Defense Against Adversarial Attacks via Controlling Gradient Leaking on Embedded Manifolds

Yueru Li*, Shuyu Cheng*, Hang Su†, and Jun Zhu†

Dept. of Comp. Sci. and Tech., BNRist Center, Institute for AI, THBI Lab,
Tsinghua University, Beijing, 100084, China
{liyr18, chengsy18}@mails.tsinghua.edu.cn,
{suhangss, dcszj}@tsinghua.edu.cn

Abstract. Deep neural networks are vulnerable to adversarial attacks. Though various attempts have been made, it is still largely open to fully understand the existence of adversarial samples and thereby develop effective defense strategies. In this paper, we present a new perspective, namely gradient leaking hypothesis, to understand the existence of adversarial examples and to further motivate effective defense strategies. Specifically, we consider the low dimensional manifold structure of natural images, and empirically verify that the leakage of the gradient (w.r.t input) along the (approximately) perpendicular direction to the tangent space of data manifold is a reason for the vulnerability over adversarial attacks. Based on our investigation, we further present a new robust learning algorithm which encourages a larger gradient component in the tangent space of data manifold, suppressing the gradient leaking phenomenon consequently. Experiments on various tasks demonstrate the effectiveness of our algorithm despite its simplicity.

Keywords: Gradient leaking, DNNs, Adversarial robustness

1 Introduction

Deep neural networks (DNNs) have shown impressive performance in a variety of application domains, including computer vision [13], natural language processing [18] and cybersecurity [6]. However, it has been widely recognized that their predictions could be easily subverted by the adversarial perturbations that are carefully crafted and even imperceptible to human beings [27]. The vulnerability of DNNs to adversarial examples along with the design of appropriate countermeasures has recently drawn a wide attention.

Recent years have witnessed the development of many kinds of defense algorithms. However, it still remains a challenge to achieve robustness against adversarial attacks. For example, defenses based on input transformation or randomization usually give obfuscated gradients [1] and hence could be cracked by

*Equal contribution. †Corresponding author.

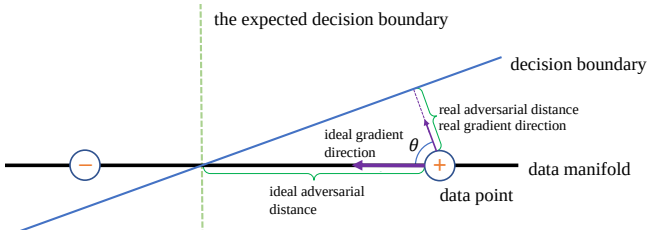


Fig. 1: An illustration of gradient leaking. If the manifold and decision boundaries are relatively flat, the robust distance in this case is approximately the order of $O(\cos(\theta))$ (θ is the angle between gradient direction and the tangent space) of the theoretical longest distance on the manifold.

adaptive attacks. Defenses such as adversarial training [17] and controlling Lipschitz constants [20] will significantly decrease the gradient norm of the model loss function, which may sacrifice the accuracy on natural images. The difficulty in designing defense algorithms is partly due to the unclear reason for the existence and pervasiveness of adversarial examples. Though various attempts have been made including the linearity of the decision boundary [8], insufficiency of samples [25], the concentration property of high dimensional constraints [26] and the computational constraints [2], it is still an open question to explore the intrinsic mechanism of adversarial examples and design better defense algorithms.

In our paper, we analyze the existence of adversarial examples from the perspective of the data manifold, and propose a new hypothesis called *Gradient Leaking Hypothesis*. When analyzing the adversarial robustness at a given data point (which is classified correctly as class y), we focus on the *adversarial gradient*, i.e. the gradient of the objective function in untargeted attack such as the negative predicted likelihood function of class y. As is illustrated in Fig. 1, the ideal direction of adversarial gradient lies in (the tangent space of) the data manifold, so that only the necessary gradient to classify the dataset remains. However, through extensive analysis we find that in most normally trained models, the gradient points to a nearly perpendicular direction to the data manifold, resulting in the leakage of gradient information and weak robustness in adversarial attacks. In such cases, adversarial examples can be found outside of but very close to the data manifold. As shown in the figure, the perturbation norm of the adversarial is the order of $\cos(\theta)$ relative to that in the ideal case.

As the adversarial gradient is approximately perpendicular to the decision boundary between the original class and the class of the adversarial example, a more intuitive description of gradient leaking is that the decision boundary is nearly *parallel to* the data manifold, which implies vulnerability to adversarial attacks. To show its reason visually, we illustrate an inspiring example in Fig. 2(a). The data points are distributed in different colored regions corresponding to the different classes. We identify an approximate 1-dimensional data manifold shown as the black parabolic curve (we exaggerate the distance from data points to the

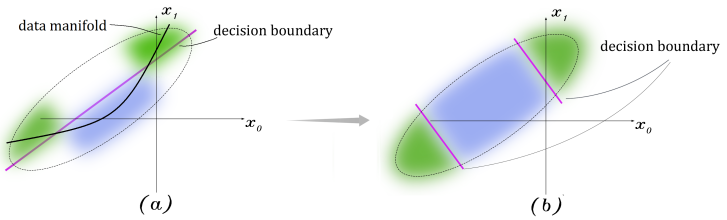


Fig. 2: An illustrative case when gradient leaking happens (a), and our method to train a robust classifier through preprocessing the dataset (b).

manifold in the figure). The dataset is linearly separable shown as the purple line. However, it is nearly parallel to the data manifold, which does not correspond to a robust classifier. The adversary could perturb the data points in a perpendicular direction to the data manifold, suggesting that gradient leaking happens. We see that vulnerability to the adversarial attack is usually caused by some small-scale features (e.g., the direction perpendicular to the decision boundary in Fig. 2(a)) which are easily learned by the classifier since they might be highly correlated to the labels. This is also in line with recent studies such as [11] which demonstrate that adversarial examples can be attributed to the non-robust features useful for classification.

Based on the above analysis, we present a novel data preprocessing framework to reduce gradient leaking during training, thereby enhancing the adversarial robustness. We first make the data manifold flat by projecting the dataset to the PCA principal subspace to eliminate the small-scale features mentioned above. After that, we add independent noise in the normal space of the data manifold to enforce the classifier to learn a decision boundary nearly perpendicular to the data manifold. The result is illustrated in Fig. 2(b), in which we change the data distribution to learn a robust classifier that remains a high accuracy on the original dataset. Extensive experiments demonstrate that we can obtain a more robust model in image classification and face recognition tasks. By simply preprocessing images before training, we can achieve a 2$\sim$3 times improvement on the mean perturbation norm of adversarial examples under a powerful ℓ_2-BIM attack. Our algorithm is nearly orthogonal to other methods and can be easily integrated with them. As an example, we integrate our method with the Max-Mahalanobis center (MMC) loss training [19], and reach much higher robustness compared to the baseline MMC. The robustness of our obtained model is close to that of the model trained by adversarial training [17], yet with a much higher clean accuracy and much lower training time cost.

Contribution. In this paper, (1) we propose a novel Gradient Leaking Hypothesis to explain the existence of adversarial samples and analyze its possible mechanisms under an empirical investigation; (2) and we present a novel algorithm of robust learning based on our hypothesis, yielding superior performance in various tasks.

2 Related Work

[2] analyzes four opinions on adversarial samples. The authors point out that though robust models could exist, the requirement of robustness and computational efficiency might contradict in specific task. Our analysis and empirical evidence support the idea that there is a trade-off between "easiness to learn" and robustness. [25] claims that sampling complexity may be the reason for adversarial samples, but we point out that having more samples on the manifold may not help if the gradient leaking is not suppressed efficiently.

The authors in [7] analyze the geometric properties of DNN image classifiers in the input space. They show that DNNs learn connected classification regions, and the decision boundary in the vicinity of data points is flat along most directions. But they claim the results reflect the complex topological properties of classification regions. We believe that it could happen in spaces with simple topology (for example, homeomorphic to $\mathbb{R}^D$) caused by gradient leaking.

[11] points out that adversarial examples are highly related to the presence of non-robust features, which are brittle and incomprehensible to humans. They distill robust features from a robust model. Our work bridges the inherent geometry of data manifold and the robustness of features, and could be considered as developing a way to quantitatively study the reliance of classifier on these non-robust features, and using them to improve the robustness of DNNs.

The authors of [9] systematically suggest a framework of Gaussian noise randomization to improve robustness. They inject isotropic Gaussian noise, whose scale is trainable through solving the Min-Max optimization problem embedded with adversarial training, at each layer on either activation or weights. Our noise adding procedure could be seen as a variant considered as a subspace selection procedure to add Gaussian noise on the input.

Defense-GAN [24] shows that by projecting data points to lower-dimensional manifold during inference time, the classifier can be more robust. Their defense partially relies on obfuscated gradients [1] and is only tested on MNIST. By contrast, our method is applied to the training process, able to obtain a robust classifier used in a vanilla way, and more scalable to larger datasets.

3 Gradient Leaking Hypothesis

In this section, we first present the Gradient Leaking Hypothesis formally, and analyze its relationship with robustness empirically.

3.1 Preliminary

In general, the data points $\{x_i\}$ in a natural image dataset lie on a manifold $\mathbb{M}$ embedded in $\mathbb{R}^D$, which is called the *ambient space*. The intrinsic dimension of $\mathbb{M}$, n, is generally much lower than D. A **tangent space** of $\mathbb{M}$ on x (denoted as $T_x\mathbb{M}$) can be defined as

$$T_x\mathbb{M} := \left\{ v \middle| \exists \text{ differentiable curve } \gamma : [-\epsilon, \epsilon] \to \mathbb{M}, \gamma(0) = x,\ s.t.\ v = \frac{d\gamma}{dt}\Big|_{t=0} \right\},$$

which is a linear subspace of $\mathbb{R}^D$. Moreover, the **normal space** on x (denoted as $N_x\mathbb{M}$) is the orthogonal complement of the tangent space $T_x\mathbb{M}$.

3.2 Gradient Leaking

We now formally present the gradient leaking hypothesis. Without loss of generality, we consider a two-class classification problem. Typically, we want to learn a prediction function $h : \mathbb{R}^D \to [0, 1]$ such that $h(x) = p(y = 1|x)$. Assuming that all the data points are on the manifold $\mathbb{M}$, then the restriction of h on $\mathbb{M}$ (denoted by $h|_{\mathbb{M}}(x)$) completely determines training loss and testing accuracy. In other words, if we have a function e defined on $\mathbb{R}^D$ such that $\forall x \in \mathbb{M}$, $e(x) = 0$, then $h + e$ shares the same training/testing statistics with h since they are the same on $\mathbb{M}$. However, the adversarial robustness of $h + e$ and h could be very different, since adversarial examples usually do not lie on the manifold $\mathbb{M}$. This is an ambiguity of the functions in $\mathbb{R}^D$, when they share the same values on $\mathbb{M}$.

Considering this ambiguity issue, we need to specify an extension of some given $h|_{\mathbb{M}}(x)$ to $\mathbb{R}^D$ for adversarial robustness. Specifically, the following is a desirable property suggesting adversarial robustness of an extension h:

$$\forall x \in \mathbb{M}, \nabla h(x) \in T_x\mathbb{M}, \tag{1}$$

which means that the prediction does not change if x is perturbed in a perpendicular direction of the manifold tangent space. Intuitively, the adversary cannot perform the attack successfully by only perturbing the input image away from the manifold. We note that the tangent component of $\nabla h(x)$ is indispensible to enable the value of $h(x)$ to vary on the manifold to classify the dataset, and hence intuitively, Eq. (1) describes an ideal case to maintain the accuracy while improving robustness.

In reality, however, the classifier is usually inclined to make its gradient nearly perpendicular to the tangent space of the data manifold. We call this phenomenon **gradient leaking**. Formally speaking, we propose *Gradient Leaking Hypothesis* as follows:

> *Let h denote the learned prediction function in typical machine learning tasks and $x \in \mathbb{M}$. If we decompose the gradient into the tangent space and normal space as $\nabla h(x) = v_{\|} + v_{\perp}$, where $v_{\|} \in T_x\mathbb{M}$ and $v_{\perp} \in N_x\mathbb{M}$, then $\|v_{\|}\| \ll \|v_{\perp}\|$.*

The hypothesis suggests that even if h is a good classifier on the manifold $\mathbb{M}$, it may fail to be robust since it puts too much of its gradient in the normal space of the manifold. An geometric description of gradient leaking is that the hypersurface $\{x : h(x) = c\}$ for $c \in (0, 1)$ is nearly parallel to the data manifold, while in the ideal case they should be perpendicular to each other. When $c = 0.5$, the hypersurface $\{x : h(x) = c\}$ is the decision boundary, so gradient leaking basically means that the decision boundary is along the data manifold.

An illustrating case when gradient leaking happens is shown in Fig. 3. The data manifold is the green sine-wave surface, in which the part above the blue

surface is in one class, and the part below the blue surface is in the other class. The blue surface itself seems a natural choice to classify the two classes since it is nearly linear, but the classifier using it as the decision boundary is not a robust classifier, since we can change its prediction by perturbing the data point up or down slightly. This aligns with our gradient leaking hypothesis since the blue surface is nearly parallel to the data manifold. A more robust decision boundary should be perpendicular to the data manifold, but it must be wave-like as well, making it more difficult to learn in practice.

Intuitively, the non-robust classifier corresponding to the blue surface utilizes the direction of fluctuation of the data manifold, which could be rather small despite correlating with the true label well. We call such directions *small-scale features*, and call their opposite, the main spanning direction of the data manifold, *large-scale features*. Similar cases of gradient leaking happens in real datasets as well, since there exist such small-scale features like textures. We assume that in the learning procedure, the classifier would rely on the most discriminative dimensions, which may be small-scale but linear separable. We perform a data poisoning experiment on CIFAR10 (see Appendix E) to verify the hypothesis that the preference for small scale features may cause the classifier to be non-robust.

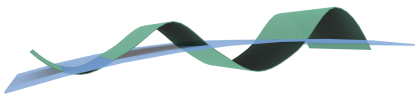


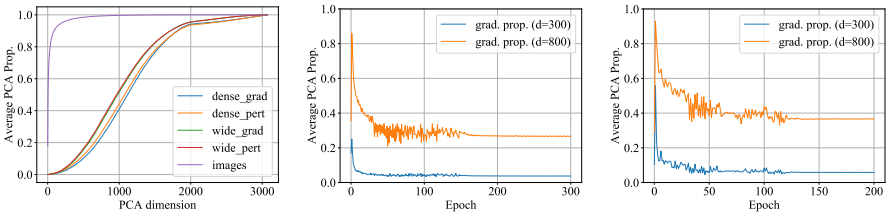
Fig. 3: Illustration of the gradient leaking phenomenon. The green surface is the data manifold (all of the data points are on it), and the label is decided by whether the data point is above or below the blue surface. However, if the blue surface is chosen as the decision boundary, then gradient leaking occurs and the classifier is not robust.

Our hypothesis can explain the limitation of the present methods. For example, the gradient regularization methods [23, 12] or Lipschitz limited methods [5, 15] assume that one could design a robust model by making the gradient norm smaller. Adversarial training [8] has shown a great performance which can reduce the gradient norm considerably. However, these methods do not distinguish between the tangent space and the normal space, and cannot preserve the useful gradient component in the tangent space while reducing that in the normal space. Our hypothesis suggests that to improve robustness, we should focus on the direction instead of the norm of the gradient. Moreover, simply increasing the number of training data points may not help much [4], because points sampled from the data manifold tell the classifier nothing about what its prediction should be outside of the manifold.

3.3 Empirical Study

In this section, we empirically show that the gradient leaking phenomenon widely exists in DNNs.

Evaluation Metric To detect the gradient leaking phenomenon in the real scenario, it is expected that we can clearly recognize the tangent space and normal



(a) $\overline{\alpha}_d$ vs. d (b) $\overline{\alpha}_d$ vs. epoch (Dense) (c) $\overline{\alpha}_d$ vs. epoch (Wide)

Fig. 4: Average PCA proportion of the gradient on CIFAR10.

space on each point at a low cost, which demands an efficient way to represent the data manifold approximately. Among various choices, in this paper, we resort to the PCA subspace since it is convenient yet effective for manifold representation. Specifically, suppose the PCA eigenvectors are $\{v_1, v_2, ..., v_D\}$ in a descending order of corresponding eigenvalues, which is an orthogonal basis of the ambient space. We refer to the d-dimensional PCA (principal) subspace as spanned by $\{v_1, v_2, ..., v_d\}$, where $d \ll D$. We define an evaluation metric of gradient leaking, with the PCA principal subspace serving as an approximation to the local tangent space. For a data point x, suppose $g = \nabla f(x)$ is the adversarial gradient where f is some loss function w.r.t. the label of x. Then a larger proportion of g in the PCA subspace indicates less gradient leaking, which can be calculated as

$$\alpha_d(g) = \frac{\sum_{i=1}^{d}(g^\top v_i)^2}{\|g\|_2^2}. \tag{2}$$

We note that for $d_1 < d_2$, $0 \le \alpha_{d_1}(g) \le \alpha_{d_2}(g) \le 1$. By drawing a curve of $\alpha_d(g)$ to $d \in \{1, 2, ..., D\}$, complete information of g can be recovered.

Existence of Gradient Leaking Phenomenon We conduct experiments on CIFAR10 ($D = 32 \times 32 \times 3 = 3072$) by training two different state-of-the-art network architectures, namely DenseNet [10] and Wide-ResNet [31]. We also conduct experiments on CASIA-WebFace, a dataset for face recognition, but leave the results to Appendix C due to space limitation. To evaluate the extent of gradient leaking, we calculate the average PCA proportion of the gradient over the dataset as[1]

$$\overline{\alpha}_d \triangleq \frac{1}{N}\sum_{n=1}^{N}\alpha_d(\nabla f(x_n)). \tag{3}$$

We show the curve of $\overline{\alpha}_d$ vs. d at the last epoch in Fig. 4(a). It can be seen that the component of gradients in the PCA principal subspace is far less than the component of images in the same subspace, indicating that the model relies on small-scale features to classify, i.e. leaks gradient outside of the data manifold.

[1] We omit the dependence of the loss function f on the label of x_n.

Table 1: Statistics of 8 pretrained models on ImageNet.

Model	inc	res50	res152	vgg	ens	hgd	iat	iat-den	R^2
Accuracy	0.769	0.740	0.751	0.694	0.764	0.787	0.602	0.639	
1/Mean grad norm	0.115	0.197	0.157	0.272	0.264	0.248	1.535	1.743	0.904
$\overline{\alpha}_{400}$	0.011	0.020	0.021	0.018	0.037	0.014	0.145	0.152	**0.940**
$\overline{\alpha}_{1241}$	0.053	0.102	0.103	0.096	0.146	0.060	0.284	0.296	0.878
$\overline{\alpha}_{6351}$	0.333	0.449	0.465	0.628	0.559	0.314	0.588	0.612	0.319
Mean pert norm	0.199	0.190	0.306	0.262	1.084	0.540	1.952	2.332	-

Meanwhile, we also explore the property of the adversarial perturbation direction here. We try to perturb the original input x such that the perturbed sample is misclassified, and find such smallest ℓ_2-perturbation $\delta(x)$ by the ℓ_2-BIM attack [14] implemented by Foolbox [21]. Then we calculate $\overline{\alpha}_d$ for $d = \{1, 2, ..., D\}$ in the same way except that $\nabla f(x_n)$ is replaced by $\delta(x_n)$, and draw the curve in the same figure. The PCA proportion of the gradient and the adversarial perturbation is almost identical, suggesting that the direction of adversarial samples could be seen as a first-order effect[2], hence reducing the gradient leaking phenomenon at data points should be a good option for improving the robustness.

Furthermore, we explore the trends of gradient leaking along the training process. For the training model, we plot $\overline{\alpha}_d$ to the training epochs for $d = 300$ and 800 in Fig. 4(b) on DenseNet and in Fig. 4(c) on Wide-ResNet[3]. At Epoch 0 (before training), gradient leaking is severe since the model has no knowledge of the data manifold then. At Epoch 1, the model leaks the least proportion of gradients since it learns to classify the data points in the data manifold during the first epoch. However, the gradient leaking becomes more and more notable in the remaining epochs. This validates our conjecture that small-scale features might be preferred by DNNs when they are discriminative and easy for classification. In the beginning of training, the models discover the data manifold and learn to classify with the large-scale features along the manifold. However, DNNs would finally discover such small-scale features and use them to improve classification performance at the expense of robustness.

Gradient Leaking as an Indicator of Robustness Having established that gradient leaking phenomenon exists when training typical neural architectures, we empirically study the relationship between the robustness and the tensity of gradient leaking. We note that the norm of the gradient might not be the best indicator of robustness, since that for example, an image with a rather small gradient norm could also have an adversarial example in its neighborhood.

We analyze 8 models trained on ImageNet, in which 4 are normally trained models ('inc': Inception-v3, 'vgg': VGG-16, 'res50': ResNet-v1-50, 'res152': ResNet-v2-152) and 4 are models which are intended to be robust ('ens': Inception-

[2] Though the distance may be affected by the non-linearity introduced by softmax.

[3] In CIFAR10, PCA with $d = 300$ and 800 preserve 96.85% and 99.40% of the energy of the image dataset respectively.

ResNet-v2 through ensemble adversarial training [28], 'hgd': An ensemble of networks with a high-level representation guided denoiser [16], 'iat': a ResNet-152 model through large-scale adversarial training [29], 'iat-den': the 'iat' model with feature-denoising layers [29]). We conduct the experiments on the first 1000 images in the validation set. The results are shown in Table 1. The d in $\overline{\alpha}_d$ is chosen as 400, 1241 and 6351 since they correspond to preserving 90%, 95% and 99% of the energy respectively after the images are projected to the PCA subspace. In the last column, we show the coefficient of determination R^2 of the linear regression which predicts the mean perturbation norm of the adversarial example we find by ℓ_2-BIM (the last row). Note that according to Fig. 1, the perturbation norm of adversarial examples should be approximately proportional to $\sqrt{\overline{\alpha}_d}$. We hence show R^2 w.r.t. $\sqrt{\overline{\alpha}_d}$ instead of $\overline{\alpha}_d$ here. We find that compared with the reciprocal of mean gradient norm, $\overline{\alpha}_{400}$ turns out to be a better indicator of the mean perturbation norm which represents adversarial robustness.[4]

Discussion From the validation in the real scenario, we find that

1. Gradient leaking widely exists in DNNs. Considering that the PCA subspace can be much larger than the real data manifold, it might be worse than we already observed and be a reason for adversarial vulnerability.
2. Adversarial vulnerability is a first order phenomenon in the sense that the adversarial perturbation direction aligns with the gradient direction well.
3. During the training procedure, the gradients concentrate on main components at first, and then leak gradually, showing a clear dynamics of changing the gradients' direction to fit the small-scale features.
4. Small-scale features are preferred by DNNs. They could even generalize well on the testing dataset, but we need to reduce their effect in robustness-sensitive tasks.

4 Adversarial Defenses

With our discussion above, making gradients lie in the tangent space of data manifold might be a central mission to construct a robust classifier. Hence, we propose to improve robustness by suppressing the gradient leaking phenomenon.

4.1 Making the Data Manifold Flat

To deal with gradient leaking, we consider modifying the training dataset to make the data manifold 'flat'. Taking Fig. 3 as an example, we propose to project the data manifold onto the blue surface, making the currently shown decision boundary invalid. It forces a model with sufficient expressive power to learn to classify with a more robust feature such as the coordinate along the blue surface (although the decision boundary could be more complicated).

[4] R^2 w.r.t. $\sqrt{\overline{\alpha}_{1241}}$ and $\sqrt{\overline{\alpha}_{6351}}$ deteriorate perhaps because the intrinsic dimension of the data manifold should be much smaller than 1241.

Specifically, we propose to project the training dataset to its PCA principal subspace before training, so that the fluctuation of data manifold is partially eliminated. Note that in evaluation (Sec. 3.3), we consider the data manifold *as* the PCA hyperplane since they are similar in the large-scale stretching direction; however, they are very different in the training process as we mentioned above. Formally speaking, we project each data point x as[5]

$$x \leftarrow \sum_{i=1}^{d} \langle x, v_i \rangle v_i,$$

before training, where d is a hyperparameter representing the dimension of the PCA subspace, and $v_1, v_2, ..., v_d$ are the d principal eigenvectors. We note that we perform PCA projection during training process instead of during testing process, which suffices to improve robustness significantly. The time cost of computation of PCA principal eigenvectors is relatively small (see Appendix A).

4.2 Adding Noise in the Normal Space

A more direct way to suppress gradient leaking is to perform data augmentation so that the loss of the classifier would be large if the decision boundary is not perpendicular enough to the data manifold. The idea is best illustrated in Fig. 2. Fig. 2(a) shows the original data distribution in which regions of two different colors represent two categories of data points. We recognize the black parabolic curve as the 1-D approximate data manifold. Fig. 2(b) (approximately) shows that by adding noise independent of the label in the normal direction of the data manifold , the augmented data can force the classifier to learn a decision boundary that is perpendicular to the data manifold.

Adding noise in the normal space is in contrast to the previously introduced idea of flattening the data manifold. In Sec. 4.1, we actually did not impose constraints upon the classifier, but made it easier for it to learn the robust decision boundary. Each of the two methods can be independently applied in theory. However, it is difficult to access the tangent space or the normal space in a general data manifold. But if we combine the two methods together, thanks to the fact that the dataset has been projected into a PCA hyperplane, we can access (a subspace of) the normal space easily by identifying it as the space spanned by remaining PCA eigenvectors orthogonal to the principal hyperplane.

Specifically, utilizing the PCA basis and combining with the method in Sec. 4.1, we modify the training data x as

$$x \leftarrow \sum_{i=1}^{d} \langle x, v_i \rangle v_i + \sum_{i=d+1}^{D} \sigma_i \xi_i v_i,$$

where $\xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and $\{\sigma_i\}_{i=d+1}^{D}$ is a set of hyperparameters which could be set in a principled way. In this view, by adding label-irrelevant noise, we make

[5] For clarity, we assume the dataset has been centralized so that $\overline{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$ is 0.

Algorithm 1 Modifying training data for training robust models

Input: A training data point x; with the $N \times D$ training dataset $\mathbf{X} = [x_1, ..., x_N]^\top$, dimension d of PCA subspace, dimension m of the subspace to add noise, noise scale $c > 0$.

Output: Modified training data point x'.

1: Perform spectral decomposition on the covariance matrix $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \overline{x})(x_n - \overline{x})^\top$ (where $\overline{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$) as
$\mathbf{C} = \sum_{i=1}^{D} \lambda_i v_i v_i^\top, \quad \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D$;

2: Compute the components of $x - \overline{x}$ on $V = [v_1, v_2, ..., v_D]$ as
$a \leftarrow V^\top (x - \overline{x})$;

3: Projecting x to the PCA subspace:
For the ith component of a: $a_i \leftarrow 0, \quad \text{for} \quad i = d+1, d+2, ..., D$;

4: Add noise:
$a_i \leftarrow c\sqrt{\lambda_i}\xi_i, \quad \text{with} \quad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1) \quad \text{for} \quad i = d+1, d+2, ..., d+m$;

5: Reconstruction:
$x' \leftarrow Va + \overline{x}$;

6: **return** x'.

the small-scale directions, i.e. $\{v_{d+1}, ..., v_D\}$ hardly utilized by the model for classification even if the decision boundary is highly non-linear, thus suppressing gradient leaking.

Contrast to former robust learning methods by randomization like [9] in which authors add isotropic Gaussian noise to weights or inputs of each layer, our method augments the dataset with Gaussian noise in the direction of PCA eigenvectors and in the specific subspace of small-scale features. To maximize the efficiency of injected noises, for a small value of integer m (e.g. $m = 10$), we set $\sigma_i = 0$ for all $i > d + m$ while setting σ_i to be a relatively large value for $d + 1 \leq i \leq d + m$. With a subspace of lower dimension, the efficiency of noise sampling to cover the space could be much higher. Meanwhile, we find that this not only reduces the gradient component in the subspace spanned by $\{v_{d+1}, ..., v_{d+m}\}$ but also reduces the gradient components along other eigenvectors with small eigenvalues. A possible explanation is that by preventing the convolution kernel from being activated by some patterns, it will also drop other similar features (e.g. of similar frequency).

To summarize, we present our algorithm to preprocess the training data in Algorithm 1. Note that for $d + 1 \leq i \leq d + m$, we set $\sigma_i = c\sqrt{\lambda_i}$ which means that the scale of the ith dimension of the modified training dataset is c times larger than before.

5 Experiments

5.1 Primary Experiments

In this section, we apply our defense algorithm to improve the robustness upon the baseline training algorithm. We first present the experimental setup, and then report the quantitative results to demonstrate the effectiveness of our algorithm.

Experiment Setup
Dataset We test our algorithms on two datasets, namely CIFAR10 (shown here) and CASIA-WebFace (see Appendix D).
Backbone models Same as in Sec. 3.3, namely DenseNet and Wide-ResNet.
Metric We report the testing error rate on clean data (Err), the mean/median perturbation norm (Pert) that represents the robustness (higher is better), the mean gradient norm $\|g\|_2$ (Grad) (lower is better) and $\overline{\alpha}_d$ defined in Eq. (3) as relevant quantities of robustness (higher is better).
Attack method We perform ℓ_2-C&W attack [3] implemented in Foolbox 2.3.0 with default parameters, which is the strongest attack among the ones (including PGD [17], DDN [22], BIM and C&W) we have tested in Foolbox.

Experimental Results We compare three different types of training methods, namely the ordinary training method ('ord'), our proposed algorithm ('noise') (with hyperparameters $c = 10$ and $m = 10$), and a degenerated version of our algorithm ('pca') by skipping the step of adding noise, i.e. Algorithm 1 without Line 4. In the testing phase (including robustness evaluation), we directly feed the original test image (without any preprocessing) into the trained models.

We report the experimental results in Table 2. The subscript of each method name refers to the dimension of PCA subspace d. With a suppression of utilization of small-scale features, our method consistently outperforms the ordinary models in terms of the robustness although there are different degrees of accuracy degeneration for clean images. It, from another side, provides an evidence that the robust features are more difficult to learn and may not be that discriminative. To compare with the state-of-the-art method, we train TRADES [32], an adversarial-training method, on the two architectures and report the results in the table. Although there remains a gap of robustness (perturbation norm) between our method and TRADES, our improvement of robustness has been significant, with controllable and less deterioration of accuracy on clean data than TRADES. Moreover, in Sec. 5.2 we will propose a stronger defense by integrating our data preprocessing procedure into other defense algorithms.

The results also show that $\overline{\alpha}_d$ becomes larger as we reduce the dimension of d and add noise in the normal space, which is highly related to the improvement

Table 2: CIFAR10 results of DenseNet and Wide-ResNet. The mean/median perturbation norm are in the 'Pert' column.

Method	DenseNet					Wide-ResNet				
	Err	Grad	$\overline{\alpha}_{300}$	$\overline{\alpha}_{800}$	Pert	Err	Grad	$\overline{\alpha}_{300}$	$\overline{\alpha}_{800}$	Pert
ord	**5.92**	0.248	0.040	0.273	0.090/0.085	**9.08**	0.195	0.056	0.352	0.113/0.100
pca_{800}	6.97	0.123	0.244	0.817	0.160/0.155	9.7	0.091	0.253	0.845	0.193/0.178
noise_{800}	8.82	0.140	0.460	0.988	0.208/0.192	10.49	0.076	0.512	**0.978**	0.251/0.233
pca_{300}	11.71	0.075	0.713	0.973	0.256/0.240	13.75	0.050	0.667	0.910	0.308/0.283
noise_{300}	16.53	**0.060**	**0.942**	**0.988**	**0.308/0.265**	19.09	**0.027**	**0.843**	0.881	**0.320/0.299**
trades	21.22	0.009	0.416	0.639	0.664/0.579	19.37	0.008	0.446	0.673	0.725/0.639

of robustness. They also provide a strong evidence that for normally trained models, the gradient component leaks in the normal space. Our results are in line with our theory that the projection of the gradient on the manifold is a more essential attribute of the classification function. For instance, the DenseNet model trained by noise_{800} method has a higher gradient norm than pca_{800} which should imply less robustness, but actually its average perturbation distance is higher than pca_{800}. This, however, aligns with the improvement of $\overline{\alpha}_{300}$ and $\overline{\alpha}_{800}$. The contradictory phenomenon cannot be explained without the insight of gradient leaking that the additional Gaussian noise on the normal space further suppresses gradient leaking. We also note that the $\overline{\alpha}_d$ value of TRADES, the most robust model among those listed in the table, is relatively small compared with $\overline{\alpha}_d$ of our methods. This is perhaps because our perspective of gradient leaking cannot address the issue of in-manifold robustness, and also because our PCA approximation of the data manifold is rather rough. Nevertheless, less gradient leaking still correlates with and promotes stronger robustness well.

5.2 Integration into Other Defense Algorithms

Our method is light-weight and can be naturally integrated with other defense methods. To further boost the robustness, we provide an exemplary result by integrating it with a recently proposed method of Max-Mahalanobis center (MMC) loss [19]. Roughly speaking, it proposes to replace the softmax cross-entropy (SCE) loss with MMC loss which acts upon the layer just before the logits layer.

To further reduce gradient leaking, we preprocess our training data according to our algorithm before feeding them into the MMC training process, resulting in a even stronger defense denoted as MMC-P. We conduct experiments on CIFAR10 in which we simply project the dataset to 500/300-dimensional PCA subspace as the preprocessing procedure. We report results of MMC and MMC-P in Table 3. The robustness is evaluated using ℓ_∞ PGD attacks (see Appendix B for evaluation under ℓ_2 attacks) with different settings (targeted/untargeted, 10/50 iteration steps), and we show the natural accuracy ($\epsilon = 0$) and the robustness accuracy under attacks with different ℓ_∞ perturbation bounds $\epsilon = 8, 16, 24, 32$. In PGD_{10}, we adopt the step size $2, 3, 4, 5$ respectively for $\epsilon = 8, 16, 24, 32$; in PGD_{50}, we set the step size to 2 in all cases. We utilize the C&W loss [3] (instead of the ordinary cross-entropy loss) which constitutes a stronger attack, so the robustness accuracy we report is lower than that in [19].

We found despite that the MMC baseline is satisfactory, our proposed MMC-P outperforms the vanilla MMC by a large margin with a simple data preprocessing step. To compare with the state-of-the-art methods, we report the performance of MMC-AT (adversarial training (AT) [17] using MMC loss) proposed in [19], which is adversarially trained using 10-step targeted PGD attack under perturbation bound $\epsilon = 8$. We note that it is stronger than vanilla AT (i.e. SCE-AT in the table). Experimental results show that despite the remaining gap of robustness between MMC-P and MMC-AT, MMC-P is still a competitive defense compared with the state-of-the-art AT methods since it brings a satisfactory robustness performance with less sacrifice in accuracy on clean data. Meanwhile,

Table 3: The experimental results using MMC loss on CIFAR10.

Attack	Training Method	Clean	Untargeted				Targeted			
		$\epsilon = 0$	8	16	24	32	8	16	24	32
PGD_{10}	SCE	93.5	3.9	2.9	2.2	1.9	0.0	0.0	0.0	0.0
	MMC	92.5	25.6	11.7	5.9	3.7	45.5	29.2	20.2	14.2
	MMC-P-500	90.3	42.5	30.6	21.4	15.9	57.9	46.5	37.6	30.4
	MMC-P-300	87.9	43.9	33.4	25.5	20.6	56.5	46.6	38.9	32.7
	SCE-AT	84.0	50.5	20.9	11.2	8.7	68.2	36.5	14.6	4.3
	MMC-AT	83.3	54.2	40.1	35.1	30.8	63.2	50.8	44.2	39.3
PGD_{50}	SCE	93.5	3.8	3.1	2.3	1.8	0.0	0.0	0.0	0.0
	MMC	92.5	9.2	4.8	3.3	2.2	26.9	16.6	12.7	9.5
	MMC-P-500	90.3	24.9	15.1	11.3	9.1	41.4	30.3	26.6	22.9
	MMC-P-300	87.9	29.2	20.1	17.1	14.8	41.0	30.9	27.9	24.7
	SCE-AT	84.0	48.9	17.4	9.6	8.2	66.6	28.0	6.0	1.0
	MMC-AT	83.3	51.1	35.4	31.2	27.8	60.9	44.8	40.2	35.7

MMC-P is more convenient to use, and much faster to train. The results demonstrate that our method can further boost the robustness performance of some well-developed methods by integrating with them naturally.

6 Conclusion and Future Work

We reveals a possible path named "gradient leaking" to explain the existence and properties of adversarial samples. We further develop a method to examine the gradient leaking phenomenon, analyze its relationship with existence of adversarial samples, and further propose a novel method to defend against adversarial attacks based on the hypothesis, which adopts the linear dimension reduction and randomization technique before training. It brings an explainable robustness improvement with little extra time cost.

In the future, the mechanism of gradient leakage still requires a theoretical explanation, which may include the aspects of learning methods, data distribution and network architecture. A better data manifold representation method from which the local tangent space can be identified, such as VAE and localized GAN as suggested in [30], may lead us to deeper understanding of data manifold, more accurate estimate of gradient leaking, and more impressive robustness improvement. Combining our methods with other defense algorithms and figuring out how they work together could also be a potential direction.

Acknowledgements This work was supported by the National Key Research and Development Program of China (No. 2017YFA0700904), NSFC Projects (Nos. 61620106010, U19B2034, U1811461, U19A2081, 61673241), Tsinghua-Huawei Joint Research Program, a grant from Tsinghua Institute for Guo Qiang, Beijing Academy of Artificial Intelligence (BAAI), Tiangong Institute for Intelligent Computing, the JP Morgan Faculty Research Program, and the NVIDIA NVAIL Program with GPU/DGX Acceleration.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
2. Bubeck, S., Price, E., Razenshteyn, I.: Adversarial examples from computational constraints. arXiv preprint arXiv:1805.10204 (2018)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
4. Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., Duchi, J.C.: Unlabeled data improves adversarial robustness. arXiv preprint arXiv:1905.13736 (2019)
5. Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., Usunier, N.: Parseval networks: Improving robustness to adversarial examples. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 854–863. JMLR. org (2017)
6. Dahl, G.E., Stokes, J.W., Deng, L., Yu, D.: Large-scale malware classification using random projections and neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 3422–3426. IEEE (2013)
7. Fawzi, A., Moosavi-Dezfooli, S.M., Frossard, P., Soatto, S.: Empirical study of the topology and geometry of deep networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
9. He, Z., Rakin, A.S., Fan, D.: Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 588–597 (2019)
10. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
11. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. arXiv preprint arXiv:1905.02175 (2019)
12. Jakubovitz, D., Giryes, R.: Improving dnn robustness to adversarial attacks using jacobian regularization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 514–529 (2018)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
14. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
15. Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R., Jacobsen, J.H.: Preventing gradient attenuation in lipschitz constrained convolutional networks. arXiv preprint arXiv:1911.00937 (2019)
16. Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J.: Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1778–1787 (2018)

17. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
19. Pang, T., Xu, K., Dong, Y., Du, C., Chen, N., Zhu, J.: Rethinking softmax cross-entropy loss for adversarial robustness. arXiv preprint arXiv:1905.10626 (2019)
20. Qian, H., Wegman, M.N.: L2-nonexpansive neural networks. arXiv preprint arXiv:1802.07896 (2018)
21. Rauber, J., Brendel, W., Bethge, M.: Foolbox: A python toolbox to benchmark the robustness of machine learning models. arXiv preprint arXiv:1707.04131 (2017), http://arxiv.org/abs/1707.04131
22. Rony, J., Hafemann, L.G., Oliveira, L.S., Ayed, I.B., Sabourin, R., Granger, E.: Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4322–4330 (2019)
23. Ross, A.S., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Thirty-second AAAI conference on artificial intelligence (2018)
24. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605 (2018)
25. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially robust generalization requires more data. In: Advances in Neural Information Processing Systems. pp. 5014–5026 (2018)
26. Shamir, A., Safran, I., Ronen, E., Dunkelman, O.: A simple explanation for the existence of adversarial examples with small hamming distance. arXiv preprint arXiv:1901.10861 (2019)
27. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
28. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
29. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
30. Yu, B., Wu, J., Ma, J., Zhu, Z.: Tangent-normal adversarial regularization for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10676–10684 (2019)
31. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
32. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 (2019)