Revisiting Loss Landscape for Adversarial Robustness

Dongxian Wu 1  Yisen Wang 2  Shu-Tao Xia 1

Abstract
The study on improving the robustness of deep neural networks against adversarial examples grows rapidly in recent years. Among them, adversarial training is the most promising one, based on which, a lot of improvements have been developed, such as adding regularizations or leveraging unlabeled data. However, these improvements seem to come from isolated perspectives, so that we are curious about if there is something in common behind them. In this paper, we investigate the surface geometry of several well-recognized adversarial training variants, and reveal that their adversarial loss landscape is closely related to the adversarially robust generalization, i.e., the flatter the adversarial loss landscape, the smaller the adversarially robust generalization gap. Based on this finding, we then propose a simple yet effective module, Adversarial Weight Perturbation (AWP), to directly regularize the flatness of the adversarial loss landscape in the adversarial training framework. Extensive experiments demonstrate that AWP indeed owns flatter landscape and can be easily incorporated into various adversarial training variants to enhance their adversarial robustness further.

1. Introduction
Although deep neural networks (DNNs) have been widely deployed in a number of fields such as computer vision (He et al., 2016), speech recognition (Wang et al., 2017), and natural language processing (Devlin et al., 2019), they could be easily fooled to make incorrect predictions by adversarial examples that are crafted by adding intentionally small perturbations to normal examples (Szegedy et al., 2013; Goodfellow et al., 2015). As DNNs penetrate almost everywhere in our everyday life, ensuring their security, e.g., improving their robustness against adversarial examples, becomes more and more important.

There have emerged a number of defense techniques to improve adversarial robustness of DNNs (Papernot et al., 2016b; Xu et al., 2017; Guo et al., 2018; Madry et al., 2018; Wang et al., 2019). Across these defenses, adversarial training (Goodfellow et al., 2015; Madry et al., 2018) is the most effective and promising approach, which not only demonstrates moderate robustness, but also has thus far not been comprehensively attacked (Athalye et al., 2018). Adversarial training directly incorporates adversarial examples into the training process to solve the following optimization problem:

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \phi_{x_i, y_i}(w),$$

where $n$ is the number of training examples, and

$$\phi_{x_i, y_i}(w) = \max_{\|x'_i - x_i\|_p \leq \epsilon} \ell(f_w(x'_i), y_i).$$

Here, $x'_i$ is the adversarial example that is within the $\epsilon$-ball (bounded by an $L_p$-norm) centered at natural example $x_i$, $f_w$ is the DNN function with parameter/weight $w$, and $\ell(\cdot)$ is the standard classification loss, such as the commonly used cross-entropy (CE) loss. The whole maximization problem in Eq. (2), i.e., $\phi_{x_i, y_i}$, is called the adversarial loss on the example $(x_i, y_i)$ following Madry et al. (2018).

Note that adversarial training is still far from obtaining satisfactory generalization on adversarial robustness, e.g., according to the results on CIFAR-10 dataset (Madry et al., 2018), adversarial training still has $>50\%$ adversarially robust generalization gap (the difference of robust accuracy on adversarial examples of training and testing) while its standard generalization gap (the difference of natural accuracy on natural examples of training and testing) is $\sim 10\%$. Therefore, there are various adversarial training improvements proposed to eliminate the adversarially robust generalization gap, for example, based on adversarial training, Zhang et al. (2019) optimize a trade-off (robustness and accuracy) objective function called TRADES; Carmon et al. (2019); Uesato et al. (2019); Najafi et al. (2019); Zhai et al. (2019) leverage Semi-Supervised Learning (SSL) with a large amount of unlabeled data; and Wang et al. (2020) incorporate a misclassification aware regularization named MART. These above methods adopt different techniques but all improve the adversarial robustness. Then, one natural question raises:
Revisiting Loss Landscape for Adversarial Robustness

Figure 1. The adversarial loss landscape of minimal obtained by different adversarial training methods with varying $\alpha$, and their corresponding standard and robust generalization gap. In the right panel, each method has two columns: each entire column represents the training accuracy whose bottom and top is the testing accuracy (green bar and blue bar) and the generalization gap (red bar and orange bar) respectively. Left is on natural examples while right is on adversarial examples.

Is there something in common behind these different adversarial training approaches? If so, how can one make use of it to improve robustness?

In this paper, we investigate the surface geometry of the adversarial loss across several well-recognized adversarial training improvements, and find that the adversarially robust generalization gap is well-correlated with the flatness of the adversarial loss landscape. To illustrate this, in Figure 1 we visualize the 1-Dimension (1D) adversarial loss landscape and its corresponding generalization gap of standard adversarial training Madry (Madry et al., 2018), TRADES (Zhang et al., 2019), MART (Wang et al., 2020), and RST (one type of SSL) (Carmon et al., 2019) on CIFAR-10 dataset with WideResNet-28-10 (depth 28 and width 10) model (Zagoruyko & Komodakis, 2016). The landscape of each method is plotted following Li et al. (2018). Assuming the optimization minimum as $w$ (the center point), we first perform a perturbation along a random direction $d$ via one kind of “filter-wise normalization”:

$$d_{i,j} \leftarrow \frac{d_{i,j}}{\|d_{i,j}\|_F} \|w_{i,j}\|_F$$

where $d_{i,j}$ represents the $j$-th filter of the $l$-th layer of $d$ and $\| \cdot \|_F$ denotes the Frobenius norm, then plot the function: $g(\alpha) = \frac{1}{n} \sum_{i,j} \phi_{x_i,y_i}(w + \alpha d)$ where PGD-20 attack with random start and step size 0.01 is used to solve the maximization problem in $\phi_{x_i,y_i}$.

Compared with Madry, in Figure 1(a), various improvement techniques brought by TRADES, MART and RST all have one common effect, i.e., making adversarial loss landscape flatter. At the same time, in Figure 1(b), TRADES, MART and RST also all have smaller adversarially robust generalization gap (orange bar). Specifically, the flatter the adversarial loss landscape, the smaller the adversarially robust generalization gap. For example, RST has the flattest landscape amongst them, and it also has the smallest adversarially robust generalization gap. Besides, there is one phenomenon worth to mention that the robustness on the training set is not so much (see Section 4.1 for more details), which is very different from the standard generalization where near-zero error on the training set (the left entire column of each method in Figure 1(b)) can be always achieved. Therefore, the flatness of the adversarial loss landscape is closely relevant to the adversarially robust generalization gap. A successful improvement of adversarial training (higher adversarial robustness) perhaps should introduce a flatter landscape of adversarial loss and ensure an acceptable robustness drop on the training set simultaneously.

Motivated by these observations, we propose to directly regularize the adversarial loss landscape to make it flat. A flat loss landscape of optimization minimum means that its loss remains small even if perturbations are added on the minimum. Therefore, we define a Adversarial Weight Perturbation (AWP) to intentionally add the worst-case weight perturbation into the model and optimize it in the adversarial training framework. AWP is not only effective and efficient but also easy to implement and incorporate with existing adversarial training variants. Our main contributions are summarized as follows:

- We investigate the geometry of adversarial loss in the adversarial training, and reveal that the adversarially robust generalization gap is well-correlated with the flatness of the adversarial loss landscape. To be specific, the flatter the adversarial loss landscape, the smaller the adversarially robust generalization gap.

- We propose a simple yet effective module that intentionally injects a worst-case parameter/weight perturbation, Adversarial Weight Perturbation (AWP), into DNNs, which directly regularizes the flatness of the adversarial loss landscape.

- Through extensive experiments, we demonstrate a comprehensive understanding of our proposed AWP. On benchmark datasets, AWP significantly improves the adversarial robustness over state-of-the-art methods against a variety of white-box and black-box attacks.
2. Related Work

2.1. Adversarial Attack

Given a natural example $x_i$ with class label $y_i$ and a target DNN model $f_w$, the goal of an adversary is to find an adversarial example $x'_i$ that fools the network to make incorrect predictions, while still remaining in the $\epsilon$-ball (in this paper, we will focus on the $L_\infty$-ball) centered at $x_i$ ($\|x'_i - x_i\|_\infty \leq \epsilon$).

**Fast Gradient Sign Method (FGSM)** (Goodfellow et al., 2015). FGSM perturbs natural example $x_i$ for one step by the amount of $\epsilon$ along the gradient direction:

$$x'_i = x_i + \epsilon \cdot \text{sign}(\nabla_{x_i} \ell(f_w(x_i), y_i)).$$

(3)

**Projected Gradient Descent (PGD)** (Madry et al., 2018). PGD perturbs natural example $x_i$ for $K_1$ steps with smaller step size. After each step of perturbation, PGD projects the adversarial example back onto the $\epsilon$-ball of $x_i$, if it goes beyond the $\epsilon$-ball:

$$x^{(k+1)}_i = \Pi_\epsilon (x^{(k)}_i + \eta_1 \cdot \text{sign}(\nabla_{x_i} \ell(f_w(x^{(k)}_i), y_i))),$$

(4)

where $\eta_1$ is the step size, $\Pi_\epsilon(\cdot)$ is the projection operation, and $x^{(k)}_i$ is the adversarial example at the $k$-th step.

There are also other types of attacks including Jacobian-based Saliency Map Attack (JSMA) (Papernot et al., 2016a), Carlini and Wagner (CW) (Carlini & Wagner, 2017) and Feature Attack (FA) (Lin, 2019).

2.2. Adversarial Defense

Since the discovery of adversarial examples, many defense approaches have been developed to prevent this type of security risk such as defensive distillation (Papernot et al., 2016b), feature squeezing (Xu et al., 2017), input denoising (Guo et al., 2018; Liao et al., 2018; Samangouei et al., 2018; Bai et al., 2019), adversarial detection (Feinman et al., 2017; Grosse et al., 2017; Ma et al., 2018), gradient regularization (Gu & Rigazio, 2014; Papernot et al., 2017; Tramèr et al., 2018), model compression (Das et al., 2018; Liu et al., 2018), activation pruning (Dhillon et al., 2018) and adversarial training (Goodfellow et al., 2015; Madry et al., 2018). Among them, adversarial training has been demonstrated to be the most effective method (Athalye et al., 2018). Based on adversarial training, a number of new techniques are introduced to enhance its performance further.

**TRADES** (Zhang et al., 2019). TRADES optimizes an upper bound of adversarial risk that is a trade-off between accuracy and robustness:

$$\phi^{\text{TRADES}}_{x_i, y_i} = \text{CE}(f_w(x_i), y_i) + \beta \cdot \max \text{KL}(f_w(x_i) \| f_w(x'_i)),$$

(5)

where $\text{KL}$ is the Kullback–Leibler divergence, $\text{CE}$ is the cross-entropy loss, and $\beta$ is the hyperparameter to control the trade-off between natural accuracy and robust accuracy.

**MART** (Wang et al., 2020). MART incorporates an explicit differentiation of misclassified examples as a regularizer of adversarial risk:

$$\phi^{\text{MART}}_{x_i, y_i} = \text{BCE}(f_w(x'_i), y_i) + \lambda \cdot \text{KL}(f_w(x_i) \| f_w(x'_i)) \cdot (1 - |f_w(x_i)|_{y_i}).$$

(6)

where $|f_w(x_i)|_{y_i}$ denotes the $y_i$-th element of output vector $f_w(x_i)$ and $\text{BCE}(f_w(x_i), y_i) = - \log(|f_w(x'_i)|_{y_i}) - \log(1 - \max_{k \neq y_i} |f_w(x'_i)|_k)$.

**Semi-Supervised Learning (SSL)** (Carmon et al., 2019; Uesato et al., 2019; Najafi et al., 2019; Zhai et al., 2019). SSL-based methods utilize additional unlabeled data. They first generate pseudo labels for unlabeled data by training a natural model on the labeled data. Then, adversarial loss $\phi^{\text{SSL}}_{x_i, y_i}$ is applied to train a robust model based on both labeled and unlabeled data:

$$\phi^{\text{SSL}}_{x_i, y_i} = \phi^{\text{labeled}}_{x_i, y_i} + \lambda \cdot \phi^{\text{unlabeled}}_{x_i, y_i},$$

(7)

where $\lambda$ is the weight on unlabeled data, and $\phi^{\text{unlabeled}}_{x_i, y_i}$ are usually the same. For example, RST in Carmon et al. (2019) uses TRADES loss and the semi-supervised extension of MART in Wang et al. (2020) uses MART loss.

2.3. Standard and Robust Generalization

On natural examples, DNNs exhibit good standard generalization behavior, even when the number of parameters is significantly larger than the amount of training data. A number of complexity metrics are proposed to understand their standard generalization (Neyshabur et al., 2014; Dziu-}

& Roy, 2017; Bartlett et al., 2017). Amongst them, the geometry of optimization minimum (i.e., sharpness/flatness or loss landscape) is the most intuitive. The flatness can be defined as the size of the connected region around the minimum where the training loss remains small (Hochreiter & Schmidhuber, 1997). The magnitude of the eigenvalue of Hessian matrix, approximated by the maximum loss in a bounded neighborhood of the minimum, was suggested to characterize flatness in Keskar et al. (2017). However, these quantitative measures may be problematic because they are not invariant to network weight scaling, resulting in the failure to determine the generalization ability (Dinh et al., 2017). Further, local entropy addressed the network weight scaling issue, but it is difficult to accurately compute (Chaudhari et al., 2017). Except for these theoretical methods, an effective visualization technique, “filter normalization”, was proposed to visualize the loss function curvature and showed that landscape geometry has a dramatic effect on the standard generalization (Li et al., 2018).
Compared with standard generalization, training DNNs with adversarially robust generalization is particularly difficult (Madry et al., 2018), and often possesses significantly higher sample complexity (Khim & Loh, 2018; Yin et al., 2019; Montasser et al., 2019) and needs more data (Schmidt et al., 2018; Dziugaite & Roy, 2017). Nakkiran (2019) showed that a model requires more capacity to be robust (i.e., simple models can have good standard generalization but are less likely to have good adversarially robust generalization). Tsipras et al. (2019); Zhang et al. (2019) demonstrated that adversarial robustness may be inherently at odds with natural accuracy. Moreover, Prabhu et al. (2019) visualized the loss landscape of adversarial training via the standard loss (without maximization process) to find that adversarially trained DNNs do not have smoother loss landscapes.

Parallel to these studies, in this paper, we investigate and identify the relationship between the adversarial loss landscape and the adversarially robust generalization.

3. Proposed Adversarial Weight Perturbation

As indicated in Figure 1, flatter adversarial loss landscape narrows adversarially robust generalization gap. Therefore, in this section, we propose a Adversarial Weight Perturbation (AWP) method to directly regularize the flatness of adversarial loss landscape.

Revisiting Figure 1(a), we can obviously find that a flat loss landscape means that its loss remains small even if perturbations are added on the optimal parameters. Inspired by this, we propose to minimize the loss of a perturbed model $f_{w+v}$ (v denotes the weight perturbation) that is in a small region of original model $f_w$ so as to achieve a flat loss landscape. Based on Eq. (1), the objective function is then refined as:

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \phi_{x_i, y_i}(w + v), \quad (8)$$

where $\phi_{x_i, y_i}$ is the adversarial loss in Eq. (2). In the following, we will describe 1) how to choose the perturbation direction; 2) how to decide the perturbation size; and 3) how to incorporate AWP into adversarial training framework.

3.1. Direction of Weight Perturbation

With regard to the direction of weight perturbation v, the simplest method is random weight perturbation, i.e., sampling a random point in the small region around $f_w$: $v \sim \mathcal{P}$. $\mathcal{P}$ is a specific distribution such as multivariate Gaussian distribution used in Parametric Noise Injection (PNI) (He et al., 2019) which involves trainable Gaussian noise injection at each layer on weights. However, PNI can improve robustness only when noise injection is still open at the test time. Even so, it can also be easily attacked by adversarial examples generated by the average gradient of multiple times back propagation as will be shown in Section 4.1.

Instead, we propose an alternative method, denoted as Adversarial Weight Perturbation (AWP), that uses the worst-case in the small region around $f_w$. Intuitively, in order to own a flat loss landscape, the highest loss (i.e., worst-case) in the local region should be small enough. That is,

$$\min_w \max_{v \in \mathcal{N}(w)} \sum_{i=1}^{n} \phi_{x_i, y_i}(w + v), \quad (9)$$

where $\mathcal{N}(w)$ is the neighbour of current weight $w$. Recalling the adversarial loss $\phi_{x_i, y_i}$ in Eq. (2), Eq. (9) becomes:

$$\min_w \max_{v \in \mathcal{N}(w)} \sum_{i=1}^{n} \max_{\|x_i' - x_i\|_p \leq \epsilon} \ell(f_{w+v}(x_i'), y_i). \quad (10)$$

Note that the maximization on $v$, similar to the minimization on $w$, depends on the entire samples (at least the batch samples) to make whole loss (not the loss on each sample) maximal, thus the two maximizations are not exchangeable.

3.2. Size of Weight Perturbation

Following the weight perturbation direction, we need to determine how much perturbation should be injected. The perturbation cannot be too small which cannot effectively regularize the flatness of loss landscape, and it cannot be too large which will make DNNs hard to train3, i.e., reducing training accuracy and further affecting test robustness. Thus, the perturbation constraint should be properly selected.

Different from the constraint on adversarial inputs, in which the maximal perturbation is a fixed value $\epsilon$, we restrict the weight perturbation $v_l$ using its relative size to weights $w_l$:

$$\|v_l\| \leq \gamma \|w_l\|, \quad (11)$$

where $\gamma$ is the constraint on weight perturbation size, $w_l$ denotes the weight of $l$-th layer, and $\| \cdot \|$ represents size measurement (matrix norm) such as Frobenius norm.

The reasons for using relative size to constrain weight perturbation lie on two aspects: (1) the numeric distribution of weights is different from layer to layer, so it is impossible to constrain weights of different layers using a fixed value; and (2) there is scale invariance on weights, e.g., when non-linear ReLU is used, the network remains unchanged if we multiply the weights in one layer by 10, and divide by 10 at the next layer. Thus, we use a relative size constraint layer by layer. The influence of different size constraint $\gamma$ and size measurement will be analyzed in Section 4.1.

3.3. Incorporating AWP into Adversarial Training Framework

AWP can be further incorporated into adversarial training framework. To this end, we propose a new training strategy, denoted as AWP-Adv, which is an alternative to PNI. The objective function is then refined as:

$$\min_w \max_{v \in \mathcal{N}(w)} \sum_{i=1}^{n} \max_{\|x_i' - x_i\|_p \leq \epsilon} \ell(f_{w+v}(x_i'), y_i). \quad (10)$$

To achieve the goal of adversarial training, we use AWP-Adv to train DNNs. The new training strategy is as follows:

1. Initialize model $f_w$.
2. Compute adversarial examples $x'_i$.
3. Update model parameters $w$ by using the gradient of loss function $\ell(f_w(x'_i), y_i)$.
4. Add a weight perturbation $v_l$ using the relative size constraint $\gamma \|w_l\|$.
5. Repeat steps 2 to 4 until convergence.

In this way, the objective function is then refined as:

$$\min_w \max_{v \in \mathcal{N}(w)} \sum_{i=1}^{n} \max_{\|x_i' - x_i\|_p \leq \epsilon} \ell(f_{w+v}(x_i'), y_i). \quad (10)$$

Note that the maximization on $v$, similar to the minimization on $w$, depends on the entire samples (at least the batch samples) to make whole loss (not the loss on each sample) maximal, thus the two maximizations are not exchangeable.
3.3. AWP-based Adversarial Training Algorithm

Once the direction and size of weight perturbation are determined, we propose an algorithm to solve the adversarial weight perturbation and incorporate it into the adversarial training framework. For the two maximization problems in Eq. (10), we iteratively first generate adversarial example \( x'_i \), and then update weight perturbation \( \nu \), which is cycled for multiple iterations. Empirically, we adopt PGD to solve the two maximization problems just like Madry et al. (2018). We find that it works well in our experiments. With regard to some results on theoretical measurements or guarantees for the maximization problem like Wang et al. (2019), we leave it for further work.

In the following, we describe the procedure of AWP-based standard adversarial training, named Madry+. Pseudo-code is shown in Algorithm 1. Accordingly, we first craft adversarial examples \( x' \) using PGD attack on \( f_{w+v} \) (Line 6-11):

\[
x'_i = \arg \max_{\|x'_i-x_i\|_\infty \leq \epsilon} \ell(f_{w+v}(x'_i), y_i),
\]

where \( \nu \) is 0 for the first iteration. Then we calculate adversarial weight perturbation based on the generated adversarial examples \( x' \) (Line 12-14):

\[
\nu \leftarrow \Pi_{\gamma} \left( \nu + \eta_2 \frac{\nabla \nu}{\| \nabla \nu \|_\infty} \sum_{i=1}^{m} \ell(f_{w+v}(x'_i), y_i) \right),
\]

where \( m \) is the batch size. Similar to generating adversarial examples \( x' \) via FGSM (one-step) or PGD (a multi-step version of FGSM), there are also one-step or multi-step methods to solve \( \nu \). Following that, we alternately generate \( x' \) and calculate \( \nu \) for a number of iterations \( A \) (Line 5-15). As shortly will be shown in Section 4.1, one iteration for \( A \) and one-step for \( \nu \) are enough to get good robustness improvements. Finally, we update the parameters of the perturbed DNN \( f_{w+v} \) using SGD. Note that after optimizing the loss of a perturbed point on the landscape, we should come back to the center point \( w \) again for the next start. Thus, the actual parameter update follows (Line 16):

\[
w \leftarrow (w + \nu) - \eta_3 \nabla_{w+v} \frac{1}{m} \sum_{i=1}^{m} \ell(f_{w+v}(x'_i), y_i) - \nu.
\]

And then the AWP \( \nu \) and the DNN parameter \( w \) of TRADES are updated similarly following Eq. (13) and Eq. (14) respectively, where \( \ell(f_{w+v}(x'_i), y_i) \) is TRADES-specific as \( CE(f_{w+v}(x_{i}), y_i) + \beta \cdot KL(f_{w+v}(x_{i}) || f_{w+v}(x'_i)) \).

Similarly, we can get corresponding AWP-based methods for MART and RST.

4. Experiments

In this section, we first conduct a series of experiments to have a deep understanding of our proposed AWP, and then empirically evaluate its effectiveness on benchmark datasets against both white-box and black-box attacks.

4.1. Empirical Understanding of the Proposed AWP

For a comprehensive understanding of AWP, we first evaluate the influence of different optimization strategies in AWP. Then, we delve into AWP to investigate it from the following 4 perspectives: (1) weight perturbation direction; (2) weight perturbation size; (3) flatness of adversarial loss landscape; and (4) adversarially robust generalization gap.

Experimental Setup. We train ResNet-18 on CIFAR-10 using standard adversarial training Madry and AWP-based one Madry+. All the models are trained using SGD with momentum 0.9, weight decay \( 2 \times 10^{-4} \) and an initial learning rate of 0.1, which is divided by 10 at the 75-th and 90-th epoch (100 epochs in total). Simple data augmentations such as 4-pixel padding with \( 32 \times 32 \) random crop and random horizontal flip are applied. The maximal input

| Algorithm 1 AWP-based Standard Adversarial Training |
|-----------------------------------------------|
| 1: Input: Network \( f_w \), training dataset \( S \), batch size \( m \), |
| learning rate \( \eta_1 \), total epochs \( T \), PGD step size \( \eta_1 \), |
| PGD steps \( K_1 \), AWP constraint \( \gamma \), AWP step size \( \eta_2 \), AWP |
| steps \( K_2 \), alternate iteration \( A \). |
| 2: Output: Robust model \( f_w \). |
| 3: for \( t = 0 \) to \( T - 1 \) do |
| 4: Load mini-batch \( B = \{(x_1, y_1), \ldots, (x_m, y_m)\} \) |
| 5: for \( a = 0 \) to \( A - 1 \) do |
| 6: for \( i = 1, \ldots, m \) (in parallel) do |
| 7: \( x'_i \leftarrow x_i + \epsilon \delta, \) where \( \epsilon \sim \text{Uniform}(-1, 1) \) |
| 8: for \( k = 1, \ldots, K_1 \) do |
| 9: \( x'_i \leftarrow \Pi_{\gamma} \left( x'_i + \eta_1 \text{sign}(\nabla_{x'_i} \ell(f_{w+v}(x'_i), y_i)) \right) \) |
| 10: end for |
| 11: end for |
| 12: for \( k = 1, \ldots, K_2 \) do |
| 13: \( \nu \leftarrow \Pi_{\gamma} \left( \nu + \eta_2 \frac{\nabla \nu}{\| \nabla \nu \|_\infty} \sum_{i=1}^{m} \ell(f_{w+v}(x'_i), y_i) \right) \) |
| 14: end for |
| 15: end for |
| 16: \( w \leftarrow (w + \nu) - \eta_3 \nabla_{w+v} \frac{1}{m} \sum_{i=1}^{m} \ell(f_{w+v}(x'_i), y_i) - \nu \) |
| 17: end for |
perturbation $\epsilon = 8/255$ (image pixels are normalized into $[0, 1]$) and the maximal weight perturbation $\gamma = 5 \times 10^{-3}$ unless otherwise specified. The training attack is PGD-10 with random start and step size $0.007$, while the test attack is PGD-20 with random start and step size $0.01$.

**Analysis on Optimization Strategy.** According to Algorithm 1, we study the influence of different alternation iteration $A$ and step number $K_2$ in solving adversarial weight perturbation $\nu$. However, for step number $K_1$ in generating $x'$, previous work has showed that PGD-10 based adversarial training usually obtains better robustness than FGSM based one (Madry et al., 2018; Wang et al., 2019), so its default setting is $K_1 = 10$. For alternation iteration $A$, we test Madry+ using $A \in \{1, 2, 3\}$ while keeping $K_2 = 1$. As shown by the orange bars in Figure 2(a), we can see that one iteration ($A = 1$) already achieves 51.19% robust accuracy on test set, and extra iterations only bring few improvements but with much overhead. For step number $K_2$ in solving $\nu$, we assess Madry+ with $K_2 \in \{1, 5, 10\}$ while keeping $A = 1$. The green bars in Figure 2(a) show that various $K_2$ achieves almost the same robust accuracy on test set, which is different from the influence of $K_1$ in generating $x'$ (FGSM and PGD-10 has significant difference). Based on these results, in practice and the following experiments, we implement AWP with $A = 1$, $K_1 = 10$, $K_2 = 1$ and training time overhead is $\frac{\text{time}_{\text{original}}}{\text{time}_{\text{AWP}}} \approx 6\%$.

**Comparison of AWP and Random Weight Perturbation.** Here, we empirically compare the proposed AWP with random weight perturbation. Recalling the statement in Section 3.1 that AWP is able to find the worst-case in a small region, we show the adversarial loss of pre-trained Madry and its adversarially perturbed (calculated by Eq. (13)) and randomly perturbed (sampled from a multivariate Gaussian distribution) versions in Figure 2(b). Note that adversarial loss is the output of the maximization problem in Eq. (2) without the following minimization process, which is different from the commonly used training loss (the output of the minimization problem in Eq. (1)). In Figure 2(b), the adversarial loss of the randomly perturbed model (orange curve) is similar to that of pre-trained unperturbed Madry model (blue curve) throughout the training, while the adversarially perturbed model (green curve) has much higher adversarial loss value than others, especially at the late stage of training. It indicates that, on the adversarial loss landscape, AWP indeed can find the worst-case point in a small region of the original model even with only one single step perturbation in solving $\nu$.

Next, following the procedure of Madry+, we only replace AWP by random weight perturbation (RWP, first generating adversarial example, then random weight perturbation), and get the robust accuracy of 46.50% in Table 1. Compared with Madry, we can see that random weight perturbation has a negligible improvement on robust accuracy while Madry+ achieves a large gain. This indicates the superiority of AWP over random weight perturbation. Finally, we show a special random weight perturbation method PNI (first injecting adversarial example, and PNI is still enabled at the test time). Due to the existence of randomness when testing, we evaluate PNI on adversarial examples generated by the average gradient of multiple times back propagation following (Athalye et al., 2018). Table 1 shows that PNI fails to be robust under this average attack while other three methods are almost unaffected (no weight randomness in them when testing), which verifies the effectiveness of AWP again.

**Analysis on Weight Perturbation Size.** We explore the effect of weight perturbation size on the final robustness from two aspects: size constraint $\gamma$ and size measurement (norm). The robust accuracy with varying $\gamma$ on Madry+ and TRADES+ is shown in Figure 3(a), we can see that both methods can achieve notable robustness improvements in a certain range $\gamma \in [1 \times 10^{-3}, 5 \times 10^{-3}]$. As analyzed in Section 3.2, $\gamma$ cannot be too small or too large (either will influence the performance). However, once $\gamma$ is properly selected, it has a relatively good transferability across models (improvements of Madry+ and TRADES+ have an overlap on $\gamma$, though their highest points are not the same).

Moreover, recalling the size constraint is deployed through the specific size measurement/norm, we assess two matrix norms: $L_1$ and $L_2$ (also called Frobenius norm $L_F$) in

---

We additionally test FGSM ($K_1 = 1$) for generating $x'$ while keeping $A$, $K_2 = 1$, its robust accuracy is 43.68%. On one hand, it shows the big effect of $K_1$ on robustness (51.19% for $K_1 = 10$), on the other hand, it demonstrates the great potential of AWP (AWP-based FGSM adversarial training is close to PGD-10 adversarial training (Madry) with 46.36% robust accuracy.)
Figure 3(b). It demonstrates that different norms almost perform the same, i.e., both norms have similar robustness improvements and similar trends with varying $\gamma$.

**Effect on Flatness of Loss Landscape.** Following Figure 1(a), we visualize the adversarial loss landscape of Madry+ with different $\gamma$ in Figure 3(c). The black dashed line of $\gamma = 0$ is the baseline Madry, i.e., without weight perturbation. As $\gamma$ grows, the regularization on the flatness of loss landscape becomes stronger, that is, the landscape becomes flatter. Revisiting the red line in Figure 3(a), the highest robustness is obtained at $\gamma = 1 \times 10^{-2}$, while the flattest landscape is attained at a larger $\gamma = 2 \times 10^{-2}$ in Figure 3(c). This implies that flatter loss landscape does not mean a certain higher adversarial robustness, because larger $\gamma$ indeed introduce stronger regularization, but simultaneously it will result in the difficulty on training DNNs, i.e., reducing robust training accuracy and further affecting test robustness, as shortly will be shown in the following part.

**Effect on Robust Generalization Gap.** To understand the mechanism of flat loss landscape, following Figure 1(b), we illustrate the robust accuracy on adversarial examples of Madry+ with different $\gamma$ in Figure 3(d) where the entire column represents the robust training accuracy whose bottom and top is the robust testing accuracy (blue bar) and robust generalization gap (orange bar) respectively. When we enlarge $\gamma$, the robust generalization gap decreases significantly. Revisiting the corresponding landscape in Figure 3(c), it verifies that a flatter adversarial loss landscape is well-correlated with a smaller generalization gap on robust accuracy. However, a larger $\gamma = 2 \times 10^{-2}$ decreases the robust accuracy on training set dramatically. This is because when injected large weight perturbations into the model, making the loss remain small in a larger range is hard to optimize. This also well explains the phenomenon that $\gamma = 2 \times 10^{-2}$ has the flattest landscape and smallest robust generalization gap but does not own the highest robust accuracy, which is caused by the training accuracy drop.

Therefore, when talked about the robustness on test set, we need to think of it from both the robust generalization gap and the robust accuracy on training set. Satisfactory robust test accuracy can be obtained only if the robust generalization gap becomes smaller at the cost of an acceptable decrease of robustness on training set.

### 4.2. Adversarial Robustness Evaluation

In this part, we evaluate the effectiveness of our proposed AWP on MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009) to benchmark the state-of-the-art robustness against white-box and black-box attacks.

**Baselines.** (1) Madry (Madry et al., 2018); (2) TRADES (Zhang et al., 2019); (3) MART (Wang et al., 2020); and (4) RST (one type of SSL\(^3\)) (Carmon et al., 2019).

**Training Settings.** For MNIST, all defense models are built on a 4-layer CNN and trained using SGD with momentum 0.9. The initial learning rate is 0.01 and divided by 10 at the 20-th and 40-th epoch (50 epochs in total). The maximum input perturbation $\epsilon = 0.3$. For CIFAR-10, the settings are almost the same as Section 4.1, except that we use WideResNet-34-10 (depth 34 and width 10) here. For AWP, we set $\gamma = 5 \times 10^{-3}$. Other hyperparameters of the baselines are configured as per their original papers.

**White-box Robustness.** First, we evaluate the robustness of defense models against white-box attacks, in which the architecture and parameters are accessible by an adversary. For MNIST, three types of attack are used: FGSM, PGD-20 (20-step PGD with random start and step size $\epsilon/10$) and $CW_{\infty}$ ($L_\infty$ version of CW loss optimized by PGD-100). For CIFAR-10, in addition to above 3 attacks, we test two more attacks: PGD-100 (100-step PGD with random start and step size $\epsilon/10$) and Feature Attack (FA) (20-step with step size 1/255 and 100 restart). Table 2 shows the robust accuracy of all defense models, where “Natural” denotes the accuracy on natural test images. Our proposed AWP\(^4\) consistently improve the robustness of current state-of-the-art methods against all types of attacks on both MNIST and CIFAR-10. AWP has larger improvements especially on

---

\(^3\)Four concurrent SSL methods have similar robustness.

\(^4\)If $\gamma$ is tuned specifically for each method and dataset, we believe the improvements will be larger.
Table 2. White-box robust accuracy (%) on MNIST and CIFAR-10. We test four adversarial training methods and their AWP-based counterpart marked by “+”. RST does not have results on MNIST due to the lack of unlabeled data.

| Defense   | MNIST          | CIFAR-10       |
|-----------|----------------|----------------|
|           | Natural | FGSM | PGD-20 | CW\(_\infty\) | Natural | FGSM | PGD-20 | PGD-100 | CW\(_\infty\) | FA |
| RST       | –       | –    | –      | –         | 89.69   | 67.83 | 63.10  | 62.30    | 60.47   | 63.10 |
| RST+      | –       | –    | –      | –         | 87.61   | 88.77 | 87.58  | 88.06    | 89.00   |       |

Table 3. Black-box robust accuracy (%) against black-box test attacks on MNIST and CIFAR-10. We test four adversarial training methods and their AWP-based counterpart marked by “+”. RST does not have results on MNIST due to the lack of unlabeled data.

| Defense   | MNIST          | CIFAR-10       |
|-----------|----------------|----------------|
|           | FGSM | PGD-20 | PGD-100 | CW\(_\infty\) | FGSM | PGD-20 | PGD-100 | CW\(_\infty\) |
| RST       | –       | –    | –      | –         | 86.51   | 87.62 | 87.58  | 88.06    |
| RST+      | –       | –    | –      | –         | 87.61   | 88.77 | 88.70  | 89.00    |

CIFAR-10. This is, in our belief, because the pattern inside CIFAR-10 is more complex and harder to generalize on robustness, and AWP aims at achieving a flatter landscape that improves the robust generalization. We also evaluate on Feature Attack (FA) (Lin, 2019) which generates adversarial examples through narrowing the distance in feature space, and succeeds in breaking some recently proposed defenses that work well under PGD-20 (Zhang & Wang, 2019). The baselines (Madry/TRADES/MART/RST) in our experiments are still robust to FA, which indicates their performance is still trustful. The improvement brought by AWP is also unaffected under FA, showing the effectiveness and trustworthiness of AWP. Besides, we conduct an additional check using a gradient-free attack SPSA (Uesato et al., 2018). It does not achieve lower robust accuracy than above gradient-based attacks, which confirms that the robust improvement of AWP is not due to gradient masking.

**Black-box Robustness.** We also test the robustness under black-box attacks, in which attacks are crafted from natural test images by attacking a surrogate model. The surrogate model is a copy of the defense model trained independently on MNIST, and a normally trained WideResNet-28-10 (natural accuracy 96.1% on test set) on CIFAR-10. The attacking methods used here are: FGSM, PGD\(_{20}\), PGD\(_{100}\), and CW\(_\infty\). The black-box robustness of all defense models are reported in Table 3. Again, the robustness improved by AWP is consistent amongst different adversarial training variants. We also find that robustness on strong attacks like CW\(_\infty\) is higher than weak attacks like FGSM, which indicates that strong attacks have less transferability than weak attacks (Madry et al., 2018). Besides, compared with the white-box results in Table 2, all defense methods achieve much better robustness against black-box attacks.

5. Conclusion

In this paper, we first revealed that the adversarially robust generalization gap is closely related to the flatness of the adversarial loss landscape, and empirically demonstrate that several well-recognized adversarial training variants all introduce a flatter adversarial loss landscape though they used different techniques to improve adversarial robustness. Based on this finding, we proposed **Adversarial Weight Perturbation (AWP)** to directly make the landscape flat, and developed an effective and efficient algorithm that can be easily incorporated into different adversarial training variants. On benchmark datasets, our proposed AWP consistently improves the adversarial robustness of different adversarial training methods by a notable margin.
References

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Bai, Y., Feng, Y., Wang, Y., Dai, T., Xia, S.-T., and Jiang, Y. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *NeurIPS*, 2017.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *S&P*, 2017.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017.

Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Li, S., Chen, L., Kounavis, M. E., and Chau, D. H. Compression to the rescue: Defending from adversarial attacks across modalities. In *KDD*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. In *ICLR*, 2018.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *ICML*, 2017.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Gu, S. and Rigazio, L. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Guo, C., Rana, M., Cisse, M., and Van Der Maaten, L. Countering adversarial images using input transformations. In *ICLR*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, Z., Rakin, A. S., and Fan, D. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *CVPR*, 2019.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

Khim, J. and Loh, P.-L. Adversarial risk bounds for binary classification via function transformation. *arXiv preprint arXiv:1810.09519*, 2, 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

Lin, D. Feature attack. *https://github.com/Line290/FeatureAttack*, 2019.

Liu, Q., Liu, T., Liu, Z., Wang, Y., Jin, Y., and Wen, W. Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In *ASP-DAC*, 2018.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICML*, 2018.
Revisiting Loss Landscape for Adversarial Robustness

Montasser, O., Hanneke, S., and Srebro, N. Vc classes are adversarially robustly learnable, but only improperly. In COLT, 2019.

Najafi, A., Maeda, S.-i., Koyama, M., and Miyato, T. Robustness to adversarial perturbations in learning from incomplete data. In NeurIPS, 2019.

Nakkiran, P. Adversarial robustness may be at odds with simplicity. arXiv preprint arXiv:1901.00532, 2019.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. arXiv preprint arXiv:1412.6614, 2014.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In EuroS&P, 2016a.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In S&P, 2016b.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In Asia CCS, 2017.

Prabhu, V. U., Yap, D. A., Xu, J., and Whaley, J. Understanding adversarial robustness through loss landscape geometries. arXiv preprint arXiv:1907.09061, 2019.

Samangouei, P., Kabkab, M., and Chellappa, R. Defensegan: Protecting classifiers against adversarial attacks using generative models. In ICLR, 2018.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In NeurIPS, 2018.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In ICLR, 2018.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In ICLR, 2019.

Uesato, J., ODonoghue, B., Kohli, P., and Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In ICML, 2018.

Uesato, J., Alayrac, J.-B., Huang, P.-S., Stanforth, R., Fawzi, A., and Kohli, P. Are labels required for improving adversarial robustness? In NeurIPS, 2019.

Wang, Y., Deng, X., Pu, S., and Huang, Z. Residual convolutional ctc networks for automatic speech recognition. arXiv preprint arXiv:1702.07793, 2017.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. On the convergence and robustness of adversarial training. In ICML, 2019.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In ICLR, 2020.

Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017.

Yin, D., Kannan, R., and Bartlett, P. Rademacher complexity for adversarially robust generalization. In ICML, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.

Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J., and Wang, L. Adversarially robust generalization just requires more unlabeled data. arXiv preprint arXiv:1906.00555, 2019.

Zhang, H. and Wang, J. Defense against adversarial attacks using feature scattering-based adversarial training. In NeurIPS, 2019.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In ICML, 2019.