Guardian: Guarding against Gradient Leakage with Provable Defense for Federated Learning

Mingyuan Fan
School of Data Science & Engineering,
East China Normal University
Shanghai, China
fmy2660966@gmail.com

Yang Liu
Xidian University
Shanxi, China
bcds2018@foxmail.com

Cen Chen*
School of Data Science & Engineering,
East China Normal University
Shanghai, China
cenchen@dase.ecnu.edu.cn

Chengyu Wang
Alibaba Group
Hangzhou, China
chengyu.wcy@alibaba-inc.com

Minghui Qiu
ByteDance
Shanghai, China
minghuiqiu@gmail.com

Wenmeng Zhou
Alibaba Group
Hangzhou, China
wenmeng.zwm@alibaba-inc.com

ABSTRACT

Federated learning is a privacy-focused learning paradigm, which trains a global model with gradients uploaded from multiple participants, circumventing explicit exposure of private data. However, previous research of gradient leakage attacks suggests that gradients alone are sufficient to reconstruct private data, rendering the privacy protection mechanism of federated learning unreliable. Existing defenses commonly craft transformed gradients based on ground-truth gradients to obfuscate the attacks, but often are less capable of maintaining good model performance together with satisfactory privacy protection. In this paper, we propose a novel yet effective defense framework named *guarding against gradient leakage* (*Guardian*) that produces transformed gradients by jointly optimizing two theoretically-derived metrics associated with gradients for performance maintenance and privacy protection. In this way, the transformed gradients produced via *Guardian* can achieve minimal privacy leakage in theory with the given performance maintenance level. Moreover, we design an ingenious initialization strategy for faster generation of transformed gradients to enhance the practicality of *Guardian* in real-world applications, while demonstrating theoretical convergence of *Guardian* to the performance of the global model. Extensive experiments on various tasks show that, without sacrificing much accuracy, *Guardian* can effectively defend state-of-the-art gradient leakage attacks, compared with the slight effects of baseline defense approaches.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → *Computer vision*; *Machine learning*;

KEYWORDS

Federated Learning, Gradient Leakage Defense, Privacy Protection

ACM Reference Format:
Mingyuan Fan, Yang Liu, Cen Chen, Chengyu Wang, Minghui Qiu, and Wenmeng Zhou. 2024. *Guardian:* Guarding against Gradient Leakage with Provable Defense for Federated Learning. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3616855.3635758

1 INTRODUCTION

Recent years have witnessed the unprecedented development of federated learning (FL) in a variety of privacy-focused scenarios [8, 13]. Through gradient transformation, FL avoids direct upload of private data during the collaborative model learning process. However, not long after the proposal of FL, some works [26, 27] point out that such a privacy protection mechanism of FL is not as reliable as expected. As shown in Figure 1, a malicious server can easily recover the private data of any participant by solving the gradient matching problem, i.e., *gradient leakage attack* [11, 12, 26, 27]. In more detail, with fixed model parameters, the attacker can reconstruct a batch of data points whose corresponding gradients are close to the uploaded ones. Starting from random data points, the reconstruction can be completed in only several rounds of optimization, and the returned data points can be very close to the original ones [18, 21]. Undoubtedly, gradient leakage poses a great threat to the security of FL applications, and an effective countermeasure against the attack is of imperative need.

In response to the rising concerns about FL, there exist two primary approaches, namely encryption-based methods [6, 25] and perturbation-based methods [20, 27]. While encryption-based methods ensure effectiveness, their high computational cost restricts their applicability in many scenarios [13, 15], often leading to significantly slower processing times that can be dozens to hundreds of times longer [6, 25]. In contrast, perturbation-based methods offer a more lightweight and efficient alternative, which garner significant attention in recent research endeavors.

State-of-the-art perturbation-based methods include differential noises [1] and gradient pruning [20, 27]. However, these defenses [1, 10, 19, 20, 27] were broken soon after the introductions. Existing works show that pure differential noises and gradient compression

*Corresponding author.

WSDM '24, March 4–8, 2024, Merida, Mexico.

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635758

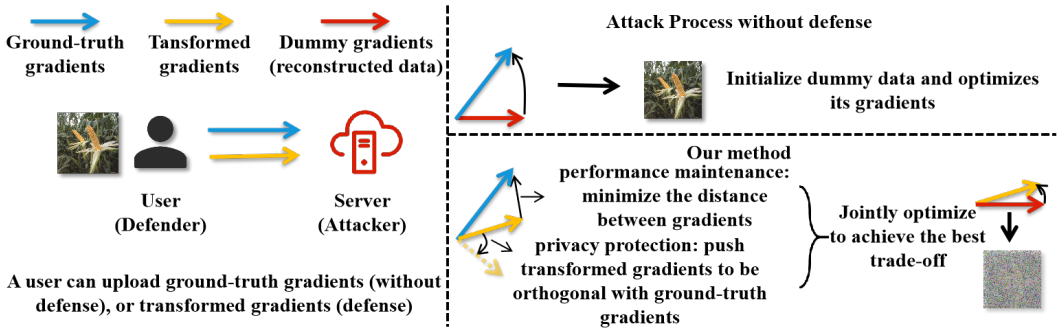

Figure 1: The sketch picture of our method.

are not resilient against state-of-the-art attacks [11, 12, 21, 24, 26]. To mitigate the problem, more recently proposed works [10, 19, 20] introduce the heuristic defense by mixing multiple gradient transformation strategies. Again, Mislav *et. al.* [2] effortlessly breaches these advanced defenses by introducing Bayes optimal adversary, and shows that an adversary can always explore an effective way against most of the existing defenses based on the Bayes rules. Therefore, the creation of a more rigorous and theory-backed defense against general gradient leakage attacks is still deemed to be an open problem for now [2].

We propose a defense *guarding against gradient leakage* (*Guardian*) that can significantly maintain model performance while achieving less privacy leakage compared with existing works. Specifically, our design is inspired by the observation: the original data can be reconstructed from their corresponding gradients because the specific mapping relationships between them can be easily reversed by solving the gradient matching problem [27]. Then, a reasonable idea to block the attacks is adding perturbations to obfuscate the relationships. However, the dilemma lies in that perturbation is a delicate work of art in practice [7, 22, 23]. Excessively added perturbations easily make gradients no longer informative while mild ones are unable to provide enough defense effect [21, 27]. To get rid of the dilemma, *Guardian* leverages two deftly designed metrics, i.e., Performance Maintenance Metric (PMM) and Privacy Protection Metric (PPM). Theoretically, these metrics are proved to be able to measure the model performance change and privacy leakage risk, respectively. By jointly optimizing the two metrics (Figure 1), *Guardian* can always approximate to the optimal perturbation point where privacy is well protected with minimal model performance degradation. Also, considering the overhead of defense is of vital importance to practice, we design a better initialization strategy to substantially accelerate the convergence of *Guardian*, resulting to that the computational cost of *Guardian* becomes competitive among the state-of-the-art methods. Furthermore, we theoretically and experimentally analyse the convergence of *Guardian*, and study the potential factors that affect the effectiveness of existing defenses, including security assumptions and etc. Our contributions:

- Built upon solid theory foundations, we design a novel yet effective defense method called *Guardian*, which can maintain model performance while minimizing privacy leakage risk by jointly optimizing two deftly designed metrics, PMM and PPM.
- We conduct an in-depth theoretical analysis for *Guardian* including convergence guarantee, security assumptions, etc., and propose a better initialization strategy for *Guardian*.
- To the best of our knowledge, *Guardian* is the only method whose defense is still effective even in the most critical scenarios suggested by [2], i.e., white-box scenarios with the Bayes optimal adversary, and this is examined across diverse tasks, implying great generalizability of *Guardian* in real-world applications. Besides, we are the first to conduct gradient leakage attacks and defenses on the vision transformer architecture.

2 RELATED WORK

Gradient Leakage Attacks. With the increasing concerns of users' privacy, FL [16] was introduced to train a deep neural network without directly sharing data. However, Mislav *et al.* [27] attempted to examine the effectiveness of the privacy protection mechanism of FL and proposed gradient leakage attack (GLA) to breach the mechanism. GLA reveals the private information by solving a gradient matching problem (detailed in Section 3). At the beginning, GLA supposed the ground-truth labels to be unknown, which degraded the attack effectiveness. Latter, inference gradient leakage attack (iGLA) [26] proposed that the attacker could steal the labels by an analytical procedure, thereby significantly enhancing the attack performance. However, GLA and iGLA are only effective on small batches (e.g., fewer than 8 data instances a batch) [21, 24, 26, 27]. Follow-up works attempt to relax this constraint by using handcrafted input-regularization (prior knowledge) [11] and smarter initialization [21], etc.

Gradient Leakage Defenses. In response to the rise of gradient leakage attacks, several defenses [1, 9, 20] are developed. The core idea of these defenses is imposing noises into ground-truth gradients to obfuscate the attacker. As the most known privacy-protection method, differential privacy [1, 2, 27] was introduced to resist the attacks, which adds Gaussian or Laplace noises into the gradients to resist gradient leakage attacks. In addition, gradient compression also was commonly adopted by [11, 24, 26, 27] as a baseline defense to examine the robustness of attacks. Recently, Soteria [20] leveraged a similar idea to gradient compression but with a smarter prune strategy, so as to decrease privacy leakage risks with similar performance maintenance. However, a recent work [2] evaluated the so-called effective defenses and broke them

by introducing a Bayes optimal adversary. Therefore, developing an effective defense against such attacks is imperative for now.

3 SCENARIO DESIGN

In this paper, we design a defense method against general gradient leakage attacks. To this end, we introduce a rather harsh scenario to validate whether a defense method is truly effective.

Scenario description. In FL, a user (i.e., the defender) uploads local gradients $\nabla_\theta L(F_\theta(x), y)$ to the server (i.e., attacker), where L, F, x, y denote the loss function, the training model parameterized by θ, a batch of training data and labels with size n, respectively. For the simplicity of symbols, in the remainder of this paper, we use $L(x, \theta)$ to denote $L(F_\theta(x), y)$, i.e., omitting y and F. The attacker attempts to solve an optimization problem (i.e., gradient matching) to reconstruct user data as follows:

$$x^* = \arg\min_{x^*} Dist(\nabla_\theta L(x, \theta), \nabla_\theta L(x^*, \theta)) + R(x^*), \tag{1}$$

where x^* are reconstructed data[1], and $Dist(\cdot, \cdot)$ and $R(\cdot)$ represent a certain distance between gradients and regularization items for reconstructed data, like Euclidean distance [27] and total variance [11], respectively.

Attacker's abilities. To better evaluate the effectiveness of our defense methods, we assume the attacker with the most powerful ability suggested by [2]. The attacker can have full knowledge about the target model information and the defender's defense strategy, including loss functions, hyperparameters, defense methods, etc. More precisely, except original training data, the attacker is allowed to access any other information to launch any attacks in polynomial time. In addition, the attacker, as the server, can also proactively select "weak" hyperparameters to advantage its attacks, e.g., setting the batch size $n = 1$ as shown in [18, 26, 27][2].

Defender's goals. The defender aims to alleviate the risk of the server to derive user private data from uploaded gradients. To achieve the goal, the defender searches for perturbations δ^* to obfuscate the mapping information between gradients and original data. We call the perturbed gradients as *transformed gradients*. The ideal transformed gradients should at least have the following three properties:

- *Performance maintenance.* Transformed gradients should be as informative as the original gradients to maintain the target model performance in FL.
- *Privacy protection.* Transformed gradients should ensure that the attacker cannot reconstruct data whose distance between the original data is smaller than a given threshold ε_0.
- *Practicality in applications.* To ensure practicality, the generation of transformed gradients should cost as few computational resources as possible for the defender.

4 APPROACH

We formulate a loss function involving two metrics that can measure the utility and privacy leakage risk of gradients, respectively.

[1]Commonly, the attacker can analytically steal the ground-truth labels by adopting label inference technique introduced by [26]. Thus, instead of labels, most data leakage attacks focus more on data reconstruction.

[2]The larger the batch size is, the harder it is to find x'. Intuitively, the introduction of more data into a single batch makes the attacker need to search x' in a higher dimensional space, i.e., more epochs required to make Eq. 1 converge.

Table 1: We record the running time for obtaining PMM with different methods. The iteration number is fixed to be 20.

Model	$L(x, \theta + \delta)$	Our	Speedup
ResNet18	6.06s	64.9ms	93.37x faster
ResNet34	11.6s	74.8ms	155.08x faster
ResNet50	18.5s	80.5ms	228.96x faster
ResNet101	33.6s	87.8ms	382.69x faster

By optimizing the loss, the defender can efficiently find a series of crafted perturbations to obfuscate gradients with minimized model performance degradation.

4.1 PMM: Measurement of Performance Improvement

We first give the metric to measure how much the perturbed gradients can contribute to the model performance improvement. With this metric, the generated perturbation is constrained to not bias too much from the desired convergence direction. The derivation of the metric is based on the expansion of gradient descent based optimization methods [3]. Given the parameter updates $-\delta$, the derivation proceeds as follows:

$$L(x, \theta - \delta) = L(x, \theta) - \nabla_\theta L(x, \theta)^T \delta + O(||\delta||), \tag{2}$$

where $F_\theta(\cdot)$ [3] is assumed to be differentiable. Moreover, to make the above expansion feasible, we set $||\delta|| \leq \epsilon$ where ϵ is assumed to be small enough for avoiding the error induced by ignoring the remainder $O(||\delta||)$. In fact, ϵ can be regarded as learning rate and generally set into $10^{-5} \sim 10^{-2}$ in practice. Then, the best solution for δ to minimize L is $\epsilon \frac{\nabla_\theta L(x,\theta)}{||\nabla_\theta L(x,\theta)||}$ [3]. As shown in Equation 2, the reduced loss before and after parameter updating can be accurately approximated by $\epsilon \nabla_\theta L(x, \theta)^T \frac{\nabla_\theta L(x,\theta)}{||\nabla_\theta L(x,\theta)||}$. Similarly, if δ is set equal to the transformed gradients δ^* ($||\delta^*|| \leq \epsilon$), the reduced loss associated with δ^* is $\nabla_\theta L(x, \theta)^T \delta^*$. Therefore, we can do the following computations to estimate the performance contribution change after gradients are transformed.

$$\rho_{PMM} = ||\epsilon \nabla_\theta L(x, \theta)^T \frac{\nabla_\theta L(x, \theta)}{||\nabla_\theta L(x, \theta)||} - \nabla_\theta L(x, \theta)^T \delta^*||_2. \tag{3}$$

In other words, Equation 3 defines the *performance maintenance metric* (PMM) used in *Guardian*. Intuitively, the lower PMM is, the more δ^* contributes to the model performance.

In fact, an alternative to define PMM is directly optimizing $L(x, \theta - \delta)$, i.e., searching optimal parameter perturbations δ to minimize $L(x, \theta - \delta)$. However, two reasons make us discard such an idea. First, as shown in Table 1, the computation of $L(x, \theta - \delta)$ is much slower than our methods because the searching process of δ needs to be conducted in multiple iterations over the whole model, while ρ_{PMM} can be directly obtained with past gradients. Second, iterative optimizations to search δ can easily make the loss of the target model about x to vanish, i.e., causing the overfitting of x.

[3]Note, for symbol simplicity, in this paper, we use $L(x, \theta)$ to denote $L(F_\theta(x), y)$, i.e., omitting y and F.

4.2 PPM: Measurement of Privacy Leakage

Let x^* denote the data reconstructed from the transformed gradient δ^* by solving the Equation 1, where δ^* is assumed to be $\nabla_\theta L(x^*, \theta)$ like prior works [18, 21, 27]. To protect the participant's privacy, the defender has to maximize the distance between x and x^*, i.e., $||x - x^*||_2$. Thus, an intuitive way to measure the privacy leakage risk is to find a general function that can evaluate the change of $||x-x^*||_2$ with δ^*. However, considering that deep neural networks per se are highly non-linear and non-convex functions, finding such a function that precisely models the relationship between $||x-x^*||_2$ and δ^* is intractable, particularly when the training process of neural networks is constantly changed. Therefore, to circumvent this problem, a feasible idea is to construct a lower bound associated with δ^* of $||x-x^*||_2$. Then, as the lower bound increases, the privacy leakage risk decreases, and vice versa. In mathematics, a common tool used to tightly bound $||x-x^*||_2$ is the Lipschitz coefficients and the overwhelming majority of neural networks satisfy the Lipschitz Assumption 4.1 in practice. But as shown in Assumption 4.1, the lower bound of $||x - x^*||_2$ is not associated with δ^* and we have to make some transformation in the lower bound. The lower bound is the output difference of the model w.r.t. x and x^* and there is an intuitive idea to guide us: the output difference will be raised if the output of the model for x^* is unchanged when the model steps along the direction that enables the output of the model for x changed sharply, i.e., x and x^* are treated to be orthogonal from the model output perspective. Interestingly, the direction commonly accords with the gradient convergence direction [3], and thus, we can establish the relationship between the difference of x and x^* and their gradients by harnessing the idea. Theorem 4.1 below implements the idea and reveals that $||x - x^*||_2 \propto -\frac{\nabla_\theta L(x,\theta)^T}{||\nabla_\theta L(x,\theta)||_2}\frac{\delta^*}{||\delta^*||_2}$. Note that we force $\nabla_\theta L(x,\theta)^T\delta^* \geq 0$ to guarantee model convergence according to Theorem 4.2 (in section 4.4), this also can be intuitively realized by that, the loss function decreases if $\nabla_\theta L(x,\theta)^T\delta^* \geq 0$ in Equation 2. Therefore, we can evaluate the *privacy protection metric* (PPM) by computing:

$$\rho_{PPM} = |\frac{\nabla_\theta L(x,\theta)^T\delta^*}{||\nabla_\theta L(x,\theta)||_2||\delta^*||_2}|. \tag{4}$$

Assumption 4.1. *Let L, F satisfy the Lipschitz condition. There exists a concrete positive real number α, which makes the following inequality relationship stand for $\forall a, b, \theta$:*

$$||L(a,\theta) - L(b,\theta)||_2 \leq \alpha||a-b||_2.$$

Theorem 4.1. *Let Assumption 4.1 stand. Given a fixed x, a variable x^*, and $\nabla_\theta L(x,\theta)\cdot\delta^* \geq 0$, the lower bound between x and x^* is negatively correlated with the cosine distance between $\nabla_\theta L(x,\theta)$ and δ^*. Specifically, if $\nabla_\theta L(x,\theta)\cdot\delta^* = 0$, the lower bound between x and x^* can be maximized.*

4.3 Putting All into One

We now formulate the final optimization problem for defense against gradient leakage attack. First, to maintain model performance, its corresponding measurement metric PMM should be minimized. Then, we have to minimize the gradient leakage attack risk PPM.

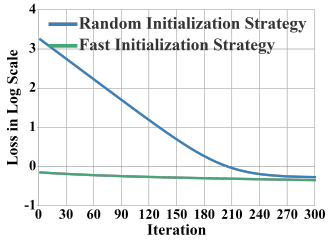

Figure 2: Convergence of *Guardian* with different initialization strategies.

In all, the final optimization loss can be expressed as:

$$\begin{aligned}\arg\min_{\delta^*} &||\epsilon\nabla_\theta L(x,\theta)^T\frac{\nabla_\theta L(x,\theta)}{||\nabla_\theta L(x,\theta)||} - \nabla_\theta L(x,\theta)^T\delta^*||_2 \\ &+ \beta\,|\frac{\nabla_\theta L(x,\theta)^T\delta^*}{||\nabla_\theta L(x,\theta)||_2||\delta^*||_2}|,\ \beta \geq 0,\ ||\delta^*|| \leq \epsilon,\end{aligned} \tag{5}$$

where β is served as a balance factor. In Equation 5, the left term is minimal when $\epsilon\frac{\nabla_\theta L(x,\theta)}{||\nabla_\theta L(x,\theta)||}$, i.e., gradients without perturbation. The right term aims to increase the cosine distance between δ^* and $\nabla_\theta L(x,\theta)$ by optimizing δ^* to be orthogonal with $\nabla_\theta L(x,\theta)$. Note that Equation 5 demonstrates privacy protection has to come at a price of model performance. However, compared with prior works [1, 20], our method allows the defender to pay less "price" by utilizing more precise metrics to evaluate the losses and gains simultaneously. Moreover, Equation 5 is a typical convex optimization task with constraints that can be well solved by optimization methods, e.g., projected gradient descent [3].

Better initialization for faster convergence. Besides method effectiveness, convergence speed also counts a lot for the practicality of defenses. In our evaluation, we discover that the commonly used random initialization strategy [1] does not suit for *Guardian* very well, shown in Figure 2. To get rid of the issue, *Guardian* adopts a faster convergence strategy.

Reconsider that starting an optimization task from a point near the optimal one can generally reduce the required iterations to converge. Here, the goal of defender is to minimize the utility degradation of transformed gradients caused by perturbations. Following the idea, the optimal convergence point for *Guardian* can be always around the original gradients. Thus, instead of random noises [1], a better initialization strategy for *Guardian* is to leverage the original gradients as the initial point. Figure 2 validates that such an initialization strategy can increase the convergence speed of *Guardian* significantly.

4.4 Convergence Analysis

In this subsection, we derive the convergence guarantee of the model updated with transformed gradients obtained by Equation 5. Following the existing works [20, 21], We assume that the model is Lipschitz gradient continuity with coefficient of τ. Then, we can obtain convergence guarantee of *Guardian*, as shown in Theorem 4.2.

Theorem 4.2. *Let θ^* denote the optimal parameters of the model over x. Let the iteration number be k and the constant update step*

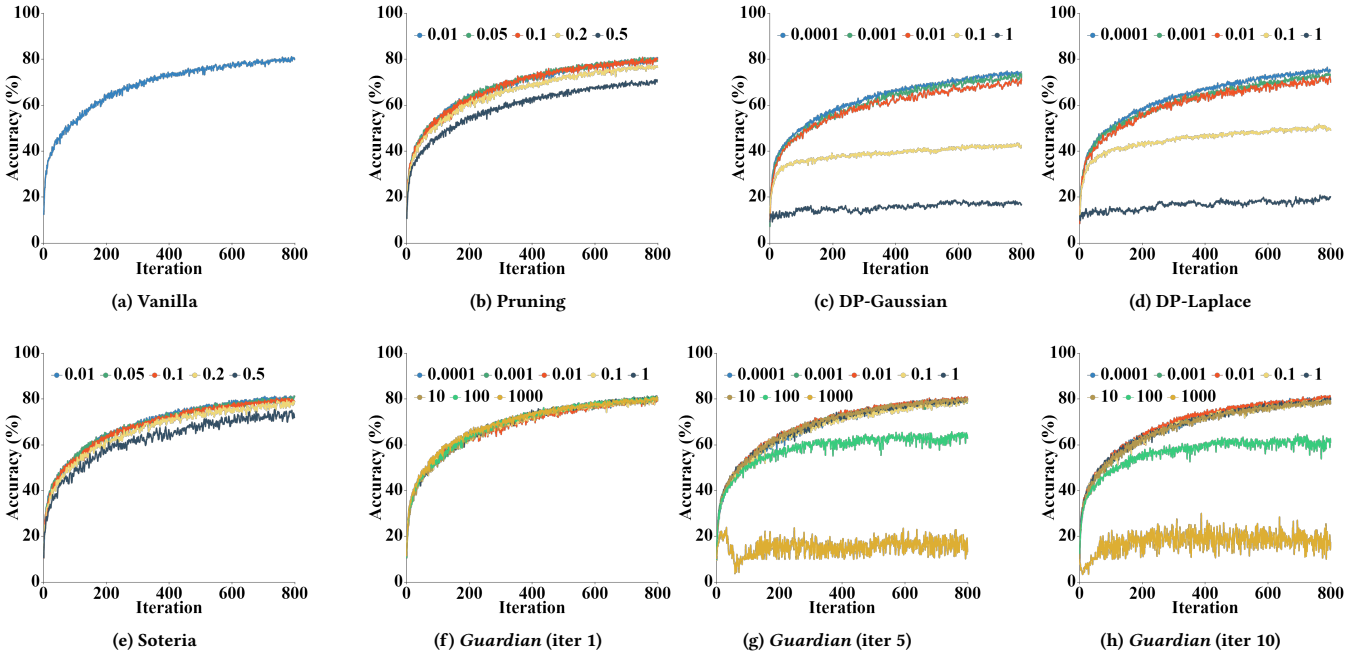

Figure 3: Accuracy of model with various defenses over different iterations in CIFAR-10. We tune the pruning rate for Pruning and Soteria and noise magnitude for DP-Gaussian and DP-Laplace. We tune β and iter (i.e., iteration) for solving Equation 5.

size $\eta \leq \frac{1}{\tau}$. *Suppose the model is updated using* $\delta_t^* = \nabla_{\theta_t} L(x, \theta_t) + \delta_t$ *in the* t*-th* ($t = 0, 1, \cdots, k-1$) *iteration where* δ_t^* *denotes the optimal solution of Equation 5, i.e.,* δ_t *is the crafted gradient perturbations. If* $\nabla_{\theta_t} L(x, \theta_t)^T \delta_t \geq 0$, *and* $||\nabla_{\theta_t} L(x, \theta_t)||_2^2 \geq ||\delta_t||_2^2$ *for* $t = 0, 1, \cdots, k-1$, *the following inequality relationship holds:*

$$L(x, \theta_k) \leq L(x, \theta^*) + \frac{1}{2k\eta} ||\theta_0 - \theta^*||_2^2 + \eta ||\delta_{max}||_2^2, \tag{6}$$

where $||\delta_{max}||_2^2 = max\{||\delta_0||_2^2, \cdots, ||\delta_{k-1}||_2^2\}$. *In particular, if* k *is large enough, compared to the optimal solution* θ^*, *the extra loss induced by our method is no more than* $\eta ||\delta_{max}||_2^2$, *i.e.,*

$$\begin{aligned} \lim_{k \to +\infty} L(x, \theta_k) &\leq \lim_{k \to +\infty} (L(x, \theta^*) + \frac{1}{2k\eta} ||\theta_0 - \theta^*||_2^2 + \eta ||\delta_{max}||_2^2) \\ &= L(x, \theta^*) + \eta ||\delta_{max}||_2^2. \end{aligned} \tag{7}$$

4.5 Comparison with Other Defense Methods

Here, we discuss the relationships among existing defense methods and the possibility of the defenses being broken by the white-box attacker [2]. In fact, existing defense approaches resist gradient leakage attacks by transforming ground-truth gradients into new gradients, i.e., dubbed transformed gradients, to upload, and the transformed gradients may fool the attacker to recover false data. If $\mathcal{M}$ denotes the gradient transformation function, existing defenses can be reduced to concretize $\mathcal{M}$ in different fashions. We give a handful of examples to illustrate this point:

- For differential privacy [1], $\mathcal{M}$ first normalizes the ground-truth gradients and then adds random noises, following a certain distribution such as Gaussian distribution, into the normalized gradients to produce transformed gradients.
- For gradient compression [27], $\mathcal{M}$ discards the elements below a certain threshold in the ground-truth gradients to generate transformed gradients.
- For Soteria [20], similar to gradient compression, $\mathcal{M}$ removes some specified gradients in the fully-connection layer of the model via analytically solving an optimization task.
- For *Guardian*, $\mathcal{M}$ optimizes Equation 5 to craft transformed gradients.

Security discussion. In the white-box scenario, the attacker is allowed to access full information about $\mathcal{M}$, e.g., the distribution for differential privacy. For differential privacy and gradient compression, the attacker can adopt Bayes gradient strategy [2] to break the defenses. Soteria only optimizes the gradients in the (last) fully-connected layer, indicating the gradients in other layers are identical to ground-truth gradients. Therefore, the attacker can evade the defense Soteria by resetting weights of gradients in the fully-connected layer in Equation 1 as zero. Now we consider *Guardian*. On the one hand, we empirically validate that *Guardian* is effective against gradient leakage attacks with Bayes gradient strategy. On the other hand, adopting the attack strategy similar to Soteria cannot break *Guardian*, because *Guardian* optimizes all gradients, not gradients belonging to one or several specified layers. In addition, $\mathcal{M}$ defined in *Guardian* (Equation 5) is not reversible. Even if the attacker knows the final loss value, it still cannot reverse the ground-truth gradients, as the equation is under-determined (there are infinite feasible solutions for solving it).

5 EXPERIMENTAL EVALUATION

5.1 Setup

Attack methods. We select four strong attack methods, including GLA [27], iGLA [26], InvertingGradients [11], and GradInversion [24]. To further raise the attack ability, we also equip these attacks with techniques that can improve the fidelity of recovered data, such as input regularization, smarter initialization, and normalization. To accurately evaluate the defense performance, we raise the attack iterations from 200 for GLA and iGLA and 1000 for InvertingGradients and GradInversion to 10000. *Notice that, by default, each attack-defense pair is equipped with the corresponding Bayes optimal strategy with the same setting to [2] throughout the experiments.*

Competitors. We compare *Guardian* with various state-of-the-art defenses as follows: 1) gradient compression (Pruning) [27] that discards a certain percentage of gradients with the lowest magnitude, 2) differential privacy (DP) [1] that injects certain random noises to the gradients to fool the attacker and the gradient norms are clipped into 1 by default following [2], and 3) Soteria [20] that perturbs data representation in the final fully-connected layer[4]. Moreover, we also set ϵ in Equation 5 to 1.

Hyperparameter configurations. For fair comparisons, we first conduct experiments on two benchmark datasets MNIST with LeNet and CIFAR-10 with ResNet18 [21] to search the optimal hyperparameters, which are used throughout the experiments. Following the original papers [11, 24, 26, 27], GLA and iGLA leverage the L-BFGS with a learning rate of 1 while InvertingGradients and GradInversion use Adam with a learning rate of 0.1. Moreover, we examine the effectiveness of defenses in the early training process, as gradient leakage attacks perform better in the stage [18].

Evaluation metrics. Two commonly used metrics [18, 21, 27] are considered: PSNR and SSIM. PSNR is the logarithmic L_2 distance between original and recovered images, while SSIM computes the structural similarity between two images. The lower PSNR or SSIM is, the higher the defense effect is.

5.2 Hyperparameter Search

We first examine the impact of *Guardian* on the performance of the global model. Following [20], in each communication round (iteration), 5 participants are randomly selected from all ones (10 participants) to upload locally-computed gradients with batch size of 32. The server averagely aggregates the uploaded gradients and updates the global model with a learning rate of 0.01. Figure 3 shows the accuracy of defenses with varying hyperparameters over different iterations in CIFAR-10. For fair comparisons, we select the highest (optimal defense effect) hyperparameters within the range that the model can achieve similar performance compared with one without defenses for CIFAR-10. Specifically, in the following experiments, we use pruning rate of 0.2 for Pruning, noise magnitude of 0.001 for both DP-Gaussian and DP-Laplace, L_0-norm constraint of 0.2 for Soteria, and β of 10 and epoch of 10 for our defense. Notice that, due to page restrictions, we only provide empirical convergence results in CIFAR-10, but we highlight that results in MNIST and for other model architectures also empirically show that these hyperparameters are the best in terms of the choices of hyperparameters.

[4]Since we do not find the official implementation of Soteria, we resort to a third-party implementation.

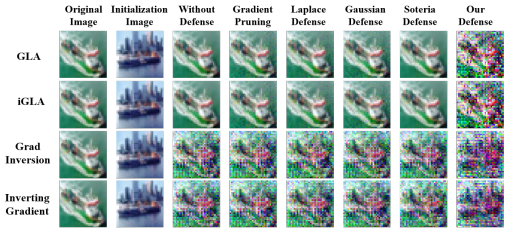

Figure 4: Visualization of different defenses.

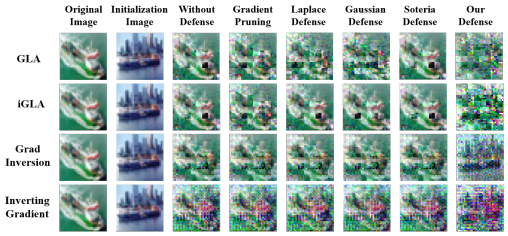

Figure 5: Defense display on Vision Transformer.

5.3 Comparison with State-of-the-art

Table 2 and Table 3 report the defense performance, measured by PSNR and SSIM, of *Guardian* with four baseline defenses against four attacks on MNIST and CIFAR-10. There are two key observations. First, compared with other defenses, *Guardian* can effectively lower the attack performance of these advanced attacks. In fact, we notice that baseline defenses only present negligible defense effectiveness, as these defenses hardly reduce PSNR or SSIM scores, which also is validated in [2]. To perceptually demonstrate the effectiveness of *Guardian* compared with baselines, Figure 4 visualizes the reconstructed images with different defenses for a randomly selected image instance. We observe the overall semantic information in the original image can be exactly reconstructed by four attacks against baseline defenses; whereas, the attacks only can steal a little trivial information under our defense. Second, the fidelity of images recovered by InvertingGradients and GradInversion is worse than GLA and iGLA. This is because, GLA and iGLA adopt L_2-norm distance between gradients as the loss function, which is a better option over negative cosine distance in our case. In the early training stage, the convolution filters fail to effectively capture the semantic information contained in the inputs, i.e., gradient direction does not make sense. As a result, only encouraging alignment between dummy gradients and uploaded gradients is not sufficient to recover ground-truth images.

5.4 Generality Validation for Transformer

***Guardian* on Vision Transformer [5].** Table 4 reports PSNR and SSIM of *Guardian* compared with baselines. Overall, the performance of *Guardian* surpasses baselines by a huge margin. Besides, we observe that the attack effectiveness on Vision Transformer

Table 2: The defense effectiveness of different methods measured by SSIM and PSNR on MNIST.

Defense	Vanilla		Pruning		DP-Gaussian		DP-Laplace		Soteria		Our	
Metric	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
GLA	0.90	20.20	0.87	19.04	0.88	19.88	0.87	19.22	0.85	18.59	**0.66**	**10.77**
iGLA	0.91	20.60	0.90	19.87	0.90	20.03	0.90	19.21	0.89	18.88	**0.62**	**10.98**
InvertingGradients	0.85	18.12	0.79	16.94	0.78	16.87	0.78	17.57	0.76	16.89	**0.62**	**9.70**
GradInversion	0.88	18.85	0.83	17.68	0.81	18.70	0.80	18.80	0.76	17.03	**0.61**	**10.55**

Table 3: The defense effectiveness of different methods measured by SSIM and PSNR on CIFAR-10.

Defense	Vanilla		Pruning		DP-Gaussian		DP-Laplace		Soteria		*Guardian* (Our)	
Metric	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
GLA	0.91	14.82	0.81	11.67	0.88	13.43	0.84	12.09	0.89	11.23	**0.53**	**6.58**
iGLA	0.92	15.16	0.83	12.15	0.90	13.62	0.86	12.32	0.91	11.53	**0.55**	**6.64**
InvertingGradients	0.83	12.29	0.77	10.60	0.85	12.13	0.82	12.07	0.84	10.53	**0.55**	**5.65**
GradInversion	0.86	13.09	0.79	11.45	0.88	12.72	0.84	12.62	0.87	10.71	**0.57**	**6.37**

Table 4: The effectiveness of different defenses with Vision Transformer on CIFAR-10.

Defense	Vanilla		Pruning		DP-Gaussian		DP-Laplace		Soteria		*Guardian* (Our)	
Metric	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
GLA	0.65	7.81	0.55	5.65	0.58	7.37	0.57	6.97	0.56	7.65	**0.22**	**2.64**
iGLA	0.67	7.94	0.57	5.73	0.64	7.70	0.63	7.14	0.59	7.68	**0.24**	**2.88**
InvertingGradients	0.32	3.28	0.24	2.33	0.21	2.20	0.19	1.96	0.22	2.70	**0.06**	**0.99**
GradInversion	0.34	3.34	0.28	2.42	0.26	2.66	0.30	2.24	0.25	3.31	**0.07**	**-0.01**

Table 5: The effectiveness of different defenses for text classification task with BERT model.

Attack	Defense	Vanilla	Pruning	DP-Gaussian	DP-Laplace	Soteria	*Guardian* (Our)
iGLA	Precision	34.07	32.02	32.82	30.20	29.66	**25.81**
	Recall	33.31	31.59	33.00	29.03	28.12	**27.16**
	F1	33.69	31.81	32.91	29.62	28.89	**26.49**
TAG	Precision	32.25	32.33	31.06	30.99	30.30	**26.78**
	Recall	31.93	32.11	30.50	30.25	29.85	**26.82**
	F1	32.09	32.22	30.78	30.62	30.08	**26.80**
SIM	Precision	52.01	51.90	50.92	52.09	51.21	**41.41**
	Recall	48.59	48.27	47.80	48.76	47.99	**38.15**
	F1	50.30	50.09	49.36	50.42	49.60	**39.78**

Table 6: The trade-off between model accuracy and privacy protection of different methods against iGLA in CIFAR-10.

Guardian	ACC	80.36	79.88	79.67	64.73	30.01
	PSNR	10.02	9.57	6.64	2.00	0.11
Soteria	ACC	80.33	80.21	78.83	77.40	71.09
	PSNR	11.53	11.61	11.30	11.82	11.52
DP-Gaussian	ACC	74.83	74.96	73.72	43.47	18.66
	PSNR	18.38	13.62	7.31	3.61	1.59
DP-Laplace	ACC	76.47	74.08	74.20	51.64	20.82
	PSNR	17.86	12.32	7.04	3.42	0.73

is seemingly weaker compared to convolutional architectures. To understand the reason behind it, we visualize the recovered images shown in Figure 5. Wherein, the important semantic information is indeed revealed, but the images are reconstructed by some block-like fragments in improper orders. We speculate that this probably is caused by that the inputs for vision transformer are commonly pre-processed to be cut into non-overlap patches. Furthermore, attacks can effectively reconstruct each patch but fail in arranging the location of these patches, resulting in poor attack performance.

Table 7: The performance of different methods against iGLA in Non-IID setting (CIFAR-10).

DP-Gaussian	ACC	74.63	72.71	71.23	63.27	19.92
	PSNR	18.31	13.79	7.15	3.37	1.36
DP-Laplace	ACC	74.80	73.24	71.59	64.10	20.17
	PSNR	17.79	12.68	6.91	3.75	1.08
Soteria	ACC	74.48	74.42	71.87	67.78	65.86
	PSNR	11.50	11.08	11.59	11.44	11.46
Guardian	ACC	74.37	73.96	73.63	59.58	27.31
	PSNR	10.09	9.44	6.33	1.24	0.35

***Guardian* on BERT.** For NLP tasks, the metrics for CV tasks are no longer suitable and we leverage precision, recall, and F1 scores to evaluate the defense performance. Precision measures the ability of attacks to identify only the relevant words while recall measures the ability of attacks to find all the words within the ground-truth sentence. F1 is the average value of precision and recall scores. Moreover, following [4], we evaluate the performance of defenses on SST-2 dataset for language models BERT. In addition, some gradient leakage attacks for CV is not available to NLP tasks. Therefore, aside from using iGLA and a recently-proposed attack method called TAG [4], we replace L_2-norm loss function in iGLA with the cosine distance loss function to form a new attack method called SIM to evaluate the effectiveness of defenses. Table 5 reports the performance of defenses over three attacks for BERT. As can be seen in Table 5, *Guardian* obtains better performance, i.e., lower precision, recall, and F1 metrics (about 10% drop), against three attacks. In contrast, other defenses only slightly lower the three metrics by around 1% to 4%. Moreover, the attack effectiveness of SIM significantly surpasses other attacks, since parameters of pre-trained model contain general knowledge, and increasing the

Table 8: The two reconstructed examples via SIM under different defenses against BERT. The texts in red color indicate they appear in the original sentence.

	Example 1 (short sentence)	Example 2 (long sentence)
Original	it's a charming and often affecting journey.	even horror fans will most likely not find what they're seeking with trouble every day;the movie lacks both thrills and humor.
Vanilla	a affecting often charming affecting s arc journey it a and affecting wildly	probable the trouble tortricidae of not almost triple what not fans rid find will particular trouble both even day
Pruning	umm charming a the and affecting often it charming s journey affecting journey	wrongly even will on fans sight not trouble horror lara each what what find hammer whatever most
DP-Gaussian	a often affecting it affecting erinaand seas. charming a that charming journey	horror hope clive likely trouble day us even what not cinema lacks a ind will humor
DP-Laplace	and affecting it journey premises charming contemporary journey charming charming	fans likely with not thrills ua dumont likely will fewer find trouble even undertaker what are will
Guardian (our)	is your cut ' Luton appropriately affecting differently upside it	say strait horror dessert guiana ad till smart every solve up even not location albeit chiefs chimney day dating save orbit will

alignment between gradients makes more sense. We also show the recovered sentences by SIM (best attack performance) in Table 8.

5.5 Privacy-Utility Trade-off

Here we are interested in whether a better trade-off between model performance and privacy protection can be obtained by *Guardian* compared with baselines when varying hyperparameters (defense magnitude). We tune β for *Guardian* with fixed iteration over {0.1,1,10,100,1000}, pruning rate of Soteria over {0.01,0.05,0.1,0.2,0.5}, noise magnitude for DP-Gaussian and DP-Laplace over $\{10^{-4}, \cdots, 1\}$ (these hyperparameters correspond to the results in Table 6 from left to right) and report defense results against iGLA in Table 6. Compared with DP-Gaussian and DP-Laplace, *Guardian* consistently obtains better trade-offs, i.e., lower PSNR with better (or similar) accuracy. Notice that, as shown in [2], Soteria can be easily evaded by muting the gradients of the fully-connected layer in gradient matching problems, i.e., Soteria only can achieve a negligible defense performance.

5.6 Model Performance in Non-IID setting

We endeavor to examine the effectiveness of different defenses in maintaining model performance in Non-IID setting. To do so, we reuse the training setup in Section 5.2 and solely modify the data distribution of participants. In particular, the data distribution is randomly generated by a symmetric Dirichlet distribution with a concentration parameter of 1 [14, 17]. Following the determination of the chosen distribution, we proceed to run various defense strategies. The accuracy of the model without defenses is about 74.85%, and Table 7 reports the results of different defenses against iGLA in Non-IID setting. Similar to the conclusion in Section 5.5, *Guardian* still achieves better trade-offs.

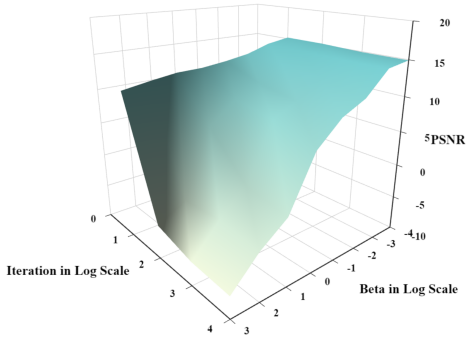

Figure 6: The defense performance of *Guardian* over different hyperparameters β and iterations.

Table 9: Time complexity comparison of different defenses.

Defense	Vanilla	Pruning	DP-Gaussian	DP-Laplace	Soteria	Our
Time (s)	5.15	6.18	6.13	6.14	4935.51	6.36

5.7 Sensitivity Analysis

Figure 6 shows that the effectiveness of *Guardian* over different β and iterations. As shown in Figure 6, lower β ($\leq 10^{-2}$) and iterations (= 1) only produce negligible defense performance against attacks. If $\alpha \geq 10$ together with iterations ≥ 100, the significant defense performance can be reaped but with the non-trivial sacrifice of model performance. In short, there is a trade-off between performance maintenance and privacy protection. The higher β and iterations are, the stronger the defense performance of *Guardian* owns, and the lower the model performance is.

5.8 Time Complexity Comparison

We compare the time complexity of different defenses under the setting used in Section 5.3. Table 9 presents the time consumption for 10 iterations of each defense. *Guardian* demonstrates competitive performance in terms of time efficiency. It is worth noting that Soteria requires a significant amount of time due to the need to run backpropagation algorithms multiple times, which is directly related to the number of neurons in the penultimate layer.

6 CONCLUSION

We developed *Guardian* which jointly optimizes PMM and PPM to produce transformed gradients. We showed the convergence guarantee of *Guardian* and introduced a better initialization strategy to decrease the overhead. We conducted extensive experiments to illustrate the superior performance of *Guardian*.

7 ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under grant number 62202170, Alibaba Group through the Alibaba Innovation Research Program, and the Open Research Fund of KLATASDS-MOE, ECNU.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Mislav Balunović, Dimitar I Dimitrov, Robin Staab, and Martin Vechev. 2022. Bayesian Framework for Gradient Leakage. In *International Conference on Learning Representations*.

[3] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[4] Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. TAG: Gradient Attack on Transformer-based Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 3600–3610.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[6] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Robert Wernsing. 2016. CryptoNets: applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*.

[7] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. 2020. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1000–1008.

[8] Mingyuan Fan, Cen Chen, Chengyu Wang, and Jun Huang. 2023. On the Trustworthiness Landscape of State-of-the-art Generative Models: A Comprehensive Survey. *ArXiv* abs/2307.16680 (2023). https://api.semanticscholar.org/CorpusID:260333997

[9] Mingyuan Fan, Cen Chen, Chengyu Wang, Wenmeng Zhou, and Jun Huang. 2022. Refiner: Data Refining against Gradient Leakage Attacks in Federated Learning. *ArXiv* abs/2212.02042 (2022). https://api.semanticscholar.org/CorpusID:254246302

[10] Wei Gao, Shangwei Guo, Tianwei Zhang, Han Qiu, Yonggang Wen, and Yang Liu. 2021. Privacy-preserving collaborative learning with automatic transformation search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 114–123.

[11] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

[12] Jinwoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. 2021. Gradient inversion with generative image prior. *Advances in Neural Information Processing Systems* 34 (2021).

[13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[14] Peter Kairouz, H. B. McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary B. Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim Y. El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Oluwasanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, R. Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Xiaodong Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2019. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* 14 (2019), 1–210.

[15] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. 2020. A review of applications in federated learning. *Computers & Industrial Engineering* 149 (2020), 106854.

[16] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[17] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics*.

[18] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. 2021. Quantifying information leakage from gradients. *arXiv e-prints* (2021), arXiv–2105.

[19] Daniel Scheliga, Patrick Mäder, and Marco Seeland. 2022. PRECODE-A Generic Model Extension to Prevent Deep Gradient Leakage. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1849–1858.

[20] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9311–9319.

[21] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. 2020. A framework for evaluating client privacy leakages in federated learning. In *European Symposium on Research in Computer Security*. Springer, 545–566.

[22] Dongxian Wu and Yisen Wang. 2021. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems* 34 (2021).

[23] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. 2020. Adversarial t-shirt! evading person detectors in a physical world. In *European conference on computer vision*. Springer, 665–681.

[24] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16337–16346.

[25] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In *USENIX Annual Technical Conference*.

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

[27] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems* 32 (2019).