Supplementary Material for: 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, State Key Lab for Intell. Tech. & Sys., Institute for AI, THBI Lab, Tsinghua University, Beijing, 100084, China
{liyr18, chengsy18}@mails.tsinghua.edu.cn,
{suhangss, dcszj}@mail.tsinghua.edu.cn

A Remark on Time Cost of PCA

The extra cost in our algorithm mainly comes from computation of PCA on the dataset. PCA on CIFAR10 is very fast. PCA on datasets with higher dimension would be slower, but thanks to recently developed modern randomized algorithm such as [1, 3, 5], the time complexity of approximate PCA could be reduced from $O(D^3)$ to $O(D \cdot d^2)$, where D is the order of matrix and d is the dimension of principal subspace. We find that PCA for ImageNet can be computed within several hours if we resize the images to 112×112 (the approximation does not affect much), and PCA for CASIA is faster. Another good news is that for a dataset we only need to conduct PCA once and reuse in multiple runs. Meanwhile, the PCA results can be shared online. Therefore, our algorithm brings little extra cost to training procedure.

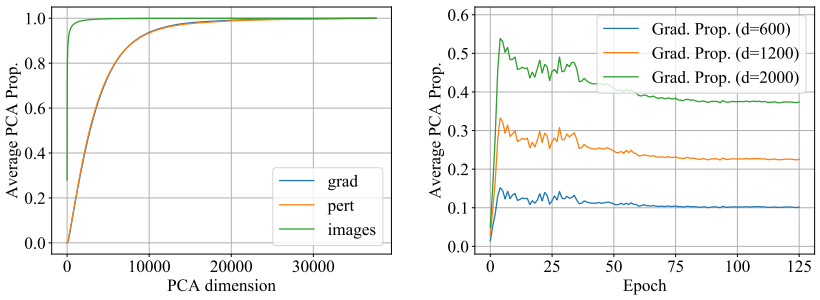
B Evaluation under ℓ_2 Attack in Sec. 5.2

In this section, we evaluate the robustness of models in Table 3 under ℓ_2 attack. We apply the ℓ_2-BIM attack [2] with 10/50 iterations under perturbation bounds $\epsilon = 1 \times 255, 2 \times 255, 3 \times 255$. We use step size $0.25 \times 255, 0.375 \times 255, 0.5 \times 255$ respectively for $\epsilon = 1, 2, 3$ in BIM_{10}, and use step size 0.25×255 for all ϵ in BIM_{50}. We note that the AT-based models here we used are still trained against ℓ_∞ attacks, but they are still rather robust against ℓ_2 attacks [4].

We show the results in Table 4. The results are similar to the case under ℓ_∞ attacks, but we can see that relatively our proposed MMC-P method performs better than in Table 3. For example, in BIM_{10} targeted attacks, the performance of MMC-P is close to or slightly surpasses that of AT-based methods. A possible explanation is that our theoretical framework of gradient leaking is more suited to the ℓ_2 threat model.

Attack	Training Method	Clean	Untargeted			Targeted		
		$\epsilon = 0$	1	2	3	1	2	3
BIM_{10}	SCE	93.5	4.6	4.6	4.5	0.0	0.0	0.0
	MMC	92.5	30.9	21.7	15.9	47.6	37.9	30.3
	MMC-P-500	90.3	43.4	35.6	29.5	56.4	49.9	43.9
	MMC-P-300	87.9	43.3	35.5	30.4	54.0	47.5	42.3
	SCE-AT	84.0	38.3	13.4	9.9	59.3	19.4	4.4
	MMC-AT	83.3	46.9	37.2	34.8	56.9	46.1	41.9
BIM_{50}	SCE	93.5	4.6	4.6	4.6	0.0	0.0	0.0
	MMC	92.5	10.4	8.4	8.2	25.4	20.0	20.0
	MMC-P-500	90.3	24.8	16.0	15.3	39.2	30.8	28.9
	MMC-P-300	87.9	28.8	21.0	19.7	39.3	30.6	29.2
	SCE-AT	84.0	35.1	11.6	9.5	56.5	12.2	1.0
	MMC-AT	83.3	42.8	33.0	30.6	54.3	38.9	36.9

Table 4: The experimental results using MMC loss under ℓ_2 attacks on CIFAR10.



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

Fig. 5: Average PCA proportion of the gradient on CASIA.

C Results on CASIA-WebFace Corresponding to Fig. 4

In this section, we show the experimental results corresponding to Fig. 4 on the CASIA-WebFace dataset. CASIA-WebFace (abbreviated as CASIA below) is a dataset for face recognition, and we believe it has rather different properties of data manifold from that of CIFAR10. Its cleaned version includes 455,594 images with 10,575 classes. In our experiments, each image is preprocessed to $112 \times 112 \times 3$, and hence $D = 37632$. We adopt ResNet-50 as the backbone model with the ordinary linear head layer. We show the curve of $\overline{\alpha}_d$ vs. d at the last epoch in Fig. 5(a), and the curve of $\overline{\alpha}_d$ vs. the training epochs for $d = 600$, 1200 and 2000 in Fig. 5(b)[1].

From Fig. 5(a), we can see that the result of CASIA is similar to that of CIFAR10, in the sense that the component of gradients in the PCA principal subspace (denoted 'grad' in the figure legend) is far less than the component of images in the same subspace (denoted 'images' in the legend), and the PCA components of the gradient ('grad') and the adversarial perturbation ('pert') is almost identical.

The analysis on Fig. 5(b) is similar to the case of CIFAR10. The main difference is that on CASIA, $\overline{\alpha}_d$ keeps increasing for several epochs (instead of only one epoch on CIFAR10) before decreasing. It suggests that the model takes a longer time to complete the "burn-in" process of classification since the CASIA dataset is much larger than CIFAR10. In the remaining epochs, $\overline{\alpha}_d$ keeps decreasing, which is consistent with the results of CIFAR10, implying that further training aggravates gradient leaking.

D Results on CASIA-WebFace Corresponding to Table 2

In this section, we show the experimental results corresponding to Table 2 on the CASIA-WebFace dataset. The experimental settings are the same as in Sec. C (dataset and model) and in Sec. 5.1 (training and attack method), except that we adopt $m = 15$ here for the 'noise' method, and we did not perform adversarial training for comparison due to the large scale of CASIA dataset. The results are shown in Table 5, which are consistent with the results in Table 2 and prove the effectiveness of our proposed defense.

E The Data Poisoning Experiment

In Section 3.2, we mentioned that we performed a data poisoning experiment on CIFAR10 to verify the hypothesis: In the learning procedure, the classifier would rely on the most discriminative dimensions which may be small-scale but

*Equal contribution. †Corresponding author.

[1] In CASIA, PCA with $d = 600$, 1200 and 2000 preserve 96.56%, 98.22% and 99.01% of the energy of the image dataset respectively.

Table 5: CASIA results of the ResNet-50 model. We show the mean/median perturbation norm in the 'Pert' column.

Method	Err (%)	Grad	$\overline{\alpha}_{600}$	$\overline{\alpha}_{1200}$	Pert
ord	**14.86**	0.061	0.102	0.226	0.623/0.618
pca_{2000}	17.67	0.041	0.363	0.656	0.997/0.989
noise_{2000}	18.53	0.037	0.448	0.728	1.119/1.115
pca_{1200}	20.94	0.034	0.529	0.756	1.239/1.205
noise_{1200}	22.87	**0.033**	**0.633**	**0.791**	**1.410/1.401**

linear separable. In this section we introduce the design of the data poisoning experiment and show the results.

We choose the PCA eigenvectors as the basis in the image space, and refer to the ith dimension as the direction of the eigenvector with the ith largest eigenvalue. For each data point we erase the components of the ith dimension for all $i \geq 800$, and set the component of the $(800 + y)$th dimension to c where y is the label of the data point, and c is chosen such that the variance of the $(800 + y)$th dimension after such modification is the same as before. Basically, we manually create some small-scale features that is strongly correlated with the label. Fig. 6 shows the curve of $\overline{\alpha}_d$ versus d in the data poisoning setting. We find that apart from utilizing the created features, the neural networks ignore the large-scale features of the first 800 dimensions nearly completely.

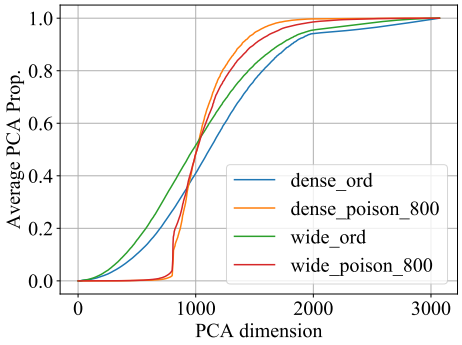
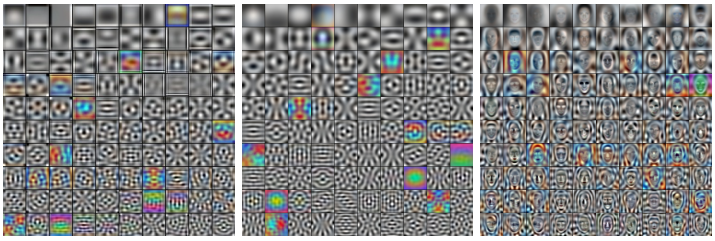


Fig. 6: CIFAR10 PCA proportion at Epoch 190 in the data poisoning experiment, where 'dense' and 'wide' refers to two network architectures, 'ord' refers to ordinary training and 'poison_800' refers to the method used in the data poisoning experiment. We could see that the gradient proportion has a step-up at dimension 800.

F Visualization of the PCA Subspace

F.1 Visualization of the Eigenvectors

To show what the large-scale features and small-scale features are like, we visualize the PCA eigenvectors corresponding to the 100 largest eigenvalues on the three datasets (CIFAR10, ImageNet and CASIA) mentioned in the paper (CASIA is an abbreviation for CASIA-WebFace). We can see some interesting properties among them. For example, roughly speaking, the scale in the dataset (i.e., eigenvalue) of each feature is negatively related to its frequency when seen as an image.

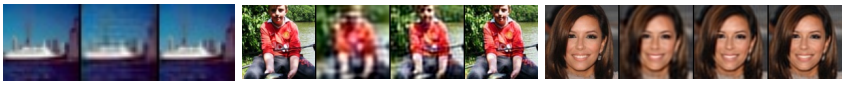


(a) CIFAR10 (b) ImageNet (c) CASIA

Fig. 7: Visualization of PCA eigenvectors corresponding to the 100 largest eigenvalues.

F.2 Visualization of the Reconstruction Results

The subspace spanned by $\{v_1, v_2, ..., v_d\}$ is refered to as the PCA (principal) subspace. The projection of x to the PCA subspace is $\sum_{i=1}^{d} \langle x, v_i \rangle v_i$, which is called its reconstruction. We show an example of the reconstruction result for each of the three datasets (CIFAR10, ImageNet and CASIA) in Fig. 8 for some choices of d. With a relatively small d such that $d \ll D$, the reconstruction could be rather close to the original image.



(a) CIFAR10 (b) ImageNet (c) CASIA

Fig. 8: Reconstruction results, (a) a CIFAR10 image example and its 300, 800 dim main component reconstruction, (b) an ImageNet example and its 400, 1241, 6351 dim reconstruction, (c) a CASIA example and its 600, 1200, 2000 dim reconstruction.

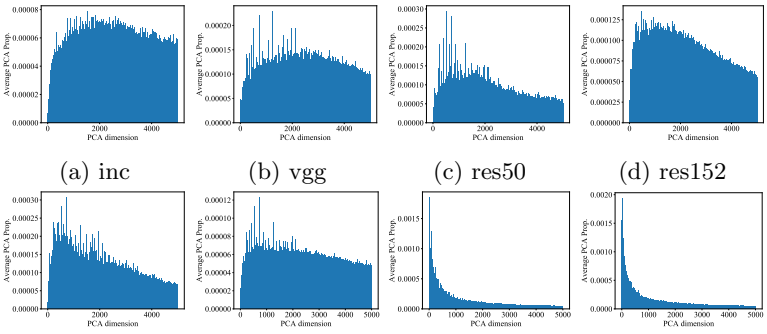
G More details on the PCA components of ImageNet model gradients

Fig. 9 shows more details regarding the gradients of pretrained models on ImageNet listed in Table 1. Specifically, we define

$$\beta_d(g) = \frac{(g^\top v_d)^2}{\|g\|_2^2} = \alpha_d(g) - \alpha_{d-1}(g)$$

as the (normalized) component of g along the dth eigenvector (sorted according to descending eigenvalues). Similarly, we define the average gradient component over the dataset as $\overline{\beta}_d = \overline{\alpha}_d - \overline{\alpha}_{d-1}$. Then a larger $\overline{\beta}_d$ shows the preference of the model for the corresponding feature (the dth eigenvector). Fig. 9 plots the value of $\overline{\beta}_d$ versus d for the 8 models. We only plot the first 5000 components for clarity; $\overline{\beta}_d$ for $d > 5000$ decreases when d increases.

We can see that for most of the models, as d increases, $\overline{\beta}_d$ first increases before decreasing (the same phenomenon occurs in CIFAR10 and CASIA). This is an intriguing property of neural networks, since by Fig. 7, the features along v_d when $d \sim 100$ should be rather useful for classification, but in fact they are not well utilized by the model. Such phenomenon is another evidence supporting our gradient leaking conjecture in the sense that the classification model tends to underutilize the features along the data manifold. By contrast, the 'iat' and 'iat-den' models provided by the authors of [6], which have experienced sufficient adversarial training and are the most robust among the 8 ImageNet models, show different properties in terms of gradient components. Basically, for these two models, $\overline{\beta}_d$ monotonically decreases as d increases, which suggests that little gradient leaking is one of the properties of an ideal robust model.



(a) inc (b) vgg (c) res50 (d) res152

(e) ens (f) hgd (g) iat (h) iat-den

Fig. 9: PCA components for different models on ImageNet. To measure the relative scale of how a gradient vector lies in or out of specific linear subspace, we square its inner product with basis vector.

References

1. Kuczyński, J., Woźniakowski, H.: Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. SIAM journal on matrix analysis and applications **13**(4), 1094–1122 (1992)
2. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
3. Liberty, E., Woolfe, F., Martinsson, P.G., Rokhlin, V., Tygert, M.: Randomized algorithms for the low-rank approximation of matrices. Proceedings of the National Academy of Sciences **104**(51), 20167–20172 (2007)
4. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
5. Woolfe, F., Liberty, E., Rokhlin, V., Tygert, M.: A fast randomized algorithm for the approximation of matrices. Applied and Computational Harmonic Analysis **25**(3), 335–366 (2008)
6. 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)