APRIL: Finding the Achilles' Heel on Privacy for Vision Transformers

Jiahao Lu[1,2], Xi Sheryl Zhang[1], Tianli Zhao[1,2], Xiangyu He[1,2], Jian Cheng[1]
[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
{lujiahao2019, xi.zhang, zhaotianli2019}@ia.ac.cn;{xiangyu.he, jcheng}@nlpr.ia.ac.cn

A. Does twin data come from lacking of parameters involved in gradient matching?

In Sec. 4.2 and Fig.7 we demonstrate and show that *twin data* can emerge from gradient attacks after disabling learnable position embedding. Formally, a *twin data* appears when a privacy attacker performs gradient matching in a Vision Transformer without matching gradients from position embedding. Some may **doubt that twin data may come from lack of parameters involved in gradient matching**. We give explicit clues that this is not the case.

We demonstrate this by using examples on MNIST. We use Architecture B (as illustrated in Fig. 1(B)) with encoder depth 4, hidden dimension of 384. So the overall parameters involved in gradient matching w/ or w/o position embedding are set out in the following table:

w/ pos_emb	w/o pos_emb
7,123,210	7,116,682

Table 2. Parameters involved in gradient matching, using Fig.1 Architecture (B) for MNIST, encoder depth = 4.

Here we make a more radical assumption: if we still use position embedding and *twin data* do not occur, when involving much less parameters than in the case without position embedding, then we can empirically prove that "twin data" do not come from lacking of model parameters.

For comparisons, we respectively disable the matching of gradients from a specific encoder layer, denoted as *encoder 1*, *encoder 2*, *encoder 3* and *encoder 4*. In these four settings, the overall parameters involved in gradient matching are 6,533,002, much fewer than the case when we disable position embedding, which has 7,123,210 parameters involved.

From Fig.9 we observe that: *twin data* do not come from lacking of parameters involved in optimization.

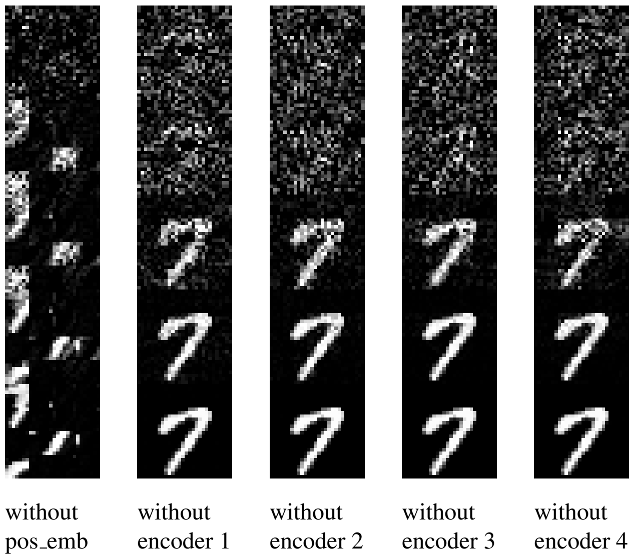


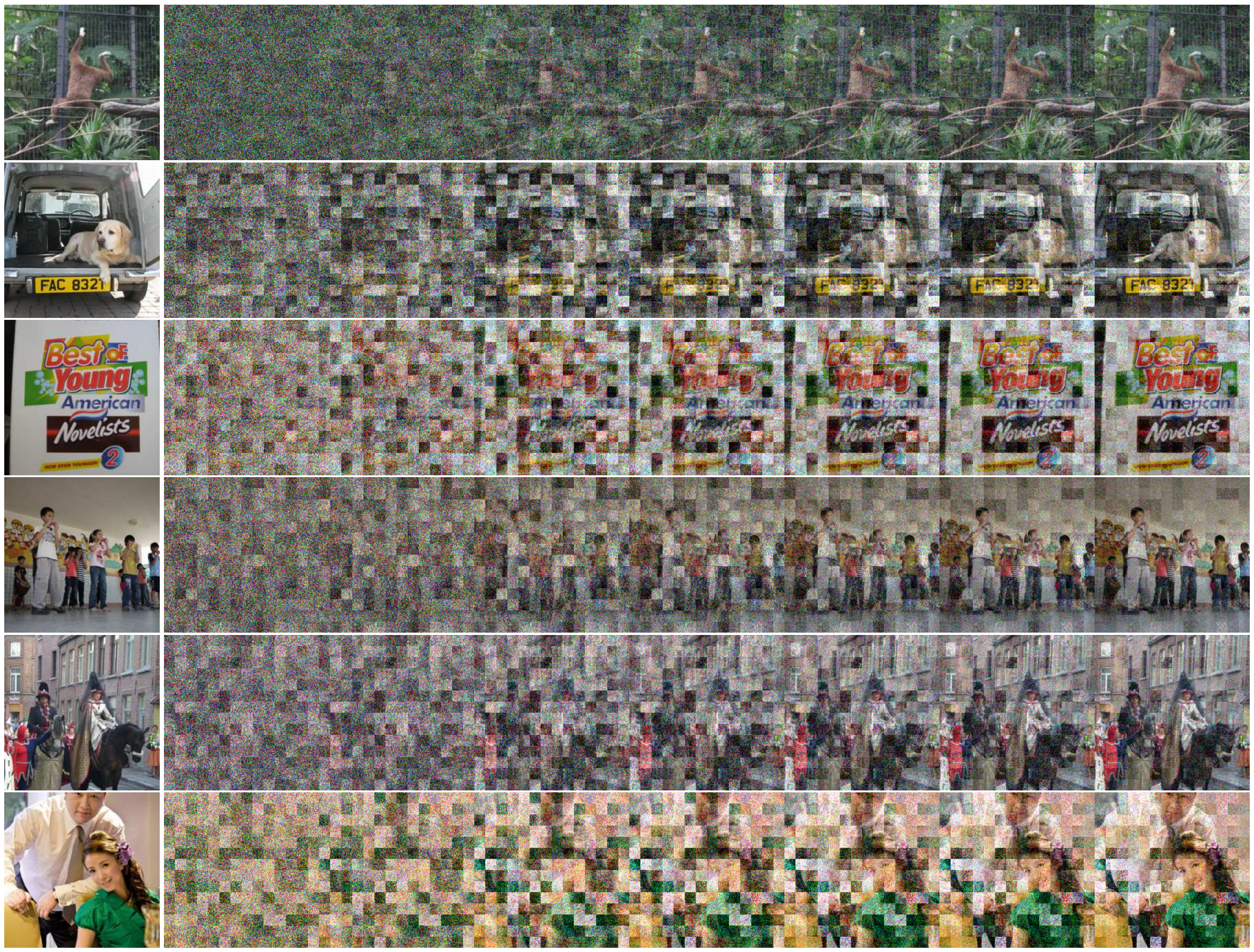
Figure 9. Results with different sets of parameters involved.

B. More results

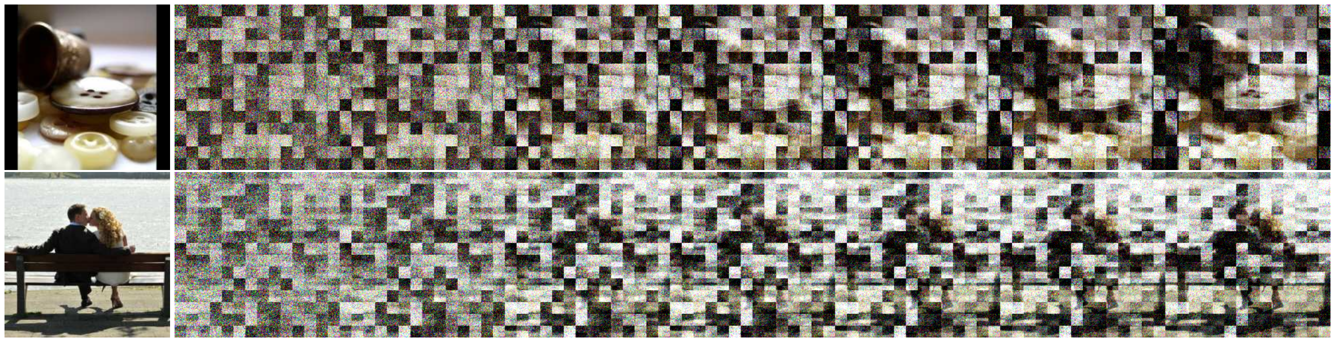
B.1. Optimization-based attack towards single image data

Here we show more ImageNet attack results in Fig.10 Under most cases, the attacks are successful and expose enough information to break privacy. Depending on the content of original images, attacks can have different levels of failures sometimes, when the optimization fall into a bad local minimum and the result has block artifacts. The hardness of attack does not have a preference for certain classes; it depends on the content of original sample. We can observe that, images with higher contrast are easier to have stronger block artifacts on their reconstructions.

We display 6 successful results and 2 failure cases in Fig.10. Each row represents the reconstruction process of a single image; ground-truth input image is shown on the left of each row.



Successful attack results.



Failure attack results.

Figure 10. More results on ImageNet for single image attacks. From left to right: the ground-truth image; reconstrutions at iteration 50, 100, 500, 1000, 2000, 3000, 5000.

B.2. Optimization-based attack towards batched images

Optimization-based APRIL attack obtains great results on batched images. We found that inverting a batch of

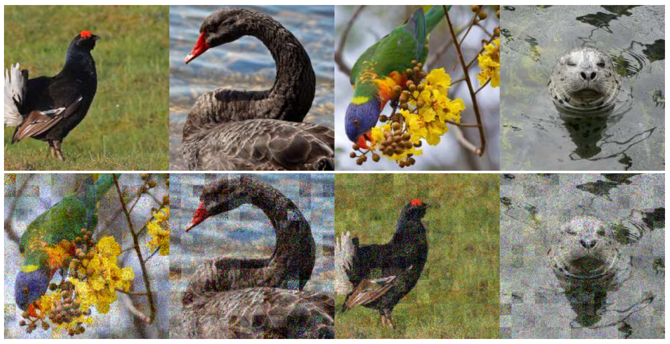


Figure 11. Reconstruction of a batch of 4 ImageNet samples. The first row: ground-truth batch images. The second row: reconstruction results by optimization-based APRIL.

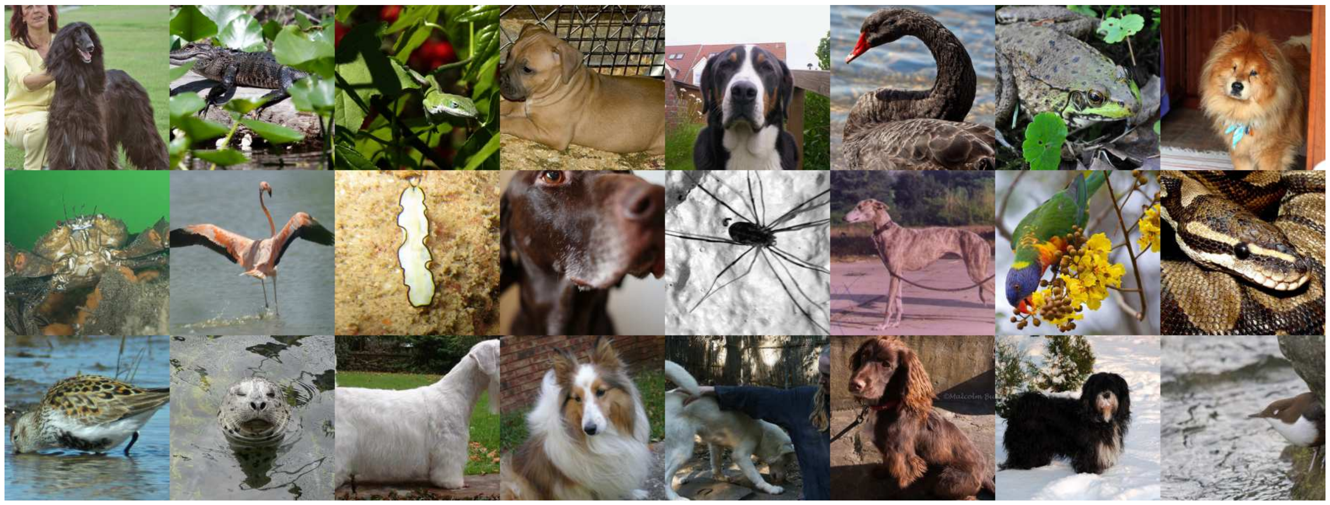


Figure 12. A batch of 24 ImageNet samples.

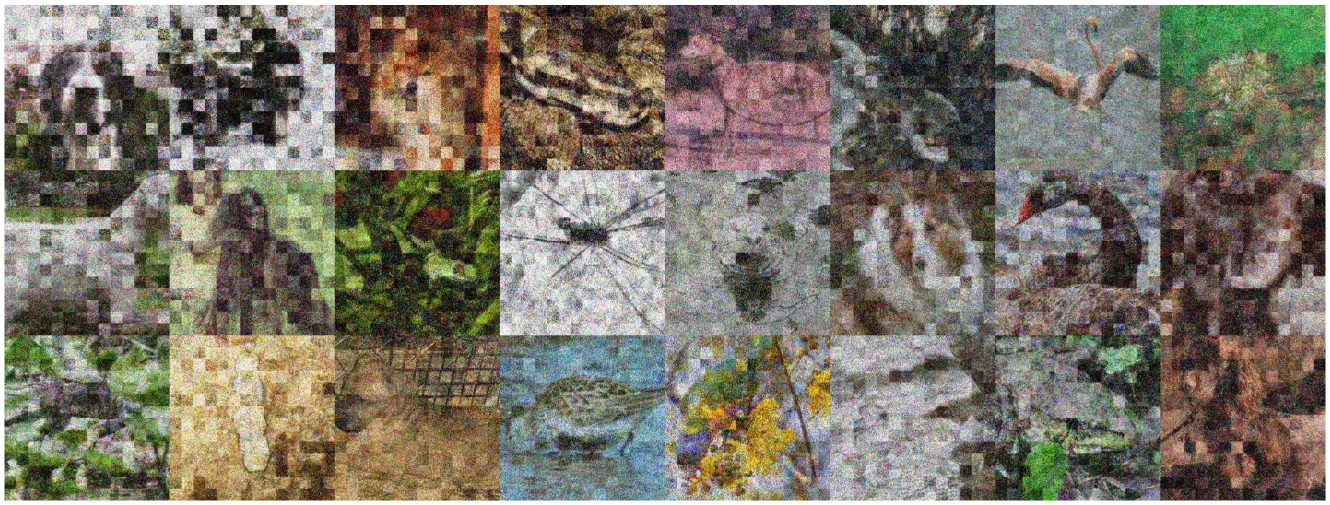


Figure 13. Reconstruction of the image batch in Fig. 12 using optimization-based APRIL.

several images is not markedly harder than inverting single image; with the same setting of iterations and learning rates, even if the batch size is increased from 4 to 24, the reconstructions are still recognizable. We show the results in Fig.11 and Fig.13.

B.3. Closed-form APRIL attack towards batched image data

It is worth to notice that the closed-form APRIL attack uses the averaged mean of gradients of all samples in a batch to solve the closed-form solution. Due to the dimensionality, we can only obtain a single image proxy for the whole batch. The results are shown in Fig.14, which give almost no useful information towards original input images.

As another closed-form attack, R-GAP [1] provides results of their approach on batched data as well in their Figure 7. Their results are easy to understand since they look like additive combinations of the original batch of images.

In contrast, our results are not recognizable and easy to be interpreted since we do not obtain the solution in an additive way, but in a multiplicative way.

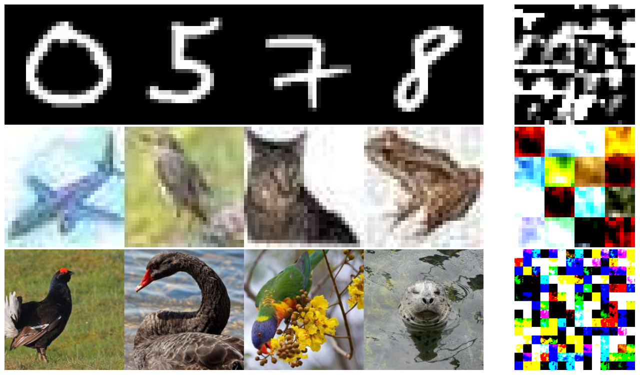


Figure 14. Closed-form APRIL attack results on batched inputs, which show perplexing patterns of tanglement.

References

[1] Junyi Zhu and Matthew B Blaschko. R-gap: Recursive gradient attack on privacy. In *ICLR*, 2020. 4