Beyond Empirical Risk Minimization: Local Structure Preserving Regularization for Improving Adversarial Robustness

Anonymous CVPR submission

Paper ID 5408

Abstract

It is broadly known that deep neural networks are susceptible to being fooled by adversarial examples with perturbations imperceptible by humans. Various defenses have been proposed to improve adversarial robustness, among which adversarial training methods are most effective. However, most of these methods treat the training samples independently and demand a tremendous amount of samples to train a robust network, while ignoring the latent structural information among these samples. In this work, we propose a novel Local Structure Preserving (LSP) regularization, which aims to preserve the local structure of the input space in the learned embedding space. In this manner, the attacking effect of adversarial samples lying in the vicinity of clean samples can be alleviated. We show strong empirical evidence that with or without adversarial training, our method consistently improves the performance of adversarial robustness on several image classification datasets compared to the baselines and some state-of-the-art approaches, thus providing promising direction for future research.

1. Introduction

Deep neural networks (DNNs) trained under the Empirical Risk Minimization (ERM) framework have been tremendously successful on plenty of image understanding tasks like image classification [27]. However, it has been shown that DNNs are often fragile to adversarial examples which are crafted by adding an imperceptible perturbation on the natural input image to yield incorrect network predictions [4, 15, 38]. Such phenomena exist as a severe threat to the applicability of DNNs, especially when deployed in autonomous driving or surveillance systems.

Many pieces of work have been devoted to explaining the existence of adversarial examples [15, 23, 36]. Nevertheless, the community has not reached a consensus. One persuasive argument is that ERM forces the DNNs to memorize the training data thus fragile when evaluated on the adversarial examples which slightly differs from the training distribution [45]. To address the issue of vulnerability against adversarial examples, currently, the most effective defense strategy is adversarial training, which is a min-max optimization process to augment the DNNs with the generated adversarial examples on the fly during training [26, 28, 46]. Yet, the methods of adversarial training are still framed under the scope of the ERM, *i.e.*, treating each example independently and emphasizing memorizing the adversarial examples.

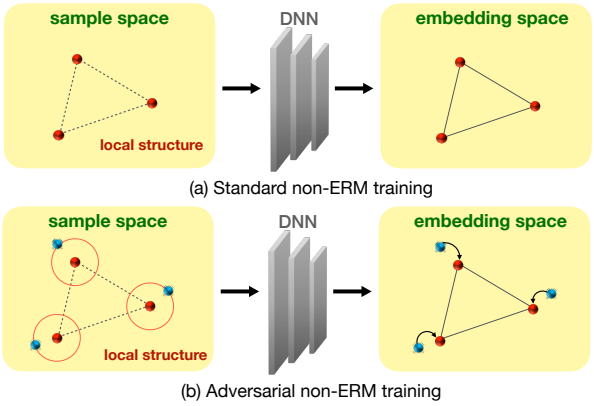

Figure 1. A schematic for Local Structure Preserving. For (a) standard training, the local structure in the sample space is constrained to be preserved in the embedding space. For (b) adversarial training, the adversarial example (blue dot) not only needs to be mapped to the embedding of the corresponding natural example (red dot) but also the local structure on natural examples should not be destroyed.

Despite the significant improvement in adversarial robustness, one of the main challenges of adversarial training is **data insufficiency**. It has been theoretically verified that the amount of samples required to train a robust model is remarkably more enormous than that of standard training, especially for high-dimensional images [35]. This explains the large gap between robust accuracy and standard accuracy when training with datasets with only limited samples.

Essentially, adversarial training under the ERM frame-

work treats each sample independently and derives the loss function in a sample-wise manner, making it require a massive amount of training samples to infer the complicated data distribution of adversarial examples. However, if we could utilize the underlying structure as prior knowledge, the strong demand for massive training samples could be alleviated. This motivates us to handle the data insufficiency challenge by exploring the internal structure that implicitly existed among training samples and investigating whether explicitly modeling them can benefit adversarial robustness.

Considering the proximity of the adversarial example and its corresponding natural example in the sample space, we argue that the ideally robust DNNs should be capable to map the nearby examples in the sample space to the embeddings close to each other. For each sample, its nearest neighbors in the training dataset, which are supposed to lie in the vicinity of this sample on the local manifold, naturally provide useful local structure information of the input space. If the training samples within a neighborhood are mapped to be close to each other in the output embedding space, the learned networks could be more regular and reduce the possibility to map nearby adversarial examples to uncertain embeddings which bring out misclassification.

With this insight, our idea is to **preserve the local structure of the input samples in the learned embedding space**. More specifically, the sample proximity of the local manifold in the input space should not be violated during the feedforward process of the deep networks. We propose an extremely concise regularization term, named **Local Structure Preserving**, to force the proximity of the local structure of both input space and output embedding space to be as close as possible. Note that the metric of input space, which is usually unknown, can be naturally obtained through a pretrained deep network that functions as a feature extractor. Through this simple regularization, our learning process treats each training sample as *non-i.i.d* and utilizes the latent relationship among them. As a consequence, it is not likely to map close samples in the input space to be farther away in the embedding space, which eases the vulnerability against adversarial attacks.

Moreover, our novel structural regularization can be simply incorporated into the adversarial training framework to further boost adversarial robustness. For a clean training sample, the local structure should not be altered significantly before and after it is adversarially perturbed. That means, the generated adversarial sample during adversarial training, should be mapped to similar positions in the embedding space as the corresponding clean one. Therefore, the output embedding of the adversarial example is not only close to the corresponding natural embedding but should be located within the local structure formed by its neighbors in the input space.

As a whole, our work mainly emphasizes the consideration of the structure that latently existed in the training samples, and thus is beyond the scope of ERM training. In this manner, the representation of the natural example or adversarial example is not independent for each sample, but follows a structure-wise alignment on each local manifold of the underlying data distribution. Thus the network prediction can be correlated by the local neighbors in the manifold. The idea of our non-ERM standard/adversarial training is depicted in Figure 1.

In a nutshell, our contributions are three-fold:

1. We propose a local structure preserving regularization for improving adversarial robustness. The structural information among samples is rarely investigated or used in previous adversarial learning methods.
2. We relate the proposed novel term with the goal of restricting the Lipschitz constant, which is known to be important for robustness.
3. We report strong empirical results on several commonly used image classification datasets, demonstrating the improvement of adversarial robustness by our method.

2. Related Work

2.1. Adversarial Attacks and Defenses

Since the seminal works of [4, 38] firstly discovered the vulnerability of deep neural networks and introduced the notion of adversarial examples that can attack well-trained networks to cause misclassification on these examples, a plethora of studies on generating aggressive adversarial attacks and establishing strong adversarial defenses have evolved. Early methods used the strategy of gradient obfuscation to defend against adversarial examples by intentionally masking the gradients since many attackers require the gradient information of the networks. Examples include defensive distillation [30], gradient shattering [5, 17], stochastic gradients [14, 43], vanishing/exploding gradients [34, 37], *etc*. However, it has been shown later that this strategy can be circumvented by stronger attacks [2, 3, 6].

In contrast, one of the most successful defensive strategies is adversarial training, which substitutes the clean training samples with the generated adversarial examples during training to improve the robustness of the deep networks. This was firstly introduced by [15], where they used the adversarial examples generated by the Fast Gradient Sign Method (FGSM) to replace the original clean ones for training, and then generalized to be applied on large-scale datasets by [26]. However, the robustness against the FGSM attacker seems questionable to other attacks, as shown in [40], where they augmented the adversarial training set with the ensemble of adversarial examples crafted from several similar classifiers. Later, [28] used Projected Gradient Descent (PGD) adversarial examples, which they called the "ultimate" first-order adversary, in adversarial training and improved robustness to

a wide range of adversarial attacks. Recently, multiple variants of adversarial training have been proposed. For example, TRADES [46] decomposed the adversarial training objective into two terms which control the natural accuracy and robustness as a trade-off. MART [42] further incorporated an explicit differentiation of clean misclassified examples in the trade-off objective.

As claimed before, the adversarial training methods require tremendously many samples to learn a robust model [35]. Some researchers augment millions of external data [39] into training [7, 16, 32] to achieve the state-of-the-art performance. But the new challenge emerges as the extremely high requirement for computational resources. For example, to accomplish adversarial training in a reasonable time, Cloud TPUs are required. Even worse, the amount of data is limited itself for certain tasks, like medical image analysis. This motivates us to study from another angle – to utilize the data structural relationship in a non-ERM manner.

2.2. Non-ERM Learning Methods

Under the non-ERM training framework, our method treats each training sample as *non-i.i.d.* and considers the structural modeling of the training samples through the idea that each sample can be influenced by its neighboring samples. A similar insight is shared in a series of data augmentation methods. For example, the method Mixup [45] trained the networks on convex combinations of the pairs of examples and their labels. Following this, CutMix [44], AugMix [22], *etc*., were proposed to augment more diverse mixup combinations of the training samples. These methods have been shown to benefit adversarial robustness. However, their methodology is significantly different from ours since their training pairs are randomly chosen and no structures are purposefully mined.

3. Method

In this section, we present the proposed idea of Local Structure Preserving (LSP). First, in Section 3.1 and 3.2, we introduce our method in the settings without and with adversarial training. At the end of this section, we discuss the theoretical motivation of our idea.

3.1. Local Structure Preserving

Suppose a natural example x lies in a metric space $(\mathcal{X}, ||\cdot||)$, where $\mathcal{X}$ denotes the sample space and $||\cdot||$ denotes the metric. The input example x is associated with a semantic class label $y \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the label space. The goal is to learn a robust DNN classifier f that maps from the sample space $\mathcal{X}$ to the label space $\mathcal{Y}$, by N training pairs $\{(x_i, y_i)\}_{i=1}^N$ randomly sampled from the joint space $(\mathcal{X}, \mathcal{Y})$. The robustness hereby is toward adversarial examples x' which is crafted within the δ-neighborhood (defined by L_p norm $||\cdot||_p$) of the input x, as follows:

$$\forall x' \in \{x' : ||x' - x||_p \leq \delta\},$$
$$\arg\max_{c \in Y} f(x')_c = \arg\max_{c \in Y} f(x)_c = y. \quad (1)$$

Here $f(x)$ is in the form of the probabilistic vector where the index of the largest element, denoted by the subscript, indicates the label prediction.

Next, we introduce our local structure preserving regularization. Suppose for an input example x, its nearest neighbors in the training dataset are $\{x_{n_1}, x_{n_2}, ..., x_{n_m}\}$, where m denotes the area of neighborhood. We call x as an anchor point and derive the differences of its neighbors to the anchor as:

$$p_x^i = \frac{||x - x_{n_i}||}{\sum_{i=1}^m ||x - x_{n_i}||}, \quad i = 1, 2, .., m. \quad (2)$$

Here, the normalization in the denominator is applied to obtain the probability vector $P_x = [p_x^1, p_x^2, ..., p_x^m]$. This vector explicitly encodes the local structure information of each input example in the sample space.

After the feedforward process of deep network f, the input x and its neighbors are mapped to the output embedding space for classification, *i.e*., the layer output before softmax of the network. Again, we can still obtain the vector conveying the differences of $f(x)$ and $\{f(x_{n_1}), f(x_{n_2}), ..., f(x_{n_m})\}$ in a similar manner as Equation 2:

$$q_x^j = \frac{||f(x) - f(x_{n_j})||}{\sum_{j=1}^m ||f(x) - f(x_{n_j})||}, \quad j = 1, 2, .., m. \quad (3)$$

And the probability vector $Q_x = [q_x^1, q_x^2, ..., q_x^m]$ accordingly indicates the local structure of the output embedding space around the anchor.

The goal of local structure preserving is to preserve these two structures, *i.e*., maintaining the proximity between closed examples of both spaces, expressed in the form of P_x and Q_x. A straightforward way is to minimize the discrepancy as follows:

$$\mathcal{L}_{\text{LSP}} = D(P_x, Q_x). \quad (4)$$

The scheme of defining discrepancy can be various. We discuss several terms, including KL-divergence, cosine similarity, and L_1/L_2 distance in Section 4.3. By introducing the regularization term in Equation 4, we no longer treat each training sample as *i.i.d.*, instead utilizing the local information among them.

However, a remaining unresolved problem is how we define the metric $||\cdot||$ in Equation 2 and 3. For Equation 3, the metric is simply L_2 distance, since $f(x)$ encodes the feature representation and is normalized to follow an approximately Gaussian distribution. As for the metric in Equation 2, since

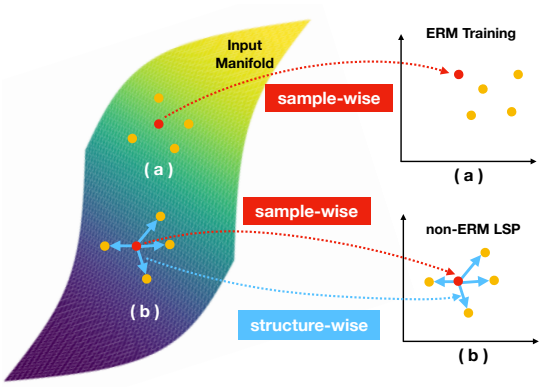

Figure 2. Left: the manifold represents the input space. Right: two coordinates represent the output embedding spaces. a) Only using sample-wise CE loss under the ERM training framework, the local structure of output space is not restricted and the order of local structure could be distorted. b) Regularizing ERM training with our proposed LSP, the local structure of input space is favored to be preserved in the output space.

we have no clue about the distribution of the input space, one natural procedure is that we can use a pre-trained neural network as a feature extractor. Motivated by [41], we use MoCo [18], denoted by g, trained by contrastive loss in a self-supervised learning manner as the feature extractor to approximate the metric of input space, as follows:

$$||x_i - x_j|| \approx ||g(x_i) - g(x_j)||, \quad \forall x_i, x_j \in \mathcal{X}. \tag{5}$$

Equation 4 serves as a regularization term in our method. Together with the traditional Cross Entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{y \in Y} y \log(f(x)_y), \tag{6}$$

the total loss for training the network is as follows:

$$\mathcal{L}_{\text{Total}} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (\mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{LSP}}), \tag{7}$$

where λ is a balancing parameter to control the trade-off of two loss terms.

In Figure 2, we illustrate the insight behind our idea. Without the aid of local structure preserving regularization, training solely by Cross Entropy under the ERM framework may lead to the result that the local structure of the output embedding space violates the original local structure in the input space, since no constraints are explicitly employed. Thus the adversarial examples which are also located within the neighborhood of the input sample may be mapped to the place far from the input in the output embedding space. By adding such an explicit structural prior constraint, the local structure around each input sample is preserved in the output embedding space. By regularizing each piece of the local manifold of the input space, the learned deep classifier is deemed to be more regular and potentially more resistant to adversarial attacks.

Algorithm 1 Local Structure Preserving

Input: training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, pretrained MoCo.
repeat
  Sample (x, y) from $\mathcal{D}$.
  Search m nearest neighbors for x by the ultimate feature of MoCo.
  Encode the local structure of x in the input sample space by Equation 2.
  if standard training **then**
    Feed-forward to compute the loss in Equation 7.
  else if adversarial training **then**
    Feed-forward to compute the loss in Equation 11.
  end if
  Back-propagate to update the gradient of f.
until training finished.

3.2. Combining with Adversarial Training

Section 3.1 introduces our method to improve the robustness of agnostic models. Here we note that our method can also be combined with adversarial training to further boost the robustness. The core of adversarial training is to solve a min-max optimization problem, in which the inner max procedure seeks to find an adversarial example x' by moving toward the gradient ascent direction to the input natural example x. Therefore, in adversarial training, the Cross Entropy term in Equation 6 becomes:

$$\tilde{\mathcal{L}}_{\text{CE}} = -\sum_{y \in Y} y \log(f(x')_y). \tag{8}$$

For the regularization term, we argue that the adversarial counterpart x' of the natural example x should be mapped by f to preserve the original local structure of x, since we wish that the output embedding of x' should not be apart from the embedding of x, thus Equation 2 still holds and Equation 3 becomes:

$$\tilde{q}_x^j = \frac{||f(x') - f(x_{n_j})||}{\sum_{j=1}^{m} ||f(x') - f(x_{n_j})||}, \quad j = 1, 2, .., m. \tag{9}$$

The LSP term and final objective in Equation 4 and 7 are adjusted accordingly as follows:

$$\tilde{\mathcal{L}}_{\text{LSP}} = D(P_x, \tilde{Q}_x). \tag{10}$$

$$\tilde{\mathcal{L}}_{\text{Total}} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (\tilde{\mathcal{L}}_{\text{CE}}^x + \lambda \tilde{\mathcal{L}}_{\text{LSP}}^x). \tag{11}$$

The pseudo-code of our proposed method is presented in Algorithm 1.

3.3. Theoretical Analysis

In the previous work, [8, 10, 21] studied to reduce the Lipschitz constant of the network f with respect to an input example x, to improve the robustness. To highlight the theoretical motivation of LSP, we use the following off-the-shelf theorem to demonstrate that if the classifier is a Lipschitz function, the adversarial robustness can be guaranteed, and then analyze its relationship to LSP in an *asymptotic* perspective.

Theorem 1. *Suppose $a, b \in \mathcal{Y}$ are the most and second-most likely class prediction of the classifier f, i.e.,*

$$a = \arg\max_{l \in \mathcal{Y}} f(x)_l, \quad b = \arg\max_{l \in \mathcal{Y} \setminus \{a\}} f(x)_l, \tag{12}$$

and p_a, p_b are the predictive probability of the classifier w.r.t. label a and b, i.e.,

$$p_a = f(x)_a, \quad p_b = f(x)_b. \tag{13}$$

If function f is locally L_x-Lipschitiz on input $x \in \mathcal{X}$, i.e.,

$$\forall x \in X, \forall x', s.t.\ ||x' - x|| \leq \delta, \tag{14}$$

$$||f(x) - f(x')|| \leq L_x ||x - x'||. \tag{15}$$

Then f is guaranteed to be robust at x up to any perturbation of magnitude $\delta \leq \frac{1}{2L_x}(p_a - p_b)$.

Theorem 1 implies that if the learned classifier satisfies the Lipschitz condition (Equation 15), certified robustness against the adversarial examples within the neighborhood of input x can be realized. The upper bound of perturbation magnitude δ is influenced by two factors, L_x and $(p_a - p_b)$. For the latter one, the explicit goal of classification is to make it larger, which indicates more confidence in the prediction of true label. As for L_x, from an *asymptotic* view, if the training samples near the input x are extremely dense, minimizing the LSP regularization in Equation 4 is equivalent to minimizing the discrepancy of $||f(x) - f(x')||$ and $||x - x'||$ in Equation 15, thus L_x is controllable as a constant. With constant denominator L_x and larger numerator $(p_a - p_b)$ during training, the robustness radius δ of our model is expected to be larger.

Although such asymptotic analysis is ideal and can hardly be practical, the conducted experimental results in Section 4 evidence the effectiveness of our method in real, thus our idea that treating each sample as *non-i.i.d.* and preserving the local structure is verified to be effective.

4. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed idea in boosting the performance of adversarial robustness against various adversarial attacks.

4.1. Experimental Setups

Datasets We evaluate the performance of our method on four public image classification datasets: (1) CIFAR-10 [25], which contains a training set of 50,000 examples and a testing set of 10,000 examples. Each example is a 32x32 color image associated with a label from 10 classes; (2) STL-10 [9], which has 10 classes, with 500 training images and 800 testing images per class. Each example is a 96x96 color image; (3) Street-View House Numbers (SVHN) [29], which consists of 73,257 digits for training, and 26,032 digits for testing. Each example is a 32x32 color image associated with a label from 10 classes; (4) Tiny-ImageNet [13], which contains 200 classes and each class has 500 images for training and 50 images for testing. Each example is a 64x64 color image. We train with an NVidia RTX 2080 GPU with 24G memory for all datasets.

Evaluation Protocol To evaluate the adversarial robustness, we use the four most common attackers to attack the trained models of ours and all comparison methods, including (1) FGSM [15]: a simple one-step scheme adversary, which finds adversarial examples by moving a single step in the ascent direction to the gradient of the loss function. (2) CW [6]: an optimization-based attacker that tries to find the minimal-distorted perturbation. The scheme is by minimizing the margin loss to generate adversarial examples that have minimal difference between the logit values of the most and second-most likely classes. (3) PGD [28]: a multi-step extension of FGSM with the random initialization, which was claimed to be the strongest first-order adversary. (4) AutoAttack (AA) [12], currently the most reliable evaluation of adversarial robustness that is composed of an ensemble of four diverse attacks including three white-box attacks (APGD-CE [12], APGD-DLR [12], and FAB [11]) and one black-box attack named Square Attack [1]. The implementation of attack algorithms is by Torchattacks [24] . Unless otherwise stated, we use L_∞-norm bounded perturbations for all adversarial attacks. Further, we also report the clean accuracy where the model is under no attacks.

4.2. Comparison and Result Analysis

Since the proposed idea is independent of adversarial training, we conduct experiments under two settings – w/wo adversarial training, in order to show our idea is beneficial to adversarial robustness *per se*, and can further boost the performance of adversarial training. We denote our method for standard and adversarial training as LSP and LSP+.

4.2.1 Standard Training

Comparison Methods For the setting of standard training, i.e., adversarial examples are not generated during training, we compare with: 1) VANILLA, baseline classification networks by Cross Entropy loss; 2) MIXUP [45], which is

Methods	Attack Types				
	Clean	FGSM	CW	PGD	AA
CIFAR-10					
VANILLA	94.56	36.38	3.68	6.26	2.44
MIXUP	94.96	59.40	47.78	12.23	0.17
CUTMIX	**95.87**	55.94	20.73	13.14	0.10
AUGMIX	94.46	47.48	16.33	16.17	9.08
LSP	94.22	**61.16**	**62.53**	**28.00**	**15.11**
STL-10					
VANILLA	78.97	26.65	48.12	19.58	17.52
MIXUP	**80.63**	27.36	46.95	14.31	8.76
CUTMIX	78.13	30.97	48.56	19.88	16.88
AUGMIX	76.12	33.10	49.68	24.06	20.83
LSP	76.85	**34.23**	**52.49**	**25.69**	**23.69**
SVHN					
VANILLA	96.55	71.08	58.03	47.80	40.60
MIXUP	96.65	71.14	62.74	23.87	6.69
CUTMIX	**97.29**	75.38	42.88	26.95	1.34
AUGMIX	96.87	71.33	45.71	44.93	36.39
LSP	96.78	**77.19**	**78.86**	**52.92**	**45.74**
Tiny-ImageNet					
VANILLA	65.37	9.19	25.85	2.56	1.62
MIXUP	66.51	13.59	28.05	4.83	1.19
CUTMIX	**68.94**	12.03	24.77	2.58	0.45
AUGMIX	65.52	16.55	32.55	5.11	3.17
LSP	67.58	**17.64**	**36.08**	**7.79**	**5.28**

Table 1. The classification accuracy comparison of methods by standard training. on different datasets. Numbers in bold indicate the best.

trained on convex combinations of pairs of samples and their labels; 3) CUTMIX [44], where the image patches are cut and pasted and their labels are mixed proportionally to the area of the patches; 4) AUGMIX [22], which further mixes diverse augmented images and adopts a Jensen-Shannon Divergence consistency loss to achieve state-of-the-art performance. For the latter three methods, we compare with them since they also consider the structure among samples that share a similar spirit with our idea, and have provided evidence to improve adversarial robustness.

Experimental Settings For CIFAR-10, STL-10 and SVHN, we use ResNet-18 [19] as the backbone. For Tiny-ImageNet, we use Preact ResNet-18 [20] as the backbone. The backbones are consistent for all comparison methods to ensure fairness. For all methods, we train 100 epochs with the optimizer of Stochastic Gradient Descent, and the learning

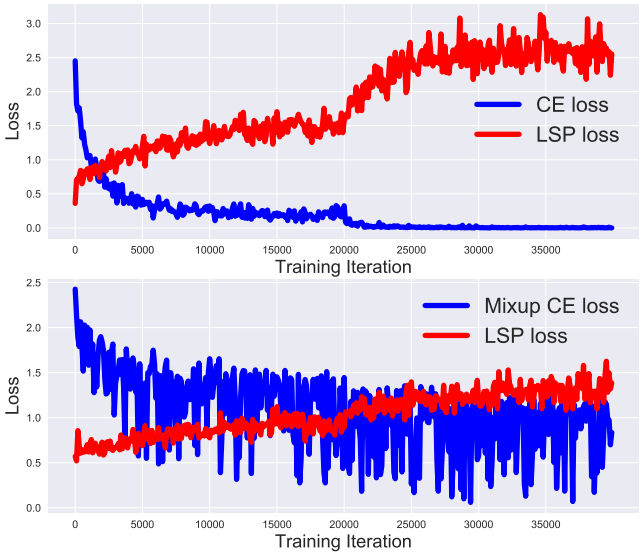

Figure 3. The training loss curves of VANILLA (top) and MIXUP (bottom) on CIFAR-10. Model is trained without LSP regularization. For both methods, LSP loss increases during training.

rate is initially set as 0.1, divided by 10 in the epoch 75 and 90. For the proposed local structure preserving loss, we use MSE loss, *i.e.*, minimizing the L_2 distance, in Equation 4 and the balancing parameter λ in Equation 7 is tuned to be 1. The number of nearest neighbors is 8. Ablation studies of these hyperparameters are provided in Section 4.3.

For the attack parameters in evaluation, since not augmented with adversarial examples during training, it is hard to expect these methods can defend against adversarial attacks with the commonly used maximum perturbation budget $\delta = 8/255$. Therefore, we seek to set the budget as a milder $2/255$ for all attacks. For the CW attack, the box-constraint parameter c is 0.01. For the PGD attack, the attacking step is 10 with step size $1/255$.

Results Analysis In Table 1, we show the results of all methods that are without adversarial training. Basically, MIXUP, CUTMIX and AUGMIX, which are also in the scope of the non-ERM training framework, can all improve the robustness against single-step attack FGSM over the baseline VANILLA model, which has also been verified in their papers. However, when it comes to stronger attacks CW and PGD, the improvement is not consistent. For example, MIXUP decreases the robustness against PGD on STL-10 and SVHN, CUTMIX decreases the robustness against CW on SVHN and Tiny-ImageNet, and AUGMIX decreases the robustness against both attacks on SVHN. Even worse, in consideration of the most reliable robustness evaluation by AA, MIXUP and CUTMIX worsen the robustness on all datasets, while AUGMIX worsens the robustness on SVHN.

As a comparison, our method consistently improves the robustness performance against all attacks over VANILLA,

and shows superior robustness over all three non-ERM methods, just by adding the local structure preserving regularization term. This indicates that preserving the local structure in the original space during training can make the network training more regular and better repair the "pockets" in the data manifold [38] which turns into adversarial examples. However, three non-ERM comparison methods only provide flimsy robustness, and we suppose the reason is that only restricting the data structure from the convex combination perspective is insufficient to provide reliable robustness due to the extreme non-linearity of high-dimensional data space. On the contrary, our operation on the data structure is more general, leading to stronger robustness. Also, note that the defensive effect of LSP is most prominent against CW attack, particularly for CIFAR-10. We analyze the reason is due to that CW attack generates adversarial examples by encouraging the logit value of the second-most likely class in the output embedding vector to exceed the one of the most likely class, while our method imposes prior information on the output embedding, thus not easy to be fooled.

In Figure 3, we depict the loss curves of two terms, Cross Entropy and LSP, for both VANILLA and MIXUP model. The Cross Entropy losses of both models exhibit a decreasing trend as anticipated, while the LSP losses keep rising. For VANILLA, this indicates that ERM memorizes the training data while may destroy the interior data structure, thus leading to the poor robustness on adversarial examples which are slightly different distributed. While for MIXUP, the LSP loss is relatively smaller but still increasing, which means the data structure is still undermined. Therefore, the LSP loss needs to be explicitly optimized and our performance shows its importance on adversarial robustness.

4.2.2 Adversarial Training

Comparison Methods In this experiment, we compare our performance with various state-of-the-art adversarial training methods, including 1) AT [28], which substitutes the original training samples with the PGD adversarial examples in training and is known for the resistance to a wide range of adversarial attacks; 2) TRADES [46], which regularizes to trade adversarial robustness off against clean accuracy, and we set trade-off parameter as 6 for optimal robustness following their paper; 3) MART [42], which extends TRADES by considering a regularized adversarial loss which incorporates an explicit differentiation of misclassified examples. The balancing parameter in MART is 5 following their paper; 7) HAT [31], a recent state-of-the-art adversarial training method that achieves a better accuracy-robustness trade-off by reducing the excessive margin of clean samples. We exactly follow their reported hyperparameters.

Experimental Settings Here we consider the three commonly evaluated dataset – CIFAR-10, SVHN and Tiny-ImageNet for adversarial robustness. The backbone network for each dataset is kept the same as in Section 4.2.1. We follow the training settings in [31]. The adversarial example generated during training is by PGD attack with perturbation budget $\delta = 8/255$ and 10 steps. We adopt the early stopping scheme [33] to evaluate the model with the best adversarial accuracy during training. For testing, CW attack refers to the PGD attack with CW loss and the perturbation budget δ is 8/255.

Methods	Attack Types				
	Clean	**FGSM**	**CW**	**PGD**	**AA**
CIFAR-10					
AT	83.60	58.19	50.91	52.37	47.15
TRADES	81.77	57.54	50.96	52.86	48.99
MART	78.51	**58.64**	50.26	**55.29**	47.79
HAT	**84.75**	58.25	50.95	52.98	48.64
LSP+	84.74	58.53	**51.01**	53.53	**49.26**
SVHN					
AT	92.02	67.94	54.51	57.19	49.50
TRADES	89.69	68.02	56.30	60.11	53.15
MART	89.83	68.44	54.89	61.22	49.64
HAT	91.98	**76.38**	**58.51**	61.11	51.43
LSP+	**92.61**	76.03	58.29	**62.33**	**53.90**
Tiny-ImageNet					
AT	48.79	25.40	21.09	23.38	16.92
TRADES	50.01	24.95	18.33	22.93	16.90
MART	44.35	25.42	19.62	23.48	17.01
HAT	51.83	24.66	18.25	22.16	16.83
LSP+	**52.57**	**25.53**	**21.98**	**23.58**	**17.63**

Table 2. The classification accuracy comparison of methods by Adversarial Training. Numbers in bold indicate the best. Note that comparing to our baseline AT, we achieve stronger robustness against all attacks.

Results Analysis As shown in Table 2, when evaluating with the most reliable robustness metric AA, our method achieves better performance than all comparison methods on all datasets. Compared to our baseline method AT, all evaluation metrics are improved, where some are of them significant, like AA for SVHN. Notably, the clean accuracy of LSP+ is also better or on par with all comparison methods. This again verifies the effectiveness of our method. It is noteworthy that we do not aim to claim that our method is the strongest defensive method, in contrast, we want to show the plausibility of the proposed local structure preserving regularization by consistent improvement over baseline without this regularization and competitive performance over

Which to Preserve	None	Global	Local
Clean	**94.56**	93.56	94.22
CW	3.68	44.17	**62.53**
PGD	6.26	21.26	**28.00**

Table 3. The performance comparison of preserving the global and local structure on CIFAR-10.

Loss	K-L	Cosine	L1	L2
Clean	94.38	92.67	86.42	93.65
CW	40.88	56.52	57.17	**58.14**
PGD	5.35	13.60	13.60	**15.87**

Table 4. Various choices of local structure preserving loss terms, testing without adversarial training on the CIFAR-10 dataset.

# Neighbors	2	4	8	16	32
Clean	93.37	93.63	**94.22**	93.78	94.03
CW	54.86	54.26	62.53	63.25	**64.45**
PGD	12.94	14.81	28.00	29.99	**30.14**

Table 5. Various choices of the number of local neighborhood.

strong defensive methods.

4.3. Ablation Study

Here we conduct various ablation experiments to analyze the influence of important factors and parameters in our model.

Global or Local Structure Preserving? The first question is whether the global or local structure should be preserved. The global scheme preserves the structure of random examples globally sampled from the dataset. In Table 3, the robustness performance of the global scheme is much worse than the local scheme in our method.

Local Structure Preserving Loss Term An important thing to consider is the optimal form of local structure preserving loss term as a regularizer. Here we empirically study the difference of several forms of discrepancy in Equation 4, including minimizing the KL-divergence, maximizing the cosine similarity, and minimizing the L1/L2 distance. The results are shown in Table 4. As shown, all choices of local structure preserving terms achieve some improvements in robustness against different attacks. Among these, minimizing L2 distance works the best.

Neighborhood Area In Table 5, when we choose different number of nearest neighbors in our algorithm, the results show that more neighbors could benefit the performance, since more useful structure prior information is involved in the learning process. This implicitly supports the asymptotic theoretical conjecture in Section 3.3.

λ	0.1	0.5	1	2	5	10
Clean	**94.45**	94.16	94.22	92.80	90.06	88.64
CW	36.91	55.59	62.53	**65.93**	55.39	52.18
PGD	8.98	19.08	**28.00**	25.58	15.51	16.12

Table 6. Various choices of loss weighting parameter λ.

Methods	VANILLA	MIXUP	CUTMIX	AUGMIX	**LSP**
PGD-L_∞	6.26	12.23	13.14	16.17	**28.00**
PGD-L_2	9.73	18.62	22.38	19.73	**34.30**

Table 7. Evaluation with different perturbation norms of PGD attack, where the budget for L_2 norm is 128/255.

Loss Weighting Parameter We also study the effect of the weighting parameter λ in Equation 7. As shown in Table 6, for all the attacks, a too-small λ can not thoroughly benefit the adversarial robustness via the proposed LSP term; while a too-large λ will be harmful to the data fitting which caused performance degradation.

Different Perturbation Norms Previously we evaluate with adversarial attacks based on L_∞-norm. Here we also test the robustness against the PGD attack with L_2-norm in Table 7. Again, our method shows stronger robustness in the metric norm of L_2 perturbation.

5. Conclusion

In this paper, we rethink the defect of the ERM learning framework for adversarial robustness. We argue that treating each training sample as *i.i.d.* may lead to the violation of the local structure of the output embedding space against the original input space, thus leading to learning a distorted network mapping that has poor adversarial robustness. To solve this issue, we propose a novel while simple idea to preserve the local structure from the input space to the output embedding space, thus making the network mapping regular. From our experimental observations, such a simple regularization significantly and consistently improves the metrics of adversarial robustness against various popular strong adversarial attacks on four datasets. Our method is also competitive against several state of the art adversarial training methods. However, we do not declare that we are the strongest defensive method at the moment. Rather, we intend to call the community to focus on investigating the relationship between *non-i.i.d.* structural modeling during learning and adversarial robustness through this work. For future work, the more delicate structural modeling scheme is worth to be studied and deeper theoretical certificates are necessary.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020. 5

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018. 2

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pages 284–293. PMLR, 2018. 2

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013. 1, 2

[5] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 2

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 2, 5

[7] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[8] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017. 5

[9] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[10] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 5

[11] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020. 5

[12] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020. 5

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[14] Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. 2

[15] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2, 5

[16] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 3

[17] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. 2

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. 6

[21] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2263–2273. Curran Associates, Inc., 2017. 5

[22] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations*, 2020. 3, 6

[23] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in Neural Information Processing systems*, 32, 2019. 1

[24] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 5

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[26] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017. 1, 2

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[28] 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*, 2018. 1, 2, 5, 7

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 5

[30] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy*, pages 582–597. IEEE, 2016. 2

[31] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Learning Representations*, 2022. 7

[32] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948, 2021. 3

[33] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. 7

[34] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. 2

[35] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in Neural Information Processing systems*, 31, 2018. 1, 3

[36] Adi Shamir, Itay Safran, Eyal Ronen, and Orr Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance. *arXiv preprint arXiv:1901.10861*, 2019. 1

[37] 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*, 2018. 2

[38] 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*, 2014. 1, 2, 7

[39] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008. 3

[40] 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*, 2018. 2

[41] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proceedings of the European Conference on Computer Vision*, 2020. 4

[42] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. 3, 7

[43] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 2

[44] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019. 3, 6

[45] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 1, 3, 5

[46] 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*, pages 7472–7482. PMLR, 2019. 1, 3, 7