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
96 ZhongGuanCun East Road, Haidian Distinct Beijing, China
{lujiahao2019, zhaotianli2019}@ia.ac.cn, {iva.shuanholmes, sheryl.zhangxi}@gmail.com, jcheng@nlpr.ia.ac.cn

Abstract

Federated learning frameworks typically require collaborators to share their local gradient updates of a global 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.

Introduction

Federated learning have been gaining massive attention from both academia and industry. 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.

Whilst this setting prevents direct privacy leakage by keeping training data invisible to collaborators, a recent line of the works (Zhu, Liu, and Han 2019; Geiping et al. 2020; Yin et al. 2021; Zhu and Blaschko 2020; Jieren et al. 2021) demonstrates that it is possible to recover private training data from the model gradients. This attack is dubbed *gradient leakage* or *gradient inversion*. Endeavors of the existing threat models mainly focus on two directions: *optimization-based attacks* and *closed-form attacks*.

Optimization-based attacks optimize an euclidean distance as follows,

$$\min_{x'_i, y'_i} \|\nabla_w l^i_w(x_i, y_i) - \nabla_w l^i_w(x'_i, y'_i)\|^2 \tag{1}$$

Deep leakage (Zhu, Liu, and Han 2019) minimizes the matching term of gradients from dummy input (x'_i, y'_i) and those from real input (x_i, y_i). On the top of this proposal, iDLG (Zhao, Mopuri, and Bilen 2020) finds that we can derive the ground-truth label from the gradient of the last fully connected layer. Also, Geiping *et al.* (Geiping et al. 2020) prove that inversion from gradient is strictly easier than recovery from visual representations. GradInversion (Yin et al. 2021) incorporates heuristic image prior as regularization by utilizing BatchNorm matching loss and group consistency loss for image fidelity.

The closed-form attack, as the other of the ingredients in this line, is introduced by Phong *et al.* (Phong et al. 2018), which reconstructs inputs using a shallow network such as a single-layer perceptron. R-GAP (Zhu and Blaschko 2020) 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.

The previous works primarily focus on inverting gradients from fully connected networks (FCNs) or convolutional neural networks (CNNs). 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 (Vaswani et al. 2017). As an inherently different architecture, Transformer can build large scale contextual representation models. Inspired by the impressive success in natural language tasks, dozens of works manage to integrate Transformer into various computer vision tasks (Touvron et al. 2021; Liu et al. 2021; Carion et al. 2020; Parmar et al. 2018; Chen et al. 2020). Despite the rapid progress of vision Transformers, 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 (Jieren et al. 2021) provides a optimization-based attack algorithm for a Transformer-based language model, the inherent reason of Transformer's vulnerability is unclear.

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 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: 1. We prove that for the classic self-attention module, if the gradient *w.r.t.*input is known, the input data can be reconstructed in a closed-form manner. 2. We demonstrate that jointly using self-attention and learnable position embedding place the model at severe privacy risk. The attacker can obtain a closed-form solution under certain conditions regardless of the complexity of networks. 3. We propose an Attention Privacy Leakage (APRIL) attack to discover Archilles' Heel. APRIL shows our results superior to SOTA. 4. We suggest to switch the learnbale position embedding to a fixed one as an effective defense against privacy attacks.

APRIL: Attention PRIvacy Leakage

Analytic Gradient Attack on Self-Attention

It has been proved that the closed-form solution for input x can always be perfectly obtained on a fully-connected layer (Phong et al. 2018). In this work, we delve into a more complicated 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:*

$$Qz = q; Kz = k; Vz = v; \tag{2}$$

$$\frac{softmax(q \cdot k^T)}{\sqrt{d_k}} \cdot v = h \tag{3}$$

$$Wh = a; \tag{4}$$

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 is

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

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:

$$\frac{\partial l}{\partial z} = Q^T \frac{\partial l}{\partial q} + K^T \frac{\partial l}{\partial k} + V^T \frac{\partial l}{\partial v} \tag{5}$$

Again, according to the chain rule of derivatives, we can derive the gradients *w.r.t.*Q, K and V from Eq. (2):

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

By multiplying z^T to both sides of Eq. (5) and substituting Eq. (6), we obtain:

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

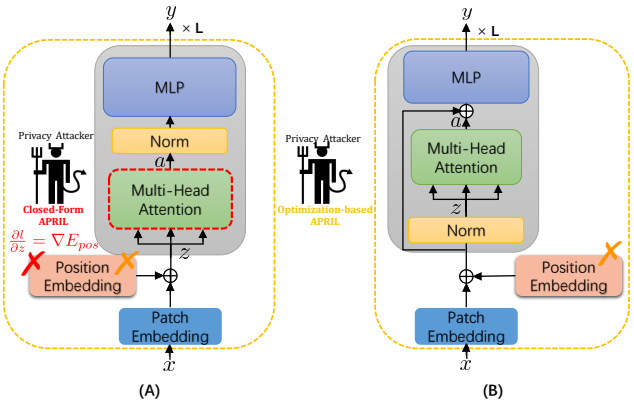


Figure 1: We consider two Transformer designs throughout the paper. (A): Encoder modules stack multi-head attention, normalization, and MLP in VGG-style. (B): A real-world design as introduced in ViT (Dosovitskiy et al. 2020). The architecture in (A) satisfies the precondition of a closed-form APRIL 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).

■

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 of 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.*

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 learnable position embedding:

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*

$$\frac{\partial l}{\partial E_{pos}} = \frac{\partial l}{\partial z} \tag{8}$$

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,

$$z = E_{patch} + E_{pos} \tag{9}$$

Straightforwardly, we compute the derivative of loss *w.r.t.*E_{pos} using Eq. (9), Eq. (8) 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 shared, ∇E_{pos} is available for potential adversaries, which means a successful attack on self-attention inputs.*

While vision Transformers (Dosovitskiy et al. 2020; Wu et al. 2021) embodies 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 to employ fixed position embedding instead of learnable one as a defensive strategy.

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

$$E_{patch} = W_p x \tag{10}$$

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.(Eq. (8)). Arguably, component ∇E_{pos} indicates the directions of the gradients of position embeddings. Intuitively, matching the updating direction of E_{pos} with an direction caused by dummy data can do benefits on the recovery. Therefore, we propose an optimization-based attack with constraints on the direction of ∇E_{pos}.

For expression simicity, we use $\nabla w'$ and ∇w denote the gradients of parameter collections for dummy data and real inputs, respectively. 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

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_G + \alpha \mathcal{L}_A \\ &= \|\nabla w' - \nabla w\|_F^2 - \alpha \cdot \frac{< \nabla E_{pos}, \nabla E'_{pos} >}{\|\nabla E_{pos}\| \cdot \|\nabla E'_{pos}\|}.\end{aligned} \tag{11}$$

where hyperparameter α balances the contributions of two matching losses. Eventually, we set Eq. (11) as another variant of our proposed method, the optimization-based APRIL attack. By enforcing a gradient matching on the learnable position embedding, it is plaguily easy to break privacy in a vision Transformer.

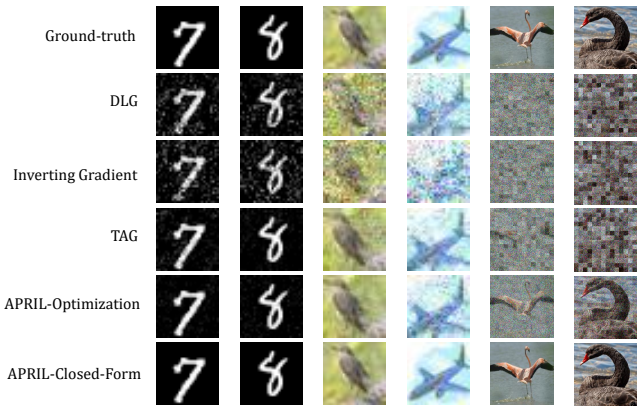


Figure 2: Results for different privacy attacking approaches on Architecture (A).

Experiments

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 (Dosovitskiy et al. 2020), which is composed of encoders each with a normalization layer and residual connection before attention module. For small datasets like CIFAR and MNIST, we refer to the implementation of ViT-CIFAR[1]. For experiments on ImageNet, we follow the original ViT design[2] and architecture setting.

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 showing nearly no difference to the original input, which proves the correctness of our theorem. Comparing optimization-based attacks, for ImageNet reconstructions, DLG, IG and TAG reconstructions have 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.

We further illustrate the optimization procedure of reconstructions in Fig. 3. An apparent observation is that our optimization-based APRIL converges consistently faster than the other two approaches. Besides, APRIL generally ends up with smoother and cleaner image reconstructions.

Apart from visualization results, we also carry out quantitative comparisons on Architecture(B), where we do not have the condition to use closed-form APRIL attack. The statistical results from Tab. 1 shows consistent good performance of optimization-based APRIL, where we obtain best results nearly across every task setting.

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.

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

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

Attack	MNIST		CIFAR-10		ImageNet	
	MSE	SSIM	MSE	SSIM	MSE	SSIM
DLG (Zhu, Liu, and Han 2019)	1.291e-04 ± 2.954e-04	0.997 ± 0.003	0.017 ± 0.009	0.959±0.045	1.328±0.593	0.056 ± 0.027
IG (Geiping et al. 2020)	0.043±0.022	0.833±0.076	0.125±0.102	0.635±0.165	1.671±0.653	0.029±0.013
TAG (Jieren et al. 2021)	**3.438e-05 ± 1.322e-05**	**0.998±0.002**	0.006 ± 0.005	0.965±0.047	1.180 ± 0.473	0.062 ± 0.026
APRIL	4.796e-05±3.593e-05	**0.998 ± 0.002**	**0.002±0.006**	**0.991 ± 0.027**	**1.092±0.663**	**0.099 ± 0.046**

Table 1: Mean and standard deviation for MSE of 500 reconstructions on MNIST, CIFAR-10 and ImageNet validation datasets.

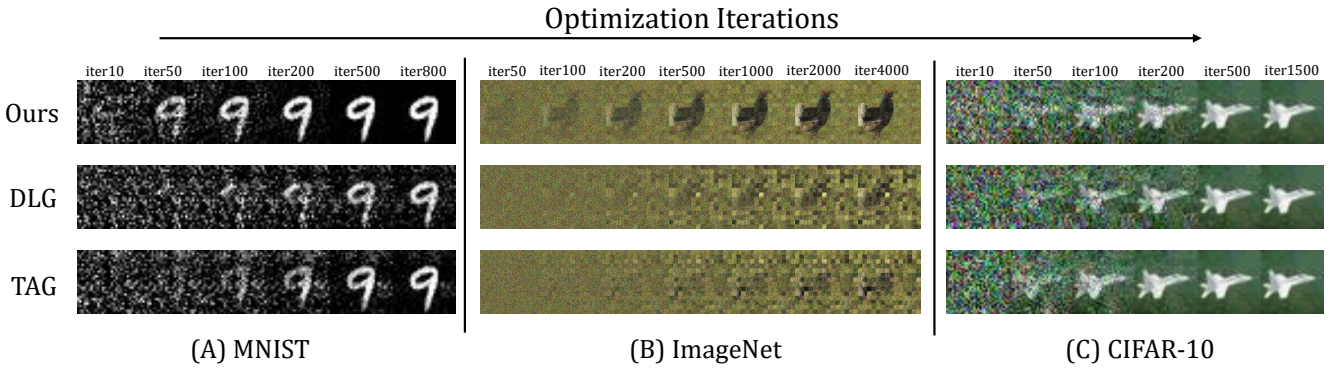
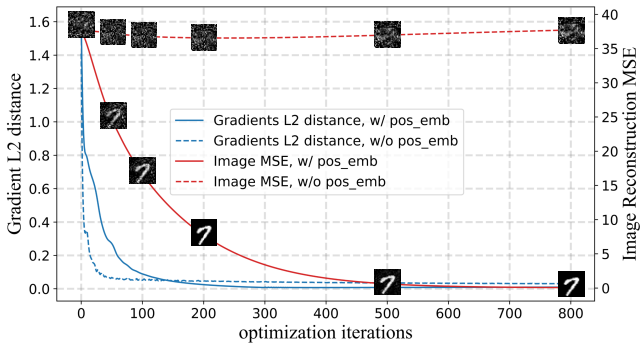
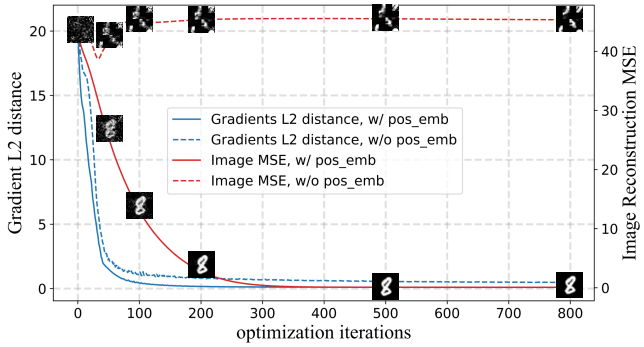


Figure 3: Visualization of the optimization process for optimization-based APRIL, DLG and TAG. Our approach has faster convergence speed and does not easily fall into bad local minima, thus yields a prominently better reconstruction result.



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



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

Figure 4: When position embedding is disabled, matching gradients does not provide semantically meaningful reconstructions.

APRIL-inspired Defense Strategy

A 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.

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. We experimented with optimization-based attacks, and we noticed similar phenomenon as *twin data* mentioned by (Zhu and Blaschko 2020). After ceasing to share position embedding gradients, the optimization results in visually anamorphic data which triggers similar gradients with ground-truth data. The optimization process is shown in Fig. 4. It shows that changing learnable position embedding to fixed ones can result in semantically meaningless reconstructions, which preserves privacy in a highly economic way.

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 the feasibility of both closed-form and optimization-based APRIL attacks in real cases. We further verified the effectiveness of using a fixed position embedding as defense. We hope this work would shed light on privacy-preserving network architecture design.

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*. Springer.

Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; and Sutskever, I. 2020. Generative pretraining from pixels. In *ICML*, 1691–1703. PMLR.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting Gradients - How easy is it to break privacy in federated learning? In *NeurIPS*.

Jieren, D.; Yijue, W.; Chao, S.; Hang, L.; Sanguthevar, R.; and Caiwen, D. 2021. TAG: Gradient Attack on Transformer-based Language Models. In *findings of EMNLP*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *ICCV*.

Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; and Tran, D. 2018. Image transformer. In *ICML*, 4055–4064.

Phong, L. T.; Aono, Y.; Hayashi, T.; Wang, L.; and Moriai, S. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*, 13(5).

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wu, K.; Peng, H.; Chen, M.; Fu, J.; and Chao, H. 2021. Rethinking and improving relative position encoding for vision transformer. In *ICCV*, 10033–10041.

Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J. M.; Kautz, J.; and Molchanov, P. 2021. See through Gradients: Image Batch Recovery via GradInversion. In *CVPR*, 16337–16346.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2020. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 6881–6890.

Zhu, J.; and Blaschko, M. B. 2020. R-GAP: Recursive Gradient Attack on Privacy. In *ICLR*.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep Leakage from Gradients. *NeurIPS*.