# Improved Adversarial Robustness via 1-Lipschitz Function

Haoxu Huang [1] Jiraphon Yenphraphai [2] Haozhen Bo [3]

## Abstract

In this work, we present a novel way to defend adversarial attack. First, we impose hard 1-Lipschitz constraint on a neural network by using spectral normalization. Second, we encourage the consistency of adversarial pairs' logit based on H-divergence. Our method is evaluated on AutoAttack and achieves $59.50\%$ robust accuracy in standard mode of AutoAttack with 1000 samples without using any extra or synthetic data during training. Our final code is available at https://github.com/NoTody/Adversarial-Robustness-via-Enforcing-1-Lipschitz-Function. To check commit history, check our development repository at https://github.com/domejiraphon/FML_project with corresponding branches.

## 1. Introduction

Adversarial attack poses a threat to deep learning models. Neural network models confidently misclassify the samples perturbed by small specific noises which are indistinguishable by human eyes even if they are well-trained. In the context of image classification, an adversarial image and its original image look similar to human eye. Surprisingly, the neural network produces the logits which are completely different. It is crucial to develop models which are robust to this attack. Thus, we aim to enforce a neural net with 1-Lipschitz functions introduced in variants of Generative adversarial network(GANs) (Goodfellow et al., 2014a).

GANs is an image generation model where it learns to replicate the underlying distribution of a given dataset and be able to generate new samples from the learned distribution. It is consisted of a generator which tries to learn a distribution and generates data samples and a discriminator which tries to discriminate real samples and generated samples. As it is like a two-person game, it is notorious for training instability. Originally, the goal of lipschitz constraint is enforced in GANs' discriminator to mitigate training instability. WGAN (Arjovsky et al., 2017) proposes to use earth moving distance as an objective function and enforces a model to be 1-Lipschitz functions by clamping the critic's parameters. Several techniques have been proposed to constraint 1-Lipschitz constraint. Sampling-based methods (Gulrajani et al., 2017) (Terjék, 2019) (Petzka et al., 2017) introduce Lipschitz regularization term in the objective function. These approaches use a soft constraint and rely heavily on sampled data. However, the state-of-the-art in imposing a hard Lipschitz constraint computes gradient for every step and uses it to normalize the logit (Wu et al., 2021). To adapt their method to our work, Jacobian matrix of the logit with respect to the input images is required for every step; thus, it is painfully slow. In this work, we choose to impose this hard 1-Lipschitz constraint on a neural network using spectral normalization (Miyato et al., 2018).

To impose 1-Lipschitz function on a layer of neural network, spectral normalization controls the spectral norm of the weights $W$ to be at most 1. The largest singular value of $W$ is used to normalize weight. As such, controlling the spectral norm of every neural network layer restricts 1-Lipschitz function on a neural network.

To boost a model's robustness, we want the model to assign same class to a image and its corresponding adversarial pair. This can be encouraged by cross-entropy loss. Yet, the model still doesn't know that this pair is the same images and should assign the same classification probabilities. That is, our objective function should incorporate this prior knowledge. We choose to do so by introducing consistency loss of H-divergence between a clean image and its adversarial pair.

(Wu et al., 2020) and (Zheng et al., 2021) investigate the flatness of loss landscape and find that it correlates with the robustness to adversarial attack. They propose an explicit way to flatten the weight loss landscape by adding the worst-case weight perturbation. Even though we don't explicitly flatten the weight loss landscape, both restricting 1-Lipschitz function and enforcing adversarial logit pair can be seen as implicit approaches to smooth the loss landscape. We investigate the loss landscape's flatness of the our proposed method and compare against the state-of-the art in flattening loss landscape via adversarial training (Wu et al., 2020).

We evaluate our approaches on AutoAttack (Croce & Hein,

[1] hh2740 [2] jy3694 [3] hb2432. Correspondence to: Haozhen Bo <hb2432@nyu.edu>.

2020b) using CIFAR-10 (Krizhevsky et al., 2009). Our contributions that differs from our starting point (Rebuffi et al., 2021) are as follows:

1. We adapt spectral normalization previously used in GANs to prevent adversarial attack.

2. We introduce consistency loss of adversarial pairs based on H-divergence.

3. We apply model ensemble and found that it does help on improving the adversarial robustness.

## 2. Related Works

**Adversarial Defense** Fast Gradient Sign Method (Goodfellow et al., 2014b) proposes a min-max objective function where an inner loops finds an adversarial attack with one-step scheme and an outer loop minimizes the classification objective. (Madry et al., 2017) argues that a model can be more robust with multi-step scheme and introduces projected gradient descent. (Wu et al., 2020) and (Zheng et al., 2021) view model's robustness from loss landscape perspective where flat minimas encourage robustness and design method to find solution landing in a wide loss basin. (Rebuffi et al., 2021) mitigates adversarial overfitting through data augmentation.

**1-Lipschitz function** WGAN (Arjovsky et al., 2017) enforces 1-Lipschitz function by clipping a critic's weights. (Gulrajani et al., 2017), (Petzka et al., 2017), (Terjék, 2019) introduce a soft constraint using the regularization term penalizing the norm of weights' gradients. On the other hand, a hard constraint to impose this 1-Lipschitz has been proposed by using spectral normalization (Miyato et al., 2018). (Wu et al., 2021) normalizes a function with its gradient and the absolute values of the logit.

**Adversarial logit pair regularization** Since the logit of images and its adversarial pairs should be similar, (Kannan et al., 2018) enforces it by introducing $l_2$ logit pair penalty as their regularization. TRADES (Zhang et al., 2019) uses KL-divergence. (Wang et al., 2019) masks this loss with the misclassification instances.

## 3. Methodology

Our goal is to defense adversarial attack. To solve this, we propose to incorporate 1-Lipschitz function on a neural network via theoretical proofs in Sections 3.1. We then use consistency regularization to enforce the logit similarity between adversarial pairs, described in Sections 3.2. In Sections 3.3, we briefly review stochastic weight averaging (SWA) (Izmailov et al., 2018) and why we use it. Finally, we explain how we do model ensemble in Sections 3.4.

### 3.1. Spectral Normalization (SN)

Spectral normalization has been proposed to stabilize the training of discriminator networks in GANs (Miyato et al., 2018). Consider a deep neural network, with input $\boldsymbol{x}$, represented as

$$f(\boldsymbol{x}, \theta) = W_{L+1}(a_L(W_L(...a_1(W_1\boldsymbol{x})...))), \tag{1}$$

where $\theta := \{W_1, ..., W_{L+1}\}$ is the set of parameters of $L+1$ layers, $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$,$W_{L+1} \in \mathbb{R}^{1\times d_L}$, and $a_l$ is the element-wise non-linear activation function. For each layer $g : \boldsymbol{h}_{in} \mapsto \boldsymbol{h}_{out}$, we would like to control the Lipschitz constant of the mapping $g$.

**Lemma 3.1.** *For linear mapping $g = W\boldsymbol{x}$, the Lipschitz norm of $g$ is $||g||_{Lip} = \sup_{\boldsymbol{x}} \sigma(\nabla g(\boldsymbol{x})) = \sigma(W)$.*

*Proof.* Suppose $g$ is $K$-Lipschitz, then

$$\begin{aligned} &||W\boldsymbol{x}|| \leq K||\boldsymbol{x}|| \\ \Longleftrightarrow &\langle W\boldsymbol{x}, W\boldsymbol{x}\rangle \leq K^2\langle x, x\rangle \\ \Longleftrightarrow &\langle (A^TA - K^2)\boldsymbol{x}, \boldsymbol{x}\rangle \leq 0. \end{aligned}$$

Expand $\boldsymbol{x}$ by a orthonormal basis of eigenvectors $\{\boldsymbol{v}_i\}$ of $A^TA$, then

$$\begin{aligned} \langle (A^TA - K^2)\boldsymbol{x}, \boldsymbol{x}\rangle &= \langle (A^TA - K^2)\sum_i x_i\boldsymbol{v}_i, \sum_j x_j\boldsymbol{v}_j\rangle \\ &= \sum_{i,j} x_ix_j\langle (A^TA - K^2)\boldsymbol{v}_i, \boldsymbol{v}_j\rangle \\ &= \sum_i (\lambda_i - K^2)x_i^2 \leq 0 \\ &\Longrightarrow \sum_i (K^2 - \lambda_i)x_i^2 \geq 0. \end{aligned}$$

Since $A^TA$ is positive semi-definite, all $\lambda_i \geq 0$, we must have

$$K^2 \geq \lambda_i, \forall i.$$

Since $||g||_{Lip}$ is the minimum $K$ that satisfies the above condition, we have

$$||g||_{Lip} = \sigma(W),$$

which is the largest singular value of $W$, or the spectral norm of $W$. □

If the weights $W$ in a layer $g$ are normalized by the largest singular value of itself, $||W_{\text{SN}}||_{Lip} \leq 1$.

$$W_{\text{SN}} = W/\sigma(W)$$

That is, $||g||_{Lip} \leq 1$ when normalized by its spectral norm.

**Lemma 3.2.** *Suppose* $g_1 \colon \mathbb{R}^n \to \mathbb{R}^m, g_2 \colon \mathbb{R}^m \to \mathbb{R}^l$, $g_2(g_1(x)) = W_2 W_1 \boldsymbol{x}$, *then*

$$||g_2 \circ g_1||_{Lip} \leq ||g_2||_{Lip} \cdot ||g_1||_{Lip}.$$

*Proof.* By chain rule, definition of spectral norm, and convexity of supreme,

$$\begin{aligned} ||g_2 \circ g_1||_{Lip} &= \sup_{\boldsymbol{x}} \sigma(\nabla(g_2 \circ g_1)(\boldsymbol{x})) \\ &= \sigma(\nabla g_2(g_1(\boldsymbol{x}))\nabla g_1(\boldsymbol{x})) \\ &= \sup_{||\boldsymbol{v}|| \leq 1} ||[\nabla g_2(g_1(\boldsymbol{x}))] \cdot [\nabla g_1(\boldsymbol{x})]\boldsymbol{v}|| \\ &\leq \sup_{||\boldsymbol{u}|| \leq 1} ||\nabla g_2(g_1(\boldsymbol{x}))\boldsymbol{u}|| \sup_{||\boldsymbol{v}|| \leq 1} ||\nabla g_1(\boldsymbol{x})\boldsymbol{v}|| \\ &\leq ||g_2||_{Lip} \cdot ||g_1||_{Lip}. \end{aligned}$$

□

**Theorem 3.3.** *Let $f$ be defined as in equation (1), suppose the activation functions $a_l$ are 1-Lipschitz, then*

$$||f||_{Lip} \leq \prod_{l=1}^{L+1} \sigma(W_l).$$

*Proof.* By Lemma 3.1 and 3.2, we have

$$\begin{aligned} ||f||_{Lip} &\leq ||W_{L+1}||_{Lip} \cdot ||a_L||_{Lip} \cdot ...||W_1||_{Lip} \\ &= \prod_{l=1}^{L+1} ||W_l||_{Lip} = \prod_{l=1}^{L+1} \sigma(W_l). \end{aligned}$$

□

By normalizing the weights $W_l$ of each layer by its spectral norm, the resulted mapping $\hat{f}$ has $||\hat{f}||_{Lip} \leq 1$ by the above Theorem.

Consider $\mathcal{F}_{nn} = \{f_{\boldsymbol{w}} \colon \boldsymbol{w} \in \boldsymbol{W}\}$, the class of d hidden-layer neural networks with h units per hidden-layer and 1-Lpischitz activation function $a_l$ satisfying $a_l(0) = 0$. Follow the PAC-Bayes framework, the following theorem in (Farnia et al., 2018) provides a margin-based adversarial generalization bound for spectral normalized networks under projected gradient descent (PGD) attack. We refer to this theorem as a theoretic support for the use of spectral normalization.

**Theorem 3.4.** *Suppose that (1) $\mathcal{X}$ is norm-bounded as $||\boldsymbol{x}||_2 \leq B, \forall \boldsymbol{x} \in \mathcal{X}$; (2) $||f||_2 \leq 1, \forall f \in \mathcal{F}_{nn}$; (3) the loss $\ell$ and its first-order derivative are 1-Lipschitz. Consider PGD attack with noise level $\epsilon$, $r$ iterations, and stepsize $\alpha$. Then, for any $\eta, \gamma > 0$ with probability $1 - \eta$, the following bound applies to the PGD margin loss of any $f \in \mathcal{F}_{nn}$*

$$\begin{aligned} &L(f) \leq \hat{L}_\gamma(f) + \\ &\mathcal{O}(\sqrt{\frac{(B+\epsilon)^2 d^2 h \log(dh)\Phi(f) + d \log\frac{rdn \log M}{\eta}}{\gamma^2 n}}), \end{aligned} \tag{2}$$

*where*

$$\begin{aligned} \Phi(f) &= \Phi_{\epsilon,\kappa,r,\alpha}(f) \\ &:= \Big\{ \prod_{i=1}^{d} ||\boldsymbol{W}_i||_2 \big(1 + \frac{\alpha}{\kappa} \frac{1 - (\frac{2\alpha}{\kappa})^r \overline{Lip}}{1 - (\frac{2\alpha}{\kappa})\overline{Lip}^r} (\prod_{i=1}^{d} ||\boldsymbol{W}_i||_2) \\ &\quad (\sum_{i=1}^{d} \prod_{j=1}^{i} ||\boldsymbol{W}_j||_2)) \Big\}^2 \sum_{i=1}^{d} \frac{||\boldsymbol{W}_i||_F^2}{||\boldsymbol{W}_i||_2^2}, \end{aligned} \tag{3}$$

*and*

$$\overline{Lip} = \overline{Lip}(\nabla \ell \circ f) := (\prod_{i=1}^{d} ||\boldsymbol{W}_i||_2) \sum_{i=1}^{d} \prod_{j=1}^{i} ||\boldsymbol{W}_j||_2 \tag{4}$$

*is an upper-bound on the Lipschitz constant of the gradient $\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y)$.*

### 3.2. Consistency Regularization with H-Divergence

The consistency regularization (Tack et al., 2021) is our baseline evaluation. Our proposed method improves upon adding several changes on it. Consider augmentations $T_1, T_2 \sim \mathcal{T}$, temperature hyperparameter $\tau$, true label $y$, a deep neural network $f_\theta$ parameterized by $\theta$, an arbitrary divergence metric $\mathcal{D}$ and adversarial perturbation $\delta_i$, the consistency regularization can be represented as

$$\mathcal{D}(f(T_1(x) + \delta_1; \tau, \theta) \| f(T_2(x) + \delta_2; \tau, \theta)) \tag{5}$$

Then, the total loss is represented as

$$\begin{aligned} \mathcal{L}_{total} &:= \frac{1}{2} \sum_{i=1}^{2} \mathcal{L}_{adv}(T_i(x), y; \theta) + \\ &\lambda \mathcal{D}(f(T_1(x) + \delta_1; \tau, \theta) \| f(T_2(x) + \delta_2; \tau, \theta)) \end{aligned} \tag{6}$$

for hyperparameter $\lambda$ and supervised adversarial training loss

$$\mathcal{L}_{adv}(x, y; \theta) := \max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(x + \delta, y; \theta) \tag{7}$$

Consistency regularization works well for adversarial training because it enforces the distribution closeness for samples with added perturbation as an added regularization term on original adversarial training, which forces hard 1-Lipschitz constraint.

#### 3.2.1. H-Divergence

The divergence metric we choose is based on H-divergence (Zhao et al., 2022), which is a generalization of Jensen-Shannon divergence. In the most general form, the H-divergence is defined as:

**Definition 3.5.** *Let $\mathcal{P}$ be the set of probability distributions over $\mathcal{X}$ and $\mathcal{A}$ an action space, for two distributions $p, q \in \mathcal{P}(\mathcal{X})$, given any continuous function $\phi\colon \mathbb{R}^2 \longrightarrow \mathbb{R}$ such that $\phi(\theta, \lambda) > 0$ whenever $\theta + \lambda > 0$ and $\phi(0, 0) = 0$, define*

$$D_\ell^\phi(p||q) = \phi(H_\ell(\frac{p+q}{2}) - H_\ell(p), H_\ell(\frac{p+q}{2}) - H_\ell(q)), \tag{8}$$

*where $H_\ell(p) = \inf_{a\in\mathcal{A}} \mathbb{E}_p[\ell(X, a)]$ is the H-entropy, $\ell$ is a loss function $\ell : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$.*

**Proposition 3.6.** *For any choice of $\ell$ and $\phi$, the H-divergence $D_\ell^\phi(p||q)$ defined in equation (4) is a probability divergence.*

*Proof.* First, note that $H_\ell$ is concave since inf is a concave function. Hence,

$$\begin{aligned} &\theta = H_\ell(\frac{p+q}{2}) - H_\ell(p) \geq \frac{1}{2}(H_\ell(q) - H_\ell(p)), \\ &\text{and } \lambda = H_\ell(\frac{p+q}{2}) - H_\ell(q) \geq \frac{1}{2}(H_\ell(p) - H_\ell(q)) \\ &\Longrightarrow \theta + \lambda \geq 0 \\ &\Longrightarrow D_\ell^\phi(p||q) \geq 0, \text{ by requirement } \phi \text{ in definition.} \end{aligned}$$

In addition, let $p = q$, then $D_\ell^\phi(p||q) = \phi(0, 0) = 0$. □

The next proposition shows that Jensen-Shannon divergence is a special case of H-divergence.

**Proposition 3.7.** *Let $\mathcal{A} = \mathcal{P}(\mathcal{X})$, and $\ell(X, p) = -\log p(X)$ for $p \in \mathcal{A} = \mathcal{P}(\mathcal{X})$, then $H_\ell(p)$ is the Shannon Entropy. Furthermore, let $\phi(\theta, \lambda) = \frac{\theta+\lambda}{2}$, then the resulted H-divergence is the Jensen-Shannon divergence.*

*Proof.* By definition,

$$\begin{aligned} H_\ell(p) &= \inf_{a\in\mathcal{A}} \mathbb{E}_p[\ell(X, a)] = \inf_{p\in\mathcal{P}(\mathcal{X})} \mathbb{E}_p[-\log p(X)] \\ &= -\sum p(x) \log p(x) = H(X). \end{aligned}$$

Denote $m = \frac{p+q}{2}$, then,

$$\begin{aligned} D_\ell^\phi(p||q) &= H(m) - \frac{1}{2}(H(p) + H(q)) \\ &= \frac{1}{2}\sum p(x) \log \frac{p(x)}{m(x)} + \frac{1}{2}\sum q(x) \log \frac{q(x)}{m(x)} \\ &= D_{JS}(p||q). \end{aligned}$$

□

We choose $H_\ell$ to be shannon entropy and the divergence loss $\mathcal{D}$ in equation (5) to be two special cases of H-divergence (Zhao et al., 2022). Namely, H-Jenson Shannon represented as

$$D_\ell^{JS}(p, q) = H_\ell(\frac{p+q}{2}) - \frac{1}{2}(H_\ell(p) + H_\ell(q)) \tag{9}$$

and H-Min represented as

$$D_\ell^{Min}(p, q) = H_\ell(\frac{p+q}{2}) - \min(H_\ell(p), H_\ell(q)) \tag{10}$$

for $q, p \in \mathcal{P}(\mathcal{X})$ with $\mathcal{X}$ to be a finite set or a finite dimensional vector space and $H_\ell$ defined from Definition (3.5). for some action space $\mathcal{A}$ and any loss function $\ell : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$.

### 3.3. Stochastic Weight Averaging (SWA)

The goal of SWA is to find a SGD solution that centers at the loss basin as this solution is more robust to dataset shift. That is, a model becomes more generalized than a solution found by a standard training procedure. Rather than optimizing parameters with standard SGD training, SWA instead runs a SGD with cyclical learning rate or constant learning rate to traverse around a loss basin. The centered solution can be approximated by averaging the final $T$ epochs of the training schedule as follows:

$$\hat{\theta}_{\text{swa}} = \frac{1}{T}\sum_{i=1}^{T} \theta_i \tag{11}$$

where $\theta_i$ is a solution of SGD. According to the findings from (Wu et al., 2020) and (Zheng et al., 2021) that the flatness of the loss landscape corresponding to the improvement in adversarial robust accuracy, we incorporate SWA in our training schedule.

### 3.4. Model Ensemble

As we observed that models trained with different methods can be more robust against different attacks, we decide to do a small scale ensembling on the logits after the last linear layer of the models to leverage the disadvantages of different models. Namely, our simple ensembling can be represented as

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f_i(x + \delta, \theta_i), \tag{12}$$

for final ensembling output $\hat{f}$ and $N$ neural networks $f_i$.

To conclude, our model uses spectral normalization to enforce 1-Lipschitz function, consistency loss to limit adversarial pair's logit, SWA to find a generalized solution, and model ensemble to provide robustness against different attacks with the objective function as in equation (6).

# 4. Experiments

## 4.1. Implementation details

We train WRN-34-10 (Zagoruyko & Komodakis, 2016b) on CIFAR-10 (Krizhevsky et al., 2009) with SGD and multi-step scheduling separately using Xavier initialization, a learning rate of $0.1$, and batch size of 512. The checkpoint with lowest validation loss is saved for the final evaluation. For H-Jenson Shannon, two trains are performed, one is by multi-step learning rate decay at 80 epochs for total of 90 epochs training, anotheris by multi-step learning rate decay at 80 and 100 epochs for total of 120 epochs training. For H-Min, multi-step decay at 100 and 150 epochs for total of 200 epochs training. We set them this way because we observe that H-Jenson Shannon overfits fast after first learning rate decay but H-Min doesn't. The training is performed on 2 NVIDIA V100 GPUs.

The transformation space $\mathcal{T}$ is set to be the collection of AutoAugment (Cubuk et al., 2019), Random Crop, Random Horizontal Flip and Cutout (DeVries & Taylor, 2017) for both $T_1$ and $T_2$. Hyperparameter $\lambda$ in equation (6) set to 1 and the backbone neural network $f(\theta)$ we used for evaluation is WRN-34-10 (Zagoruyko & Komodakis, 2016a). The training perturbation $\delta$ in equation (5) is generated by 10-step projected gradient descent (PGD) with radius $\epsilon = 8/255$ and $\infty$-norm.

In our ensembling process, we choose models trained with H-Jenson Shannon and H-Min separately as we observed that models trained with H-Jenson Shannon is a lot more robust on APGD-CE (Croce & Hein, 2020b) attacks and model trained with H-Min is a lot more robust on APGD-T (Croce & Hein, 2020b) attacks separately.

All evaluations are performed on standard mode of AutoAttack (Croce & Hein, 2020b) with radius $\epsilon = 8/255$ and $\infty$-norm with the number of evaluation samples set to be 1000.

## 4.2. Results

The result we evaluated on WResNet 34-10 is shown in Table 1. For the meaning of each abbreviation we used in the table, please see the caption description. From the table, We can see that ensemble 1 gives best robust accuracy of $59.50\%$ among all methods, which indicates the feasibility of our proposed method. To our knowledge, this result is better than state-of-the-art (SOTA) method trained without syntactic or extra data. (The SOTA method proposed by Addepalli et al. (2021) achieves $58.07\%$ from RobustBench, which is lower than ours with $59.30\%$).

## 4.3. Adversarial Weight Perturbation (AWP)

We review (Wu et al., 2020) as we view our method as an approach to smooth the loss landscape, and we compare against it. (Wu et al., 2020) argues that weight and input loss landscapes affect the robust generalization gap. The flatter these loss landscapes are, the higher the adversarial robustness is. To regularize these landscapes, they proposes to use double worst-case perturbations on both weights and adversarial inputs.

## 4.4. Ablation Studies

We evaluate the effectiveness of our main contributions which are spectral normalization and consistency loss. For ablation studies, we use the same training schedule except that we train on WRN-28-4, rather than WRN-34-10, because of time constraint.

As shown in Table 2, we see the improvement in both clean and robust accuracy on all of the methods we use. The method that achieves the highest robust accuracy of $51.14$ and the lowest robust generalization gap of $34.92\%$ is Baseline+SWA+SN which is our proposed method. Note that, we include AWP in a table for a comparison in loss landscape as elaborated in the next section.

# 5. Discussion

## 5.1. Loss landscape

Flatness of the loss landscape can be defined as how much the loss changes when weights of a neural network are perturbed in some random directions (Huang et al., 2020). Suppose our optimizer find a solution which lies in a sharp basin. The number of misclassification will increase as the small amount of perturbation causes high changes in the loss value. That is, the classification boundary is wiggly. On the other hand, the solution lying in wide basin causes less changes in the loss function such that it is more generalized, when the test set's distribution is shifted, becoming more robust to the adversarial attack.

Adversarial training and consistency loss limits the logit differences between the adversarial pairs such that the input loss landscape is flatten. Spectral normalization also restricts 1-Lipschitz function on weight space of a neural network. We view our contributions, consistency loss and spectral normalization, as one of the methods that implicitly regularize the loss landscape. In order to visualize loss landscape, we measure the loss, as the optimized weights $\hat{W}$ in the network are increasingly perturbed in one specific direction. Given a perturbation factor $\gamma$, loss is calculated on the training data, with model parameters $(1 + \gamma)\hat{W}$. Fig. 1 compares the loss landscape between baseline and the methods using consistency loss, spectral normalization, and

*Table 1.* Result of our evaluation, where Baseline is our training on (Rebuffi et al., 2021) with H-JS (H-Jenson Shannon Divergence) and HMIN (H-Min Divergence) without synthetic/extra data, SN (Spectral normalization), AWP (Average weight perturbation), SWA (stochastic weight averaging), Ensemble 1 (H-JS+SWA+SN trained with 120 epochs with weight decay at 80 and 100 epochs and H-MIN+SWA+SN), Ensemble 2 (H-JS+SWA+SN H-JS+SWA+SN trained with 120 epochs with weight decay at 80 and 100 epochs, H-MIN+SWA+SN and H-JS+SWA+SN trained with 90 epochs with weight decay at 80 epochs) Metrics shown here are evaluated from AutoAttack: Clean Accuracy APGD-CE attack (Croce & Hein, 2020b), APGD-T attack (Croce & Hein, 2020b), FAB-T attack (Croce & Hein, 2020a), Square attack (Andriushchenko et al., 2020)

.

| Method | Initial Accuracy ↑ | APGD-CE ↑ | APGD-T ↑ | FAB-T ↑ | SQUARE ↑ | Robust Accuracy ↑ |
|---|---|---|---|---|---|---|
| H-JS | 88.40 % | 60.60 % | 55.00 % | 55.00% | 55.00 % | 55.00 % |
| H-MIN | 87.10 % | 56.90 % | 55.40 % | 55.30 % | 55.30 % | 55.30 % |
| H-JS+SWA+SN | 87.50 % | 58.60 % | 54.00 % | 54.00 % | 54.00 % | 54.00 % |
| H-MIN+SWA+SN | 88.50 % | 57.90 % | 56.70 % | 56.70 % | 56.70 % | 56.70 % |
| Ensemble 1 | 88.20 % | 61.40 % | 59.30 % | 59.30 % | 59.30 % | 59.30 % |
| Ensemble 2 | 87.90 % | **63.20** % | **59.50** % | **59.50** % | **59.50** % | **59.50** % |

*Table 2.* An ablation study of our model, where Baseline is our training on (Rebuffi et al., 2021) without extra data, SN (Spectral normalization), AWP (Average weight perturbation), SWA (stochastic weight averaging), CON (consistency loss). Metrics shown here are evaluated from AutoAttack: Clean Accuracy APGD-CE attack (Croce & Hein, 2020b), APGD-T attack (Croce & Hein, 2020b), FAB-T attack (Croce & Hein, 2020a), Square attack (Andriushchenko et al., 2020)

| Method | Initial Accuracy ↑ | APGD-CE ↑ | APGD-T ↑ | FAB-T ↑ | SQUARE ↑ | Robust Accuracy ↑ |
|---|---|---|---|---|---|---|
| Baseline (Rebuffi et al., 2021) | 85.56 % | 54.63 % | 50.39 % | 50.38% | 50.38 % | 50.38 % |
| Baseline+CON | 86.04 % | 56.04 % | 50.83 % | 50.83 % | 50.83 % | 50.83 % |
| Baseline+SN | **86.22** % | 55.93 % | 51.07 % | 51.07 % | 51.07 % | 51.07 % |
| Baseline+SWA | 86.09 % | 56.39 % | 50.69 % | 50.69 % | 50.69 % | 50.69 % |
| Baseline+SWA+SN (**Ours**) | 86.06 % | **56.79** % | **51.14** % | **51.14** % | **51.14** % | **51.14** % |
| AWP | 86.38 % | 56.32 % | 51.22 % | 51.21 % | 51.21 % | 51.21 % |

stochastic weighted average. We clearly see that the solution of spectral normalization lies in the widest loss basin. From Table 2, spectral normalization gives the highest robustness of $51.07\%$, compared to consistency loss $50.83\%$, baseline $50.38\%$, and stochastic weighted average $50.69\%$. This is also an evidence that the width of loss basin correlates with adversarial robustness.

As we view our work as a way to flatten loss landscape, we compare the flatness against AWP (Wu et al., 2020) in Fig. 2. We found out that even though our robust accuracy ($51.14\%$) is a bit lower than AWP $51.21\%$, our solution lies in a wider basin.

### 5.2. Other Unsuccessful Attempts

1. We attempted to incorporate Vision Transformer (ViT) (Dosovitskiy et al., 2021) as backbone for consistency loss. However, we observed that the consistency loss always explode to infinity even with multiple attempts to adjust hyperparameters.
2. We attempted to add covariance regularization introduced by VicReg (Bardes et al., 2022) to enforce more stable model outputs. However, we don't observe any performance improve with this.

## 6. Conclusion

We have adapted a spectral normalization, consistency regularization with H-Divergence and ensembling to impose 1-Lipschitz constraint in the context of adversarial attack. These contributions allow us too achieve 59.50% in robust accuracy evaluated on 1000 samples of CIFAR-10 testset evaluated with AutoAttack by Linf and radius 8/255.

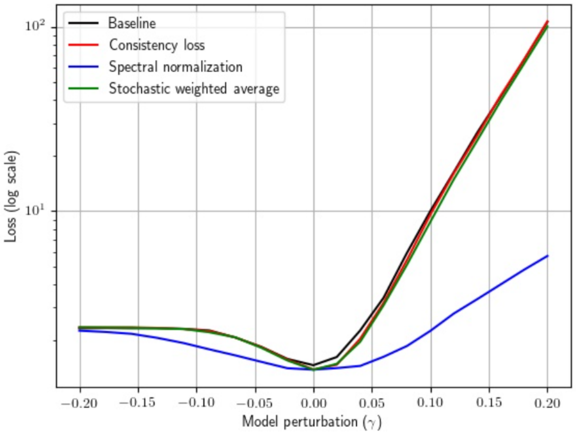

*Figure 1.* Baseline vs. Consistency loss vs Spectral normalization vs Stochastic weighted average. Y-axis is the loss function on training set in logscale and X-axis is the perturbation factor for each parameter in the model.

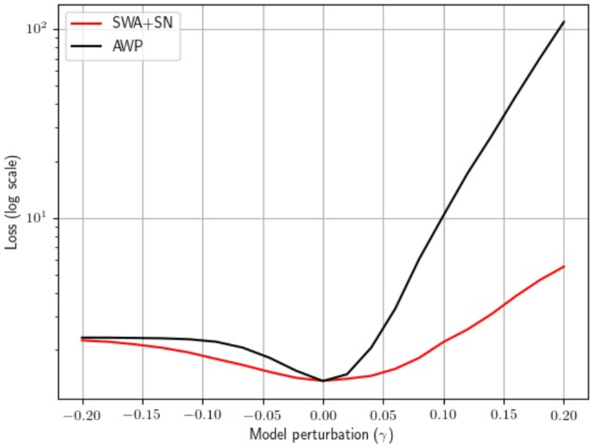

*Figure 2.* Ours (Red) vs AWP (Black). Y-axis is the loss function on training set in logscale and X-axis is the perturbation factor for each parameter in the model.

## References

Addepalli, S., Jain, S., Sriramanan, G., Khare, S., and Radhakrishnan, V. B. Towards achieving adversarial robustness beyond perceptual limits. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL https://openreview.net/forum?id=SHB_znlW5G7.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. arxiv 2017. *arXiv preprint arXiv:1701.07875*, 30:4, 2017.

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Croce, F. and Hein, M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020a.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020b.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *ArXiv*, 2017. URL https://arxiv.org/pdf/1708.04552.pdf.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Farnia, F., Zhang, J. M., and Tse, D. Generalizable adversarial training via spectral normalization, Nov 2018. URL https://arxiv.org/abs/1811.07457.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Huang, W. R., Emam, Z., Goldblum, M., Fowl, L., Terry, J. K., Huang, F., and Goldstein, T. Understanding generalization through visualizations. 2020.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Kannan, H., Kurakin, A., and Goodfellow, I. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Tack, J., Yu, S., Jeong, J., Kim, M., Hwang, S. J., and Shin, J. Consistency regularization for adversarial robustness. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL https://openreview.net/forum?id=w1Yj45siMi.

Terjék, D. Adversarial lipschitz regularization. *arXiv preprint arXiv:1907.05681*, 2019.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Wu, Y.-L., Shuai, H.-H., Tam, Z.-R., and Chiu, H.-Y. Gradient normalization for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6373–6382, 2021.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In Richard C. Wilson, E. R. H. and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016a. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87.

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

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Zhao, S., Sinha, A., He, Y., Perreault, A., Song, J., and Ermon, S. Comparing distributions by measuring differences that affect decision making. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KB5onONJIAU.

Zheng, Y., Zhang, R., and Mao, Y. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8156–8165, 2021.