3\times$ *factor*. Adversarial accuracy is based on PGD-50-10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

Wong et al. (2020) and Andriushchenko & Flammarion (2020). CIFAR-10 results are presented in Figure 1 (middle), whereas CIFAR-100 and SVHN results are reported in Figure 3.

As previously observed, FGSM and RS-FGSM both suffer from catastrophic overfitting on larger ϵ attacks. In contrast, N-FGSM prevents catastrophic overfitting and enjoys robustness properties comparable or superior to GradAlign for all ϵ, *while being 3 times faster.* With appropriate hyperparameters, ZeroGrad is able to avoid catastrophic overfitting but obtains sub-obtimal robustness – especially for large perturbations. Neither MultiGrad nor the method proposed by Kim et al. (2021) can avoid catastrophic overfitting in all settings. We also observe that Free-AT cannot overcome catastrophic overfitting as also observed by Andriushchenko & Flammarion (2020).

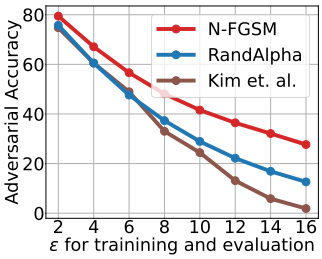
5.2 Randomized Alpha

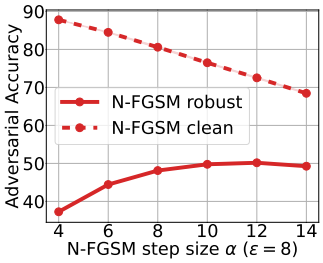
Kim et al. (2021) evaluate intermediate points along the RS-FGSM direction in order to pick the "optimal" perturbation size. However, we find that increasing the number of intermediate evaluated points does not necessarily lead to increased adversarial accuracy. Moreover, for large perturbations we could not prevent catastrophic overfitting even with twice the number of evaluations tested by Kim et al. (2021). 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 4 (left), we find that this very simple baseline, dubbed *RandAlpha*, is able to avoid catastrophic overfitting for all values of ϵ and outperforms Kim et al. (2021) on CIFAR-10. 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. We reach the same conclusions for CIFAR-100 and SVHN in Appendix F.

5.3 Hyperparameter selection

While FGSM relies on a fixed step-size (equal to the maximum radius of perturbation to be used at test time, i.e. $\alpha = \epsilon$), Wong et al. (2020) explored different values for α during the development of RS-FGSM finding that an increase of the step-size improves the adversarial accuracy – up to a magnitude before catastrophic overfitting occurs. We also ablate the value of α for N-FGSM in Figure 4 (middle). We find that by increasing the noise magnitude, N-FGSM can use larger α values than RS-FGSM, without suffering from catastrophic overfitting. As observed by Wong et al. (2020), this in turn leads to an increase in the adversarial accuracy at the expense of a decrease in the clean accuracy. In light of this trade-off and following FGSM, we also use $\alpha = \epsilon$ for N-FGSM.

Regarding N-FGSM noise hyperparameter, we find $k = 2\epsilon$ works in all but one SVHN experiment ($\epsilon = 12$, in which we set $k = 3\epsilon$), reducing the need for expensive hyperparameter tuning. In comparison, GradAlign regularizer hyperparameter or ZeroGrad quantile value need to be defined for every radius with a noticeable shift between CIFAR-10 and SVHN hyperparameters, suggesting they may require additional tuning when applied to novel datasets.





Comparison of Training Schedules

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

Figure 4: **Left:** Comparison of Kim et al. (2021) with *RandAlpha*, our baseline where we multiply the RS-FGSM perturbation by a value drawn uniformly in $[0, 1]$, on CIFAR-10 and PreActResNet18. RandAlpha does not incur the extra cost of evaluating intermediate steps and does not require hyperparameter tuning. **Middle:** Ablation of step size α in N-FGSM for $\epsilon = 8$. As we increase the magnitude of the FGSM perturbation we observe an increase in robustness coupled with a drop on the clean accuracy. **Right**: Comparison of the "fast" training schedule from Wong et al. (2020) and "long" training schedule described in Rice et al. (2020). N-FGSM shows robust oberfitting but not catastrophic overfitting with the long schedule. Adversarial accuracy is based on PGD-50-10 and experiments are averaged over 3 seeds.

5.4 Long vs fast training schedule

Throughout our experiments, we used the RS-FGSM training setting introduced in Wong et al. (2020). However, Rice et al. (2020) suggest that a longer training schedule coupled with early stopping may lead to a boost in performance. Kim et al. (2021) and Li et al. (2020) report that longer training schedules increase the chances of catastrophic overfitting for RS-FGSM and that this limits its performance. We test the longer training schedule with N-FGSM and find that it presents *robust overfitting*, i.e. the adversarial accuracy on the training set keeps increasing but it decreases when evaluated on the test set as described in Rice et al. (2020). However, it does not suffer from catastrophic overfitting. In Figure 4 (right), we show the results for $\epsilon = 8$. Although we do observe a slight increase in performance when using the long training schedule, we find the performance remarkably competitive when considering the fast one which seems to avoid robust overfitting (when relying on early stopping, the best models are usually found at the end). Thus, it might be preferable to consider the fast training schedule if computational cost is an important factor. In Appendix C, we report a robust accuracy of 47.86 ± 0.1 for GradAlign with the longer schedule compared to 48.48 ± 0.27 for N-FGSM. Note that, in order to prevent GradAlign from suffering from catastrophic overfitting for $\epsilon = 8$, we had to increase the regularizer hyperparameter (compared to the fast schedule), while N-FGSM is able to prevent catastrophic overfitting with the same settings.

5.5 Comparison to 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 $3\times$. In this section, we compare the performance of N-FGSM with multi-step attacks. We use PGD-2 with $\alpha = \epsilon/2$ and PGD-10 with $\alpha = 2$ following Wong et al. (2020) and keep the same training settings as described in Section 5.1 (note that PGD-x denotes PGD attack with x iterations and no restarts).

In Figure 5, we observe that PGD-2 also presents catastrophic overfitting when we increase the perturbation radius ϵ, which is consistent with results reported by Andriushchenko & Flammarion (2020). On the other hand, despite all methods achieving comparable clean accuracy, there is a gap on adversarial accuracy between PGD-10 and single-step methods which grows with the perturbation size. This can be partially expected since the search space grows exponentially with ϵ and PGD can explore it more thoroughly as we increase the number of iterations. Nevertheless, *computing a PGD-10 attack is* $10\times$ *more expensive than computing an N-FGSM one*. An important direction for future work should be addressing this gap and analyse, both theoretically and empirically, whether single-step methods can actually match the performance of their multi-step counterparts.

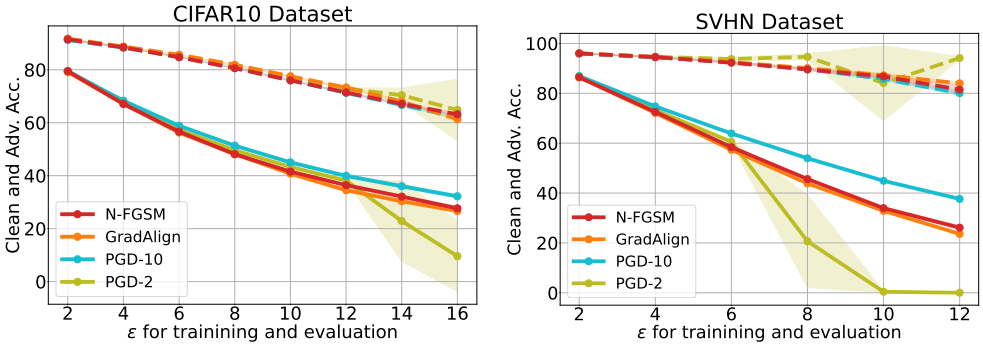


Figure 5: Comparison of N-FGSM and GradAlign with multi-step methods on CIFAR-10 (Left) and SVHN (Right) with PreactResNet18 over different perturbation radii ϵ. Despite all methods achieving comparable clean accuracy (dashed lines), there is a gap in robust accuracy between PGD-10 and single-step methods. 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.

5.6 Experiments with WideResNet28-10 architecture

We also compare the performance of all methods on WideResNet28-10 (Zagoruyko & Komodakis, 2016) architecture in Figure 7 and Figure 8 (Appendix B). As in the experiments with the PreActResNet18 architecture, N-FGSM obtains state-of-the-art PGD-50-10 accuracy among single-step methods. Nevertheless, as a general trend, we observe that catastrophic overfitting seems to be more difficult to prevent when using WideResNet. For instance, FGSM is able to consistently yield robust models up to $\epsilon = 6$ for PreActResNet18 on CIFAR-10, however, for some runs the same radius can lead to catastrophic overfitting for WideResNet models. We hypothesize this is because it is more over-parametrized –WideResNet28-10 has 36.5 M parameters, whereas PreActResNet18 has 11.2M. Regarding GradAlign, we had to increase the regularizer hyperparameter (compared to the settings for PreActResNet18) in order to prevent catastrophic overfitting on CIFAR-100. Note that, 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.

On the other hand, as observed earlier, 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 as we tried for GradAlign works better than increasing the noise. However, in both cases N-FGSM can obtain more than a trivial adversarial accuracy. Results are presented in Appendix B.

6 Discussion

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 allows improving 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 (Andriushchenko & Flammarion, 2020), 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 catastrophic overfitting for larger ϵ attacks. Moreover, we also analyze recent single-step methods and – inspired by Kim et al. (2021) – observe that uniformly choosing a step-size in $[0, \alpha]$ avoids catastrophic overfitting for RS-FGSM, which reinforces our intuition that random noise is a powerful tool towards learning robust models. 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.

Ethics Statement. The existence of adversarial examples poses a potential threat to the deployment of deep learning systems into the real world. Therefore, finding fast and effective methods to train models robust against this threat is of utmost importance. On the other hand, adversarial training methods are based on augmenting samples with adversarial perturbations, thus, they are partially based on building methods to attack neural networks. Research on adversarial attacks is naturally a sensitive path, since it can potentially be exploited for unethical purposes. However, the scope of our work is not designing stronger attacks; rather, the methods we propose are designed ad hoc to improve the adversarial robustness of learning systems, not to break other models' defenses. Therefore, although we are aware that we are carrying out research within a sensitive topic, we are not particularly concerned that the research presented in this paper can lead to harmful applications – on the contrary, we believe that it can help deploying safer machine learning applications.

Reproducibility Statement. In this paper, we compare against several adversarial training methods. The general training and evaluation settings used are described at the beginning of Section 5. Moreover, different hyperparameters have been used for each method depending on the dataset and network. These are detailed in the manuscript, where corresponding results are discussed. For all the reported experiments, we employ different random seeds to ensure replicability of our results, and report performances indicating the standard deviation values. To facilitate further research on this topic, as stated in Section 1, we will release the code to reproduce all experiments. Regarding our theoretical results, we deferred the proofs to Appendix D.

REFERENCES

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

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

Mislav Balunovic and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

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.

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.

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.

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

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

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*, 2018.

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.

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

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

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

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

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017.

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

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018.

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.

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.

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.

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

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

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

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.

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.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

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.

Jinyu Tian, Jiantao Zhou, Yuanman Li, and Jia Duan. Detecting adversarial examples from sensitivity inconsistency of spatial-transform domain. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

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.

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

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.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning (ICML)*, 2018.

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

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael Jordan. Ml-loo: Detecting adversarial examples with feature attribution. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

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

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.

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 catastrophic overfitting happens. Exact numbers for all the curves are in Appendix J.

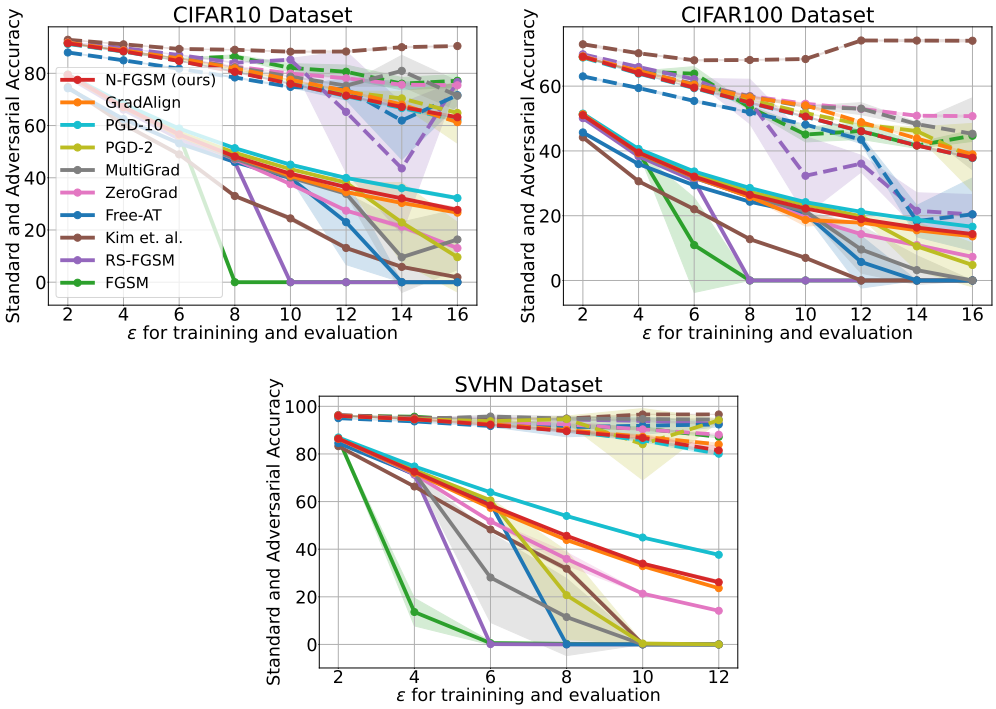


Figure 6: Comparison of all methods on CIFAR-10, CIFAR-100 and SVHN with PreactResNet18 over different perturbation radius ϵ. We plot both the robust (solid line) and the clean (dashed line) accuracy for each method. Our method, N-FGSM, is able to match or surpass the state-of-the-art single-step method 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 7 we compare all single-step methods and we do not plot the clean accuracy for better visualization. In Figure 8 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 Kim et al. (2021) in all settings for PreActResNet18, we only report RandAlpha for WideResNet. As mentioned in the main paper, we observe that catastrophic overfitting seems to be more difficult to prevent for WideResNet. In particular, for GradAlign we observed the regularizer hyperparameter settings proposed by Andriushchenko & Flammarion (2020) for CIFAR-10 (searched for a PreActResNet18) worked well. However, those parameters led to Catastrophic Overfitting for $6 \leq \epsilon \leq 12$ in CIFAR-100. Since $\epsilon = 14,\ 16$ did not show Catastrophic Overfitting, 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 Andriushchenko & Flammarion (2020) for $\epsilon = 8$.

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

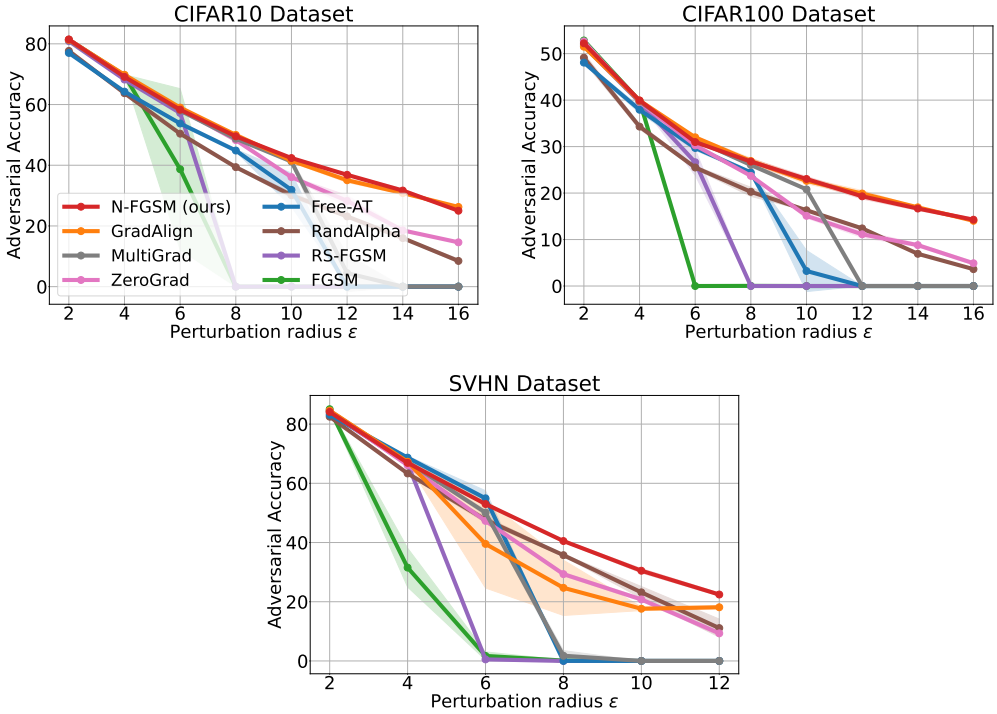


Figure 7: Comparison of single-step methods on CIFAR-10, CIFAR-100 and SVHN with WideResNet28-10 over different perturbation radius ϵ. Our method, N-FGSM, is able to match or surpass the state-of-the-art single-step method GradAlign while *reducing the cost by a* $3\times$ *factor*. Moreover, we could 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.

result. Since the model did not show typical catastrophic overfitting 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 J

C Longer Training Schedule

In our experiments, we have followed the "fast" training schedule introduced by Wong et al. (2020). However, Rice et al. (2020) 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 catastrophic overfitting. In Table 1 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 $\epsilon = 8$ and CIFAR-10 ($\lambda = 0.2$) does not prevent catastrophic overfitting. We do a hyperparameter search and keep the value with the largest final robust accuracy ($\lambda = 0.632$).

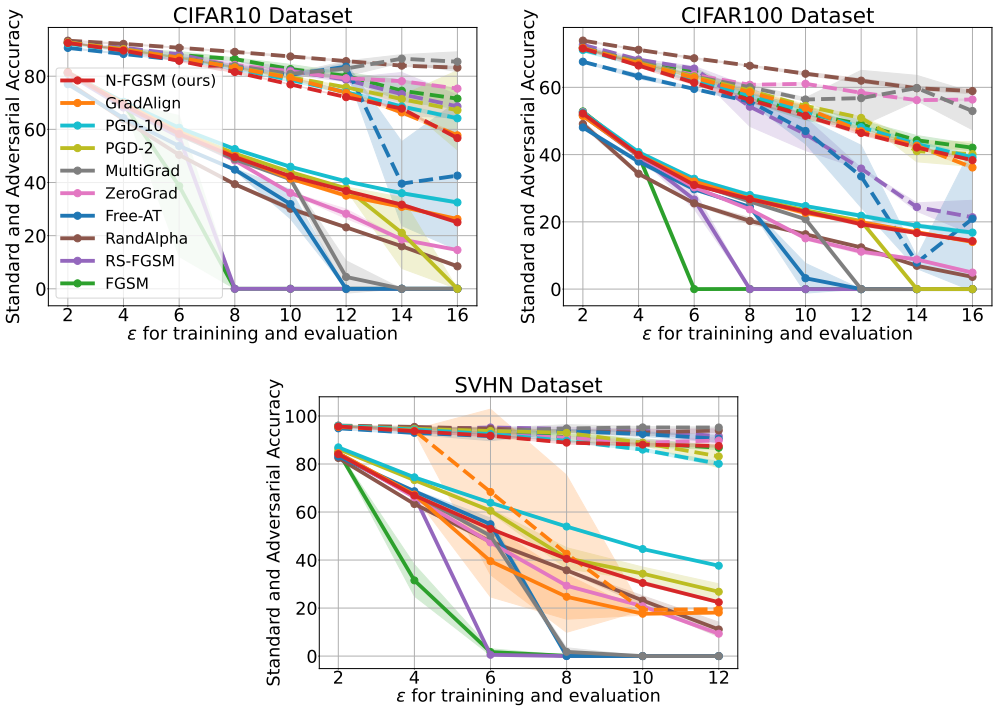


Figure 8: Comparison of all methods on CIFAR-10, CIFAR-100 and SVHN with WideResNet28-10 over different perturbation radius ϵ. We plot both the robust (solid line) and the clean (dashed line) accuracy for each method. Legend is shared among all plots.

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**

Table 1: Comparison of "long" (Rice et al., 2020) and "fast" (Wong et al., 2020) 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.

D MAGNITUDE OF N-FGSM PERTURBATIONS

Lemma 1 (Expected perturbation)**.** *Consider the N-FGSM perturbation as defined in Equation* (4)

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

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

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

and

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

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,

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

□

We state again Theorem 1 and present the proof.

Theorem 1. *Let $\delta_{N\text{-}FGSM}$ be our proposed single-step method defined by Equation (4), δ_{FGSM} be the FGSM method (Goodfellow et al., 2015) and $\delta_{RS\text{-}FGSM}$ be the RS-FGSM method (Wong et al., 2020). 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 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, Andriushchenko & Flammarion (2020) 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 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 9. 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.

E 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 2 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 catastrophic overfitting. 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.

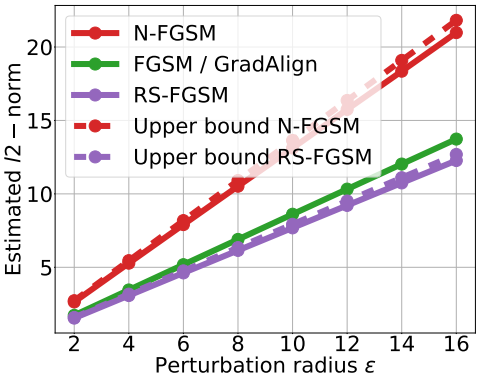


Figure 9: Monte Carlo estimations of the expected l_2-norm of perturbations from different methods and corresponding analytical upper bounds. As mentioned in Andriushchenko & Flammarion (2020), we observe that RS-FGSM perturbations have lower l_2 norm than FGSM. However, N-FGSM perturbations have a significantly 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 (Andriushchenko & Flammarion, 2020).

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

Table 2: 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.

F FURTHER RESULTS WITH RANDALPHA

In Section 5.2 we analyze the method presented by Kim et al. (2021) and suggest a baseline where, instead of evaluating intermediate points to determine the "optimal" step size, we simply choose it randomly. That is, we multiply the RS-FGSM perturbation by a random scalar sampled from a uniform distribution in $[0, 1]$. Interestingly we find that it outperforms Kim et al. (2021) without the additional cost of intermediate evaluations and without the need to perform hyperparameter selection to find the optimal number of intermediate evaluations.

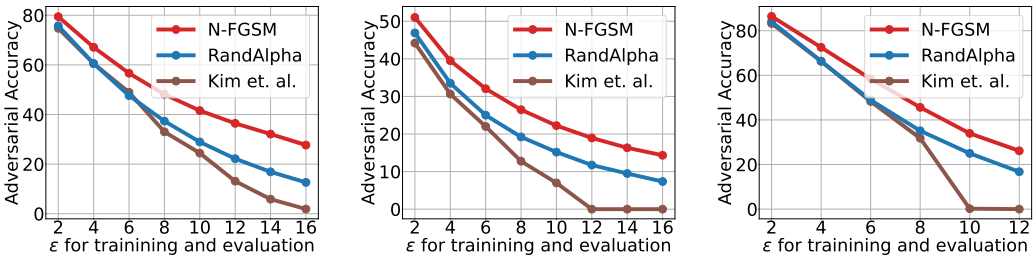


Figure 10: Comparison of Kim et al. (2021) with RandomAlpha, our baseline where we multiply the RS-FGSM perturbation by a scalar uniformly sampled in $[0, 1]$. We present results on CIFAR-10, CIFAR-100 and SVHN with PreActResNet18.

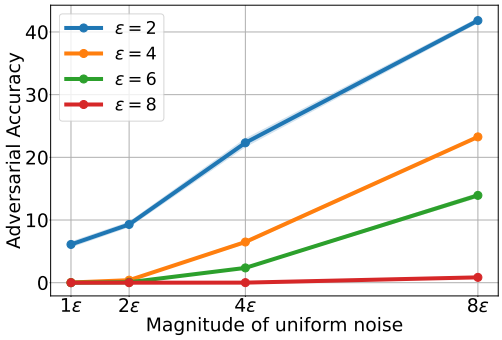


Figure 11: Training with uniform noise augmented samples improves adversarial accuracy for small perturbations but is not effective to protect against larger l_∞ radius ϵ. This motivates us to further augment the noisy samples with FGSM. All experiments are averaged over 3 runs.

G Training with noise augmented samples

Gilmer et al. (2019) and Fawzi et al. (2018) report a close link between robustness to adversarial attacks and robustness to random noise. Actually, Gilmer et al. (2019) 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 wort-case perturbations for small $\epsilon - l_\infty$ balls. However, as we increase ϵ, noise augmentation is no longer very effective. However, 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.

H Visualization of the loss surface

In this section we present a visualization of the loss surface. We adapted the code from Kim et al. (2021) to analyse the shape of the loss surface at the end of training for different methods. Kim et al. (2021) reported that after adversarial training catastrophic overfitting, 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 Tramèr et al. (2018) 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.

I 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.

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

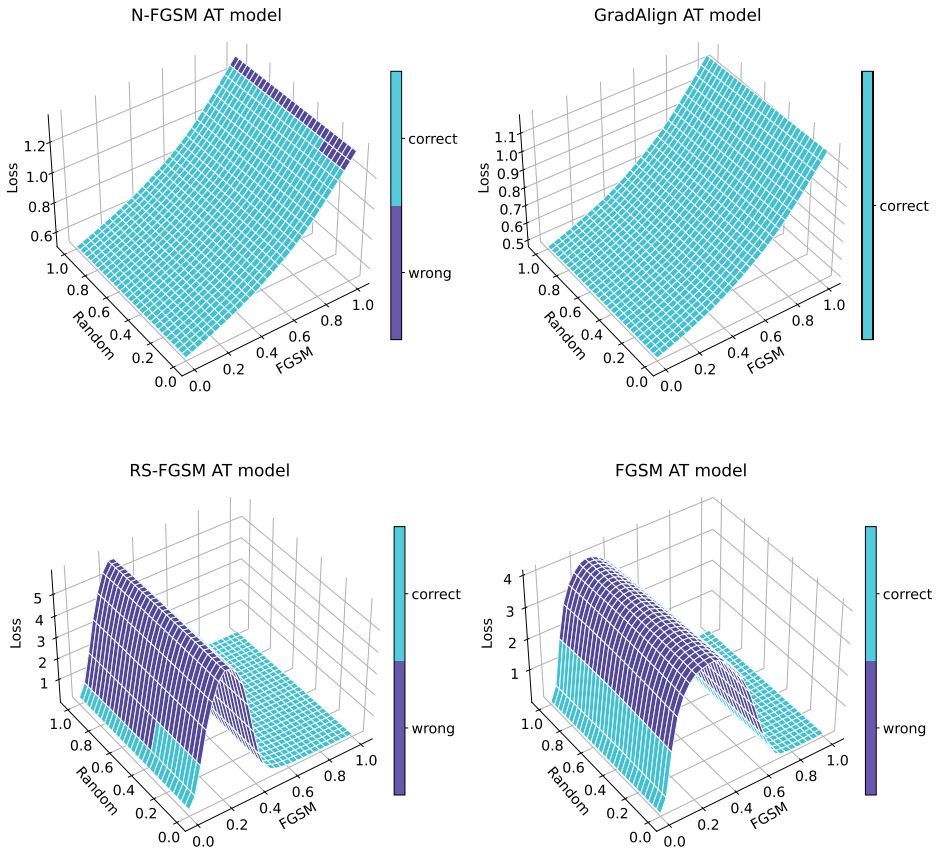


Figure 12: Visualization of the loss surface for models trained using different methods. Given a clean sample from the test set in coordinate $(0, 0)$, we compute the FGSM perturbation and evaluate the loss on the subspace generated by the FGSM perturbation direction and a random direction. That is, 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 catastrophic overfitting 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.

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.

Kim et al. (2021): Kim et al. (2021) 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. Kim et al. (2021) assume the cost of a forward is similar to that of a backward pass, following this assumption, cost of Kim et al. (2021) is 1 F/B + 2 F + 1 F/B = 3 F/B

Free-AT: Shafahi et al. (2019) 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, Wong et al. (2020) 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. Andriushchenko & Flammarion (2020) report that the cost of using GradAlign regularizer increased the cost of FGSM by 3.

J DETAILED RESULTS FOR SECTION 5.1 AND SECTION 5.6

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

	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$	$\epsilon = 12$	$\epsilon = 14$	$\epsilon = 16$
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
AAAI	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

	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$	$\epsilon = 12$	$\epsilon = 14$	$\epsilon = 16$
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
AAAI	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

	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$	$\epsilon = 12$
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
AAAI	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

	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$	$\epsilon = 12$	$\epsilon = 14$	$\epsilon = 16$
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

	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$	$\epsilon = 12$	$\epsilon = 14$	$\epsilon = 16$
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

	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$	$\epsilon = 12$
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**