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

Abstract

*Federated learning frameworks typically require collaborators to share their local gradient updates of a common model instead of sharing training data to preserve privacy. However, prior works on Gradient Leakage Attacks showed that private training data can be revealed from gradients. So far almost all relevant works base their attacks on fully-connected or convolutional neural networks. Given the recent overwhelmingly rising trend of adapting Transformers to solve multifarious vision tasks, it is highly valuable to investigate the privacy risk of vision transformers. In this paper, we analyse the gradient leakage risk of self-attention based mechanism in both theoretical and practical manners. Particularly, we propose **APRIL** - **A**ttention **PRI**vacy **L**eakage, which poses a strong threat to self-attention inspired models such as ViT. Showing how vision Transformers are at the risk of privacy leakage via gradients, we urge the significance of designing privacy-safer Transformer models and defending schemes.*

1. Introduction

Federated or collaborative learning [25] have been gaining massive attention from both academia [20, 21] and industry [7, 19]. For the purpose of privacy-preserving, the typical federated learning keeps local training data private and trains a global model by sharing its gradients collaboratively. By avoiding to transmit the raw data directly to a central server, the learning paradigm is widely believed to offer sufficient privacy. Thereby, it has been employed in real-world applications, especially when user privacy is highly sensitive, *e.g*. hospital data [2, 18].

Whilst this setting prevents direct privacy leakage by keeping training data invisible to collaborators, a recent line of the works [12, 16, 39, 41, 43, 44] demonstrates that it is possible to (partially) recover private training data from the model gradients. This attack dubbed *gradient leakage* or *gradient inversion* poses a severe threat to the federated learning systems. The previous works primarily focus on inverting gradients from fully connected networks (FCNs) or convolutional neural networks (CNNs). Particularly, Yin *et al*. [39] recover images with high fidelity relying on gradient matching with BatchNorm layer statistics; Zhu *et al*. [43] theoretically analyse the risk of certain architectures to enable the full recovery. One intriguing question of our interest is that, *does gradient privacy leakage occur in the context of architectures other than FCNs and CNNs?*

The recent years have witnessed a surge of methods of Transformer [32]. As an inherently different architecture, Transformer can build large scale contextual representation models, and achieve impressive results in a broad set of natural language tasks. For instance, huge pre-trained language models including BERT [8], XLNet [38], GPT-3 [3], Megatron-LM [30], and so forth are established on the basis of Transformers. Inspired by the success, original works [1, 6, 29, 33] seek to the feasibility of leveraging self-attention mechanism with convolutional layers to vision tasks. Then, DETR [4] makes pioneering progress to use Transformer in object detection and ViT [10] resoundingly succeeds in image classification with a pure Transformer architecture. Coming after ViT, dozens of works manage to integrate Transformer into various computer vision tasks [11, 22–24, 35–37, 40]. Notably, vision Transformers are known to be extremely data-hungry [10], which makes the large-scale learning in the federated fashion more favorable.

Despite the rapid progress aforementioned, there is a high chance that vision Transformers suffer the gradient leakage risk. Nevertheless, the line of the study on this privacy issue is absent. Although the prior work [16] provides an attack algorithm to recover private training data for a Transformer-based language model via an optimization process, the inherent reason of Transformer's vulnerability is unclear. Different with leakage on Transformer in natural language tasks [16], we claim that vision Transformers with the position embedding not only encodes positional information for patches but also enables gradient inversion from the layer. In this paper, we introduce a novel analytic gradient leakage to reveal why vision Transformers are easy to be attacked. Furthermore, we explore gradient leakage by recovery mechanisms based on an optimization approach and provide a new insight about the position embedding. Our results of gradient attack will shed light on future designs for privacy-preserving vision Transformers.

To summarize, our contributions are as follows:

- We prove that for the classic self-attention module, the input data can be perfectly reconstructed without solving an intractable optimization problem, if the gradient w.r.t. the input is known.
- We demonstrate that jointly using self-attention and learnable position embedding place the model at severe privacy risk. The attacker obtain a closed-form solution to the privacy leakage under certain conditions, regardless of the complexity of networks.
- We propose an Attention Privacy Leakage (APRIL) attack, to discover the Archilles' Heel. As an alternative, APRIL performs an optimization-based attack, apart from the closed-form attack. The attacks show that our results superior to SOTA.
- We suggest to switch the learnable position embedding to a fixed one as the defense against privacy attacks. Empirical results certify the effectiveness of our defending scheme.

2. Preliminary

Federated Learning. Federated learning [25] offers the scheme that trains statistical models collaboratively involving multiple data owners. Due to the developments in areas of privacy, large-scale training, and distributed optimization, federated learning methods have been deployed by applications which require computing at the edge [2, 9, 13, 14, 28]. In this scenario, we aim to learn a global model by locally processed *client* data and communicating intermediate updates to a central *server*. Formally, the typical goal is minimizing the following loss function l with parameters w,

$$\min_{w} l_w(x,y), \quad \text{where} \quad l_w(x,y) := \sum_{i=1}^{N} p_i l_w^i(x_i,y_i) \quad (1)$$

where $p_i \geq 0$ and $\sum_i p_i = 1$. Since the N clients owns the private training data. Let (x_i, y_i) denote samples available locally for the ith client, and $l_w^i(x_i, y_i)$ denote the local loss function. In order to preserve data privacy, clients periodically upload their gradients $\nabla_w l_w^i(x_i, y_i)$ computed on their own local batch. The server aggregates gradients from all clients, updates the model using gradient descent and then sends back the updated parameters to every client.

Gradient Leakage Attack. As an *honest-but-curious* adversary at the server side may reconstruct clients' private training data without messing up the training process, sharing gradients in federated learning is no longer safe for client data. Endeavors of existing threat models which use gradients to recover input mainly focus on two directions: *optimization-based attacks* and *closed-form attacks*.

The basic recovery mechanism is defined by optimizing an euclidean distance as follows,

$$\min_{x_i', y_i'} \|\nabla_w l_w^i(x_i, y_i) - \nabla_w l_w^i(x_i', y_i')\|^2 \quad (2)$$

Deep leakage [44] minimizes the matching term of gradients from dummy input (x_i', y_i') and those from real input (x_i, y_i)[1]. On the top of this proposal, iDLG [41] finds that in fact we can derive the ground-truth label from the gradient of the last fully connected layer. By eliminating one optimization objective in Eq.(2), the attack procedure becomes even faster and smoother. Also, Geiping *et al.* [12] prove that inversion from gradient is strictly less difficult than recovery from visual representations. GradInversion [39] incorporates heuristic image prior as regularization by utilizing BatchNorm matching loss and group consistency loss for image fidelity. Lately, GIML [17] illustrates that a generative model pre-trained on data distribution can be exploited for reconstruction.

One essential challenge of optimization procedures is that there is no sufficient condition for the uniqueness of the optimizer. The closed-form attack, as another of the ingredients in this line, is introduced by Phong *et al.* [27], which reconstructs inputs using a shallow network such as a single-layer perceptron. R-GAP [43] is the first derivation-based approach to perform an attack on CNNs, which models the problem as linear systems with closed-form solutions. Compared to the optimization-based method, analytic gradient leakage heavily depends on the architecture of neural networks and thus cannot always guarantee a solution.

Transformers. Transformer [32] is introduced for neural machine translation to model the long-term correlation between tokens meanwhile represent dependencies between any two distant tokens. The key of outstanding representative capability comes from stacking multi-head self-attention modules. Recently, vision Transformers and its variants are broadly used for powerful backbones [10, 24, 31], object detection [4], semantic segmentation [42], image generation [5, 15, 26], *etc*.

Given the fundamentals of vision Transformer, we will investigate the gradient leakage in terms of closed-formed and optimization-based manners. Thus far, almost all the gradient leakage attacks adopt CNNs as the testing ground, typically using VGG or ResNet. Besides, TAG [16] conducts experiments on popular language models using Transformers without concerning any analytic solution as well as the function of position embedding.

[1] We omit the index i for clients in followings to manifest that the algorithm can work for any client.

3. APRIL: Attention PRIvacy Leakage

In light of the missing investigation of the gradient leakage problem for vision transformers, we first prove that gradient attacks on self-attention can be analytically conducted. Next, we will discuss the possible leakage from the position embedding based on its analytic solution, which naturally gives rise to two attack approaches.

3.1. Analytic Gradient Attack on Self-Attention

It has been proven that the closed-form solution for input x can always be perfectly obtained on a fully-connected layer $\sigma(Wx + b) = z$, through deriving gradients w.r.t. weight W and bias b. The non-linear function σ is an activation [27]. In this work, we delve into a more subtle formulation of a self-attention to demonstrate the existence of the closed-form solution.

Theorem 1. *(Input Recovery). Assume a self-attention module expressed as:*

$$Qz = q; Kz = k; Vz = v; \tag{3}$$

$$\frac{softmax(q \cdot k^T)}{\sqrt{d_k}} \cdot v = h \tag{4}$$

$$Wh = a; \tag{5}$$

where z is the input of the self-attention module, a is the output of the module. Let Q, K, V, W denote the weight matrix of query, key, value and projection, and q, k, v, h denote the intermediate feature map. Suppose the loss function can be written as

$$l = l(f(a), y)$$

If the derivative of loss l w.r.t. the input z is known, then the input can be recovered uniquely from the network's gradients by solving the following linear system:

$$\frac{\partial l}{\partial z} z^T = Q^T \frac{\partial l}{\partial Q} + K^T \frac{\partial l}{\partial K} + V^T \frac{\partial l}{\partial V}$$

Proof. In spite of the non-linear formulation of self-attention modules, the gradients w.r.t. z can be derived in a succinct linear equation:

$$\frac{\partial l}{\partial z} = Q^T \frac{\partial l}{\partial q} + K^T \frac{\partial l}{\partial k} + V^T \frac{\partial l}{\partial v} \tag{6}$$

Again, according to the chain rule of derivatives, we can derive the gradients w.r.t. Q, K and V from Eq. (3):

$$\begin{aligned} \frac{\partial l}{\partial Q} &= \frac{\partial l}{\partial q} z^T \\ \frac{\partial l}{\partial K} &= \frac{\partial l}{\partial k} z^T \\ \frac{\partial l}{\partial V} &= \frac{\partial l}{\partial v} z^T \end{aligned} \tag{7}$$

Algorithm 1: Closed-Form APRIL

Input: Attention module: $F(z, w)$;
Module weights w; Module gradients $\frac{\partial l}{\partial w}$
Derivative of loss w.r.t. z: $\frac{\partial l}{\partial z}$
Output: Embedding feed into attention module: z

```
procedure APRIL-CLOSED-FORM(F, w, ∂l/∂w, ∂l/∂z)
    Extract Q, K, V from module weights w
    Extract ∂l/∂Q, ∂l/∂K, ∂l/∂V from module gradients ∂l/∂w
    A ← ∂l/∂z
    b ← Q^T · ∂l/∂Q + V^T · ∂l/∂V + K^T · ∂l/∂K
    z ← A† · b                          ▷ A†: Moore-Penrose
                                        ▷ pseudoinverse of A
    z ← z^T                             ▷ Transpose
end procedure
```

By multiplying z^T to both sides of Eq. (6) and substituting Eq. (7), we obtain:

$$\begin{aligned} \frac{\partial l}{\partial z} z^T &= Q^T \frac{\partial l}{\partial q} z^T + K^T \frac{\partial l}{\partial k} z^T + V^T \frac{\partial l}{\partial v} z^T \\ &= Q^T \frac{\partial l}{\partial Q} + K^T \frac{\partial l}{\partial K} + V^T \frac{\partial l}{\partial V} \end{aligned} \tag{8}$$

which completes the proof. ■

Remark. *Surprisingly we find that for a malicious attacker aiming to recover the input data z. Since an adversary in the context of federated learning knows both learnable parameters and gradients w.r.t. them, in this case, Q, K, V and $\frac{\partial l}{\partial Q}$, $\frac{\partial l}{\partial K}$, $\frac{\partial l}{\partial V}$. The right side of Eq. (8) is known. As a result, once the derivative of the loss w.r.t. the input $\frac{\partial l}{\partial z}$ is exposed to the adversary, the attacker can easily get an accurate reconstruction of z by solving the linear equation system in Eq. (8).*

Solution Feasibility. Suppose the dimension of the embedding z is $\mathbb{R}^{p \times c}$, with patch number p and channel number c. This linear system has $p \times c$ unknown variables yet $c \times c$ linear constraints. Since deep neural networks normally have wide channels for the sake of expressiveness, $c \gg p$ in most model designs, which leads to an overdetermined problem and thereby a solvable result. In other words, z can be accurately reconstructed if $\frac{\partial l}{\partial z}$ is available. The entire procedure of the closed-form attack is presented in Alg.1.

3.2. Position Embedding: The Achilles' Heel

Now we focus on the how to access the critical derivative $\frac{\partial l}{\partial z}$ by introducing the leakage caused by the position embedding. Under general settings of federated learning, the sensitive information related with z is invisible from users' side. Here, we show that $\frac{\partial l}{\partial z}$ is unfortunately exposed by gradient sharing for vision Transformers with a

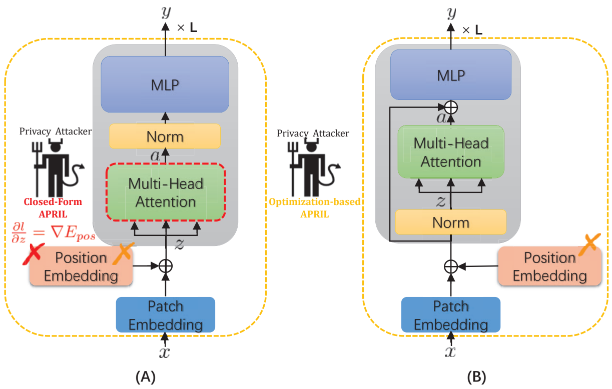


Figure 1. We consider two Transformer designs throughout the paper. (A): Encoder modules stack multi-head attention, normalization, and MLP in VGG-style. (B): A real-world design as introduced in ViT [10]. The architecture in (A) satisfies the precondition of a closed-form APRIL attack, since the output of position embedding is exactly input for multi-head attention, showing by the red dashed line box. In contrast, the optimization-based APRIL attack can be placed in any design of architectures, showing by the yellow dashed line boxes in (A) and (B).

learnable position embedding. Specifically, we give the following theorem to illustrate the leakage.

Theorem 2. *(Gradient Leakage). For a Transformer with learnable position embedding E_{pos}, the derivative of loss w.r.t. E_{pos} can be given by*

$$\frac{\partial l}{\partial E_{pos}} = \frac{\partial l}{\partial z} \tag{9}$$

where $\frac{\partial l}{\partial z}$ is defined by the linear system in Theorem 1.

Proof. Without loss of generality, the embedding z defined by Theorem 1 can be divided into a patch embedding E_{patch} and a learnable position embedding E_{pos} as,

$$z = E_{patch} + E_{pos} \tag{10}$$

Straightforwardly, we compute the derivative of loss w.r.t. E_{pos} using Eq. (10), Eq. (9) holds. ∎

Remark. *The sensitive information $\frac{\partial l}{\partial z}$ is exactly the same as the gradient of the position embedding $\frac{\partial l}{\partial E_{pos}}$, denoting as ∇E_{pos} for simplicity. As model gradients are sharing, ∇E_{pos} is available for not only legal users but also potential adversaries, which means a successful attack on self-attention inputs.*

While vision Transformers [10, 24, 34] embody prominent accuracy raise using learnable position embeddings rather than the fixed ones, updating of parameter E_{pos} will result in privacy-preserving troubles based on our theory. More severely, the attacker only requires a learnable position embedding and a self-attention stacked at the bottom in VGG-style, regardless of the complexity of the rest architecture, as shown in Fig. 1 (A). At a colloquial level, we suggest two strategies to alleviate this leakage, which is either employing one fixed position embedding instead of the learnable one or updating ∇E_{pos} only on local client without transmission.

3.3. APRIL attacks on vision Transformer

So far the analytic gradient attack have succeeded in reconstructing input embedding z meanwhile obtaining the gradient of position embedding ∇E_{pos}. One question is that can APRIL take advantage of the sensitive information to further recover the original input x. The answer is affirmative.

Closed-Form APRIL. As a matter of the fact, APRIL attacker can inverse the embedding via a linear projection to get original input pixels. For a vision Transformer, the input image is partitioned into many patches and sent through a so-called "Patch Embedding" layer, defined as

$$E_{patch} = W_p x \tag{11}$$

The bias term is omitted since it can be represented in an augmented matrix W_p. With W_p, pixels are linearly mapped to features, and the attacker calculates the original pixels by left-multiply its pseudo-inverse.

Optimization-based APRIL. Given the linear system in Theorem 1, it can also be decomposed into two components as z and ∇E_{pos} based on Eq.(9). Arguably, component ∇E_{pos} indicates the directions of the gradients of position embeddings and contributes to the linear system indepently with data. Considering the significance of the learnable position embedding in gradient leakage, intuitively, matching the updating direction of E_{pos} with an direction caused by dummy data can do benefits on the recovery. Therefore, we proposed an optimization-based attack with constraints on ∇E_{pos}. To do so, apart from architecture in Fig. 1(A), typical design of ViT illustrated in Fig. 1(B) using normalization and residual connections with a different stacked order can also be attacked by our proposed APRIL.

For expression simicity, we use $\nabla w'$ and ∇w denote the gradients of parameter collections for dummy data and real inputs, respectively. In detail, the new integrated term of gradients of ∇E_{pos} is set as $\mathcal{L}_A$. For modelling directional information, we utilize a cosine similarity between real and dummy position embedding derivatives as a regularization. The intact optimization problem is written as

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_G + \alpha \mathcal{L}_A \\ &= \|\nabla w' - \nabla w\|_F^2 - \alpha \cdot \frac{< \nabla E_{pos}, \nabla E_{pos}^{'} >}{\|\nabla E_{pos}\| \cdot \|\nabla E_{pos}^{'}\|}. \end{aligned} \tag{12}$$

Algorithm 2: Optimization-based APRIL

Input: Transformer with learnable position embedding: $F(x, w)$; Module parameter weights : w; Module parameter gradients: ∇w; APRIL loss term scaler: α

Output: Image feed into the self-attention module: x

procedure APRIL-OPTIMIZATION-ATTACK($F, w, \nabla w$)

Extract final linear layer weights w_{fc} from w

$y \leftarrow i \quad$ s.t. $\nabla w_{fc}^{i}{}^{T} \nabla w_{fc}^{j} \leq 0, \forall j \neq i$ ▷ Extract ground-truth label using the iDLG trick

Extract position embedding layer's gradients ∇E_{pos} from ∇w

$x' \leftarrow \mathcal{N}(0, 1)$ ▷ Initialize the dummy input

While not converged **do**

$\frac{\partial l}{\partial w}' \leftarrow \partial l(F(x'; w), y)/\partial w$ ▷ Calculate dummy gradients

$\mathcal{L}_G = \|\nabla w' - \nabla w\|_F^2$ ▷ Calculate L-2 difference between gradients

$\frac{\partial l}{\partial E'_{pos}} \leftarrow \partial l(F(x'; w), y)/\partial E'_{pos}$ ▷ Calculate the derivative of dummy loss w.r.t. dummy input

$\mathcal{L}_A = -\frac{<\nabla E_{pos}, \nabla E'_{pos}>}{\|\nabla E_{pos}\| \cdot \|\nabla E'_{pos}\|}$ ▷ Calculate cosine distance between derivative of input

$\mathcal{L} = \mathcal{L}_G + \alpha \mathcal{L}_A$

$x' \leftarrow x' - \eta \nabla_{x'} \mathcal{L}$ ▷ Update the dummy input

end procedure

where hyperparameter α balances the contributions of two matching losses. Eventually, we set Eq.(12) to be another variant of our proposed method, optimization-based APRIL attack. The associated algorithm is described in Alg.2. By enforcing a gradient matching on the learnable position embedding, it is plaguily easy to break privacy in a vision Transformer.

4. Experiments

In this section, we aspire to carry out experiments to answer the following questions: (1) To what extent can APRIL break privacy of a Transformer? (2) How strong is the APRIL attack compared to existing privacy attack methods? (3) What defensive strategy can we take to alleviate APRIL attack? (4) How to testify the functionality of position embedding in privacy preserving?

We mainly carry out experiments in the setting of image classification; however, APRIL as a universal attack for Transformers can also be performed in a language task setting. Here we only discuss APRIL attack for vision Transformers in this section.

We carry out experiments on two different architectures, as illustrated in Fig. 1, architecture (A) has a position embedding layer directly connected to attention module, making it possible to perform APRIL-closed-form attack. Architecture (B) has the same structure as ViT-Base [10], which is composed of multiple encoders, each with a normalization layer before attention module as well as a residual connection. For small datasets like CIFAR and MNIST, we refer to the implementation of ViT-CIFAR[2]. We set the hidden dimension to 384, attention head to 4, and partition input images into 4 patches. The encoder depth is 4, after that the classification token is connected to a classification head. For experiments on ImageNet, we follow the original ViT design[3] and architecture setting, which includes 16x16 image patch size, 12 attention heads, 12 layers of encoders with hidden dimensions of 768.

4.1. APRIL as the Gradient Attack

We first apply APRIL attacks on Architecture (A) and compare it with other attacking approaches. As Fig. 2 shows, closed-form APRIL attack provides a perfect reconstruction, which shows nearly no difference to the original input, which proves the correctness of our theorem. Comparing optimization-based attacks, for easy tasks like MNIST and CIFAR with a clean background, all existing attacking algorithms show their ability to break privacy, although DLG [44] and IG [12] have some noises in their results. The comparison is obvious for ImageNet reconstructions, where DLG, IG and TAG reconstructions are nearly unrecognizable to humans, with strong block artifacts. In contrast, the proposed APRIL-Optimization attack behaves prominently better, which reveals quite a lot of sensitive information from the source image, including details like the color and shape of the content.

We further studied the optimization procedure of reconstruction, shown in Fig. 3. We illustrate the updating process of the dummy image. We can observe that all three approaches can break some sort of privacy, but they differ in convergence speed and final effects. An apparent observa-

[2]https://github.com/omihub777/ViT-CIFAR

[3]https://github.com/lucidrains/vit-pytorch

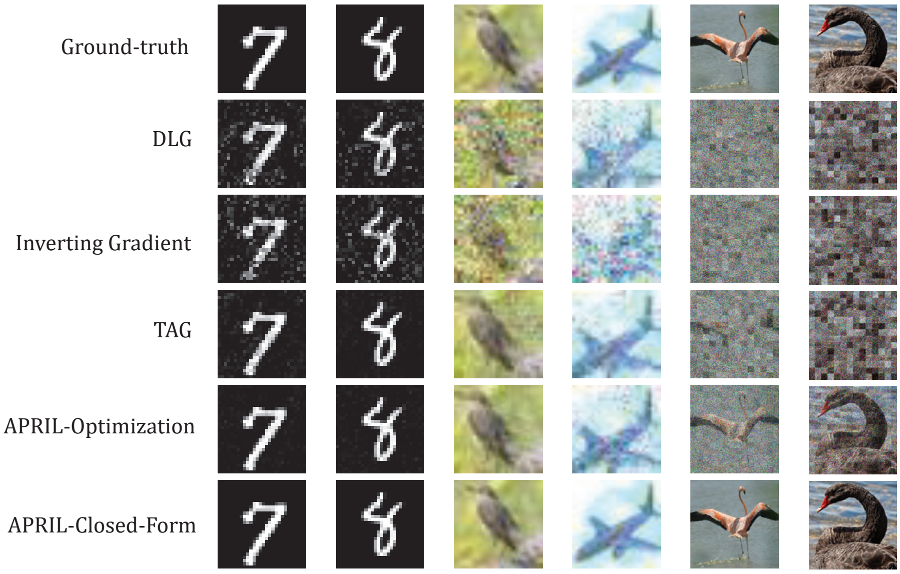


Figure 2. Results for different privacy attacking approaches on Architecture (A). For optimization-based attacks, we use an Adam optimizer to update 800 iterations for MNIST, 1500 iterations for CIFAR-10 and 5000 iterations for ImageNet. Please zoom-in to see details.

Attack	MNIST		CIFAR-10		ImageNet	
	MSE	SSIM	MSE	SSIM	MSE	SSIM
DLG [44]	1.291e-04 $\pm$ 2.954e-04	0.997 $\pm$ 0.003	0.017 $\pm$ 0.009	0.959$\pm$0.045	1.328$\pm$0.593	0.056 $\pm$ 0.027
IG [12]	0.043$\pm$0.022	0.833$\pm$0.076	0.125$\pm$0.102	0.635$\pm$0.165	1.671$\pm$0.653	0.029$\pm$0.013
TAG [16]	**3.438e-05 $\pm$ 1.322e-05**	**0.998$\pm$0.002**	0.006 $\pm$ 0.005	0.965$\pm$0.047	1.180 $\pm$ 0.473	0.062 $\pm$ 0.026
APRIL	4.796e-05$\pm$3.593e-05	**0.998 $\pm$ 0.002**	**0.002$\pm$0.006**	**0.991 $\pm$ 0.027**	**1.092$\pm$0.663**	**0.099 $\pm$ 0.046**

Table 1. Mean and standard deviation for MSE of 500 reconstructions on MNIST, CIFAR-10 and ImageNet validation datasets, respectively. We randomly selected 50 images from each class in MNIST and CIFAR-10, and one image for random 500 classes in ImageNet.

tion is that our optimization-based APRIL converges consistently faster than the other two. Besides, our approach generally ends up at a better terminal point, which results in smoother and cleaner image reconstructions.

Apart from visualization results, we want to have a quantitative comparison between these optimization-based attacks. We carry out this experiment on Architecture(B), where we do not have the condition to use closed-form APRIL attack. The statistical results from Sec. 4 shows consistent good performance of APRIL, and we obtain best results nearly across every task setting.

Finally, we try to attack batched input images. As shown in Fig. 4, our optimization results on batched input achieved impressive results as well. Note here we used the trick introduced by Yin *et al.* [39] to restore batch labels before optimization. More results are put in Appendix. It's worth mentioning that the use of a closed-form APRIL attack is limited under batched setting, since the gradients are contributed by all samples in a batch, and we can only solve an "averaged" version of z in Eq. (8). We give more reconstruction results and discuss more thoroughly on the phenomenon in Appendix.

All experiments shown above demonstrate that the proposed APRIL outperforms all existing privacy attack approaches in the context of Transformer, thus posing a strong threat to Vision Transformers.

4.2. APRIL-inspired Defense Strategy

How robust is the closed-form APRIL. In the last subsection, we show that under certain conditions, closed-form APRIL attack can be executed to get almost perfect reconstructions. The execution of this attack is based on solving a linear system. Linear systems can be unstable and ill-conditioned when the condition number is large. With this knowledge, we are interested to know how much disturbance can APRIL bear to remain a good attack? We discuss a few defensive strategies towards APRIL.

We first testified the influence of changing hidden channel dimensions. A successful closed-form reconstruction relies on the linear system with $P \cdot C$ unknowns and $C \cdot C$ constraints, to be overdetermined. As common configuration suggests C far larger than P, we deem the linear system to be solvable. To test the robustness of APRIL under different architecture settings, we try four different hidden di-

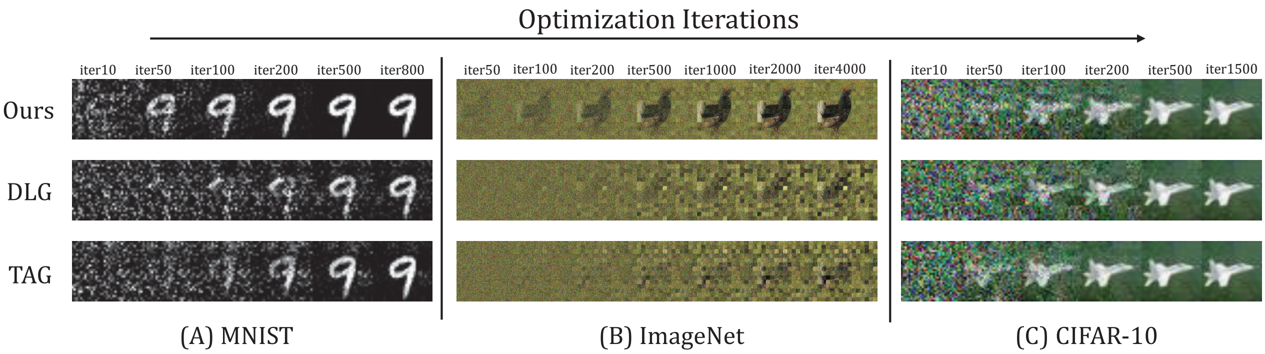


Figure 3. Visualization of the optimization process for optimization-based APRIL, DLG and TAG. Our approach has faster convergence speed and does not easily fall into bad local minima, thus yields a prominently better reconstruction result.

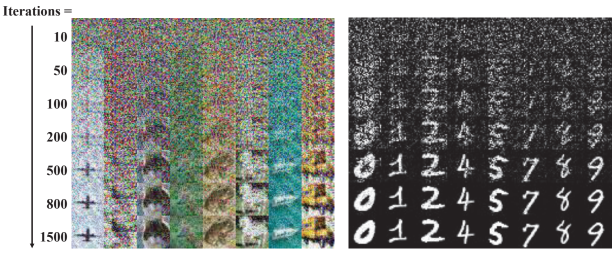
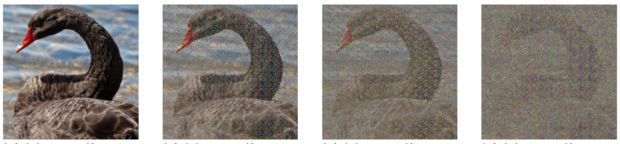


Figure 4. Optimization-based APRIL attack on batched inputs.



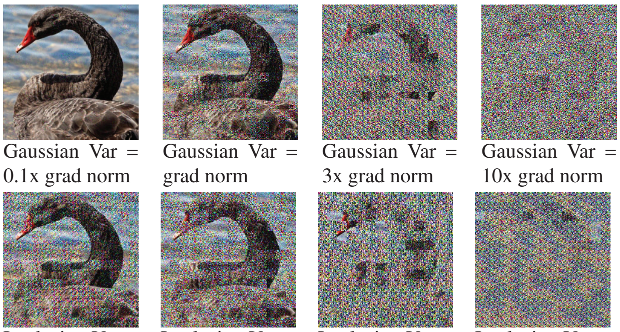
hidden dimension=768 hidden dimension=384 hidden dimension=192 hidden dimension=96

Figure 5. Influences of varying hidden dimension to the reconstruction of APRIL attack.

mensions. As Fig. 5 shows, using the original configuration of ViT-base [10] cannot be privacy-preserving, the original input image can be entirely leaked by closed-form APRIL attack. Only by shrinking hidden dimensions to a small value (*e.g.*, half of the patch number) can we have solid protection. However, in this configuration, we doubt the network's capacity to gain high accuracy with such small channel number.

Another more straightforward way to defend against privacy attacks from gradients is to add noise on gradients. We experiment with Gaussian and Laplacian noises and report results in Fig. 6. We found that the defense level does not depend on the absolute magnitude of noise variance, but its relative scale to gradient norm. Specifically, when the Gaussian noise variance is lower than 0.1 times (or 0.01 for Laplacian) of gradient norm, the defense won't work. As the variance goes up, the defense ability is greatly promoted.

A Practical and Cheap Defense Scheme. Apart from adding noise and changing channel dimensions, a more straightforward way of defending against APRIL is to switch learnable position embedding to a fixed one. In this part, we will show that this is a realistic and practical defense, not only for the proposed APRIL, but for all kinds of attacks.



Gaussian Var = 0.1x grad norm Gaussian Var = grad norm Gaussian Var = 3x grad norm Gaussian Var = 10x grad norm

Laplacian Var = 0.01x grad norm Laplacian Var = 0.1x grad norm Laplacian Var = grad norm Laplacian Var = 3x grad norm

Figure 6. Influences of adding noise to gradients.

By using a fixed position embedding, clients will not share the gradients w.r.t. the input. Therefore, it is impossible to perform closed-form APRIL attack. How will optimization-based privacy attacks act when the position embedding is transparent to the attacker?

We experimented to find out the answer. Note that when position embedding is unknown to the attacker, the optimization-based APRIL attack turns into a more general DLG attack. From results, we noticed that similar to *twin data* mentioned by [43], closing the position embedding gradients seems to result in a family of anamorphic data, which is highly different from original data, but can trigger exactly similar gradients in a Transformer. We visualize these patterns as shown in Sec. 4.2. Currently we are not sure about the relationship between twin and original data, but it's safe to conclude that if we cease sharing position embedding gradients, the gradient matching process will produce semantically meaningless reconstructions. In

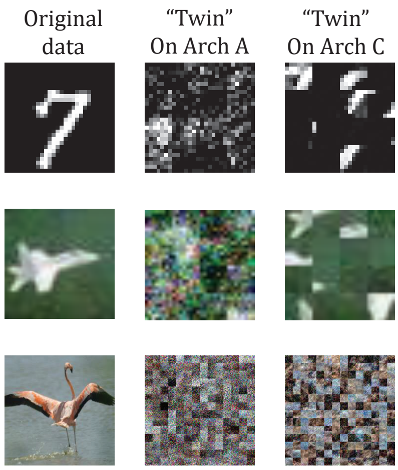
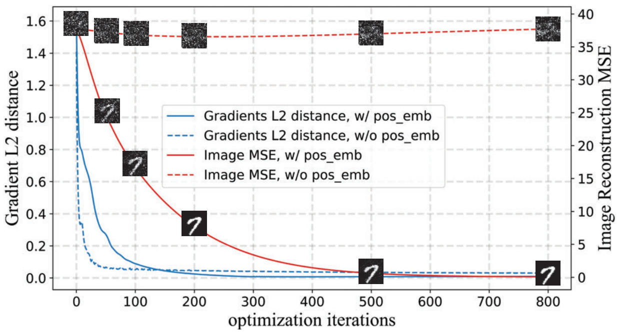
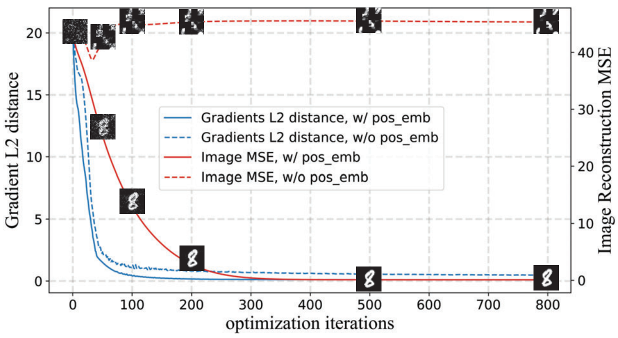


Figure 7. Twin data emerge from privacy attack after we stop sharing position embedding. It attested the validity of the defense, in which way confirms that position embedding is indeed the most critical part to Transformer's privacy.



(A) Gradient l2 loss and image MSE on Architecture A



(B) Gradient l2 loss and image MSE on Architecture B

Figure 8. Changes of gradient matching and input reconstruction versus optimization iterations. When position embedding is off, matching gradients does not provide semantically meaningful reconstructions.

this way, the attacks fail to break privacy.

To sum up, changing the learnable position to fixed ones or simply not sharing position embedding gradient is practical to prevent privacy leakage in Transformers, which preserves privacy in a highly economic way.

5. Discussion and Conclusion

In this paper, we introduce a novel approach **A**ttention **PRI**vacy **L**eakage attack (**APRIL**) to steal private local training data from shared gradients of a Transformer. The attack builds its success on a key finding that learnable position embedding is the weak spot for Transformer's privacy. Our experiments show that in certain cases the adversary can apply a closed-form attack to directly obtain the input. For broader scenarios, the attacker can make good use of position embedding to perform an optimization-based attack to easily reveal the input. This sets a great challenge to training Transformer models in distributed learning systems. We further discussed possible defenses towards APRIL attack, and verified the effectiveness of using a fixed position embedding. We hope this work would shed light on privacy-preserving network architecture design. In summary, our work has a key finding that learnable position embedding is a weak spot to leak privacy, which greatly advances the understanding of privacy leakage problem for Transformers. Based on the finding, we further propose a novel privacy attack APRIL and discuss effective defending schemes.

Limitation. Our proposed APRIL attack is composed of two parts: closed-form attack when the input gradients are exposed and optimization-based attack otherwise. Closed-form APRIL attack is powerful, nonetheless relies on a strong assumption, which makes it limited to use in real-world Transformer designs. On the other hand, optimization-based APRIL attack implicitly solves a nonlinear system. Although they all make good use of gradients from position embedding, there seems to be room to explore a more profound relationship between the two attacks.

Potential Negative Societal Impact. We demonstrate the privacy risk of learnable position embedding, as it is largely used as a paradigm in training Transformers. The privacy attack **APRIL** proposed in this paper could be utilized by the malicious to perform attack towards existing federated learning systems to steal user data. We put stress on the defense strategy proposed in the paper as well, and urge the importance of designing privacy-safer Transformer blocks.

6. Acknowledgement

This work was supported in part by the National Key Research and Development Program of China (No. 2020AAA0103400) and the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDA27040300).

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional net-

works. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019. 1

[2] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018. 1, 2

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*. Springer, 2020. 1, 2

[5] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pages 1691–1703. PMLR, 2020. 2

[6] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2019. 1

[7] Google TensorFlow Developers. TensorFlow Federated, December 2018. 1

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL, Volume 1 (Long and Short Papers)*, 2019. 1

[9] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 2, 4, 5, 7

[11] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *NeurIPS*, 2021. 1

[12] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *NeurIPS*, 2020. 1, 2, 5, 6

[13] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018. 2

[14] Li Huang, Andrew L Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of biomedical informatics*, 99, 2019. 2

[15] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *NeurIPS*, 2021. 2

[16] Deng Jieren, Wang Yijue, Shang Chao, Liu Hang, Rajasekaran Sanguthevar, and Ding Caiwen. Tag: Gradient attack on transformer-based language models. In *findings of EMNLP*, 2021. 1, 2, 6

[17] Jeon Jinwoo, Kim Jaechang, Lee Kangwook, Oh Sewoong, and Ok Jungseul. Gradient inversion with generative image prior. In *FL-ICML workshop in ICML*, 2021. 2

[18] Arthur Jochems, Timo M Deist, Johan Van Soest, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, Philippe Lambin, and Andre Dekker. Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467, 2016. 1

[19] Bonawitz Keith, Eichner Hubert, Grieskamp Wolfgang, Huba Dzmitry, Ingerman Alex, Ivanov Vladimir, Kiddon Chloe, Konecny Jakub, Mazzocchi Stefano, McMahan H.Brendan, Overveldt Timon, Van, Petrou David, Ramage Daniel, and Roselander Jason. Towards federated learning at scale: System design. In *SysML*, 2019. 1

[20] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. 1

[21] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 583–598, 2014. 1

[22] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Jinqiao Wang. Mst: Masked self-supervised transformer for visual representation. 2021. 1

[23] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint transformer: Feed forward neural painting with stroke prediction. In *ICCV*, pages 6598–6607, 2021. 1

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, October 2021. 1, 2, 4

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2

[26] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, pages 4055–4064, 2018. 2

[27] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5), 2018. 2, 3

[28] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. Privacy-preserving news recommendation model

learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1423–1432, 2020. 2

[29] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[30] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 1

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[34] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *ICCV*, pages 10033–10041, 2021. 4

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 2021. 1

[36] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. 2021. 1

[37] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *ICCV*, pages 11802–11812, 2021. 1

[38] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 32, 2019. 1

[39] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *CVPR*, pages 16337–16346, 2021. 1, 2, 6

[40] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, pages 12498–12507, 2021. 1

[41] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 1, 2

[42] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021. 2

[43] Junyi Zhu and Matthew B Blaschko. R-gap: Recursive gradient attack on privacy. In *ICLR*, 2020. 1, 2, 7

[44] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *NeurIPS*, 2019. 1, 2, 5, 6