Bayesian Framework for Gradient Leakage

Mislav Balunović, Dimitar I. Dimitrov, Robin Staab, Martin Vechev
Department of Computer Science
ETH Zurich
{mislav.balunovic, dimitadi, robin.staab, martin.vechev}@inf.ethz.ch

Abstract

Federated learning is an established method for training machine learning models without sharing training data. However, recent work has shown that it cannot guarantee data privacy as shared gradients can still leak sensitive information. To formalize the problem of *gradient leakage*, we propose a theoretical framework that enables, for the first time, analysis of the Bayes optimal adversary phrased as an optimization problem. We demonstrate that existing leakage attacks can be seen as approximations of this optimal adversary with different assumptions on the probability distributions of the input data and gradients. Our experiments confirm the effectiveness of the Bayes optimal adversary when it has knowledge of the underlying distribution. Further, our experimental evaluation shows that several existing heuristic defenses are not effective against stronger attacks, especially early in the training process. Thus, our findings indicate that the construction of more effective defenses and their evaluation remains an open problem.

1 Introduction

Federated learning [12] has become a standard paradigm for enabling users to collaboratively train machine learning models. In this setting, clients compute updates on their own devices, send the updates to a central server which aggregates them and updates the global model. Because user data is not shared with the server or other users, this framework should, in principle, offer more privacy than simply uploading the data to a server. However, this privacy benefit has been increasingly questioned by recent works [4, 13, 21, 23] which demonstrate the possibility of reconstructing the original input from shared gradient updates. The reconstruction works by optimizing a candidate image with respect to a loss that measures the distance between the shared and candidate gradients. The attacks typically differ in their loss function, their regularization, and how they solve the optimization problem. Importantly, the success of these attacks raises the following key questions: (i) *what is the theoretically worst-case attack*?, and (ii) *how do we evaluate defenses against gradient leakage*?

This work In this work, we study and address these two questions from a statistical perspective. Specifically, we introduce a theoretical framework which allows us to measure the expected risk an adversary has in reconstructing an input, given the joint probability distribution of inputs and their gradients. We then analyze the Bayes optimal adversary, which minimizes this risk and show that it solves a specific optimization problem involving the joint distribution. Further, we phrase existing attacks [4, 21, 23] as approximations of this optimal adversary, where each attack can be interpreted as implicitly making different assumptions on the distribution of gradients and inputs, in turn yielding different loss functions for the optimization. In our experimental evaluation, we compare the Bayes optimal adversary with other attacks, those which do not leverage the probability distribution of gradients, and we find that the Bayes optimal adversary performs better, as explained by the theory. We also experiment with several recently proposed defenses [3, 16, 17] based on different heuristics and demonstrate that they are not sufficient for protecting from gradient leakage against stronger

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.

attacks we design specifically for each defense. Interestingly, we find that models are especially vulnerable to attacks early in training, thus we advocate that defense evaluation should take place during and not only at the end of training. Overall, our findings suggest that creation of effective defenses and their evaluation is a challenging problem, and that our insights and contributions can substantially advance future research in the area.

Main contributions Our main contributions are:

- Formulation of the gradient leakage problem in a Bayesian framework which enables phrasing Bayes optimal adversary as an optimization problem.
- Interpretation of several prior attacks as approximations of the Bayes optimal adversary, each using different assumptions for the distributions of inputs and their gradients.
- Evaluation of several existing heuristic defenses demonstrating that they do not effectively protect from strong attacks, especially early in training.

2 Related work

We now briefly survey some of the work most related to ours.

Federated Learning Federated learning [12] was introduced as a way to train machine learning models in decentralized settings with data coming from different user devices. This new form of learning has caused much interest in its theoretical properties [8] and ways to improve training efficiency [9]. More specifically, besides decentralizing the computation on many devices, the fundamental promise of this approach is privacy, as the user data never leaves their devices.

Gradient Leakage Attacks Recent work [4, 23] has shown that such privacy assumptions, in fact, do not hold in practice, as an adversarial server can reliably recover an input image given gradient updates. They phrase the attacks as a minimization problem over the ℓ_2-distance between the gradients of a randomly initialized input image x' and a target image x on a neural network h_θ:

$$(x^*, y^*) = \arg\min_{(x', y')} ||\nabla l(h_\theta(x), y) - \nabla l(h_\theta(x'), y')||_2 \tag{1}$$

Here l denotes a loss used to train the network, usually cross-entropy, while y and y' correspond to the original and reconstructed label, respectively. Follow-up works improve on these results by using different distance metrics such as cosine similarity and input-regularization [4], smarter initialization [19], normalization [21], and others [5, 14]. A significant improvement proposed by [22] showed how to recover the target label y from the gradients alone, reducing Eq. (1) to an optimization over x' only. In Section 4 we show how these existing attacks can be interpreted as different approximations of the Bayes optimal adversary.

Defenses In response to the rise of privacy-violating attacks on federated learning, many defenses have been proposed [1, 3, 17]. Except for DP-SGD [1], a version of SGD with clipping and adding Gaussian noise, which is differentially private, they all provide none or little theoretical privacy guarantees. This is partly due to the fact that no mathematically rigorous attacker model exists, and defenses are empirically evaluated against known attacks. This also leads to a wide variety of proposed defenses: Soteria [17] prunes the gradient for a single layer, ATS [3] generates highly augmented input images that train the network to produce non-invertible gradients, and PRECODE [16] uses a VAE to hide the original input. Here we do not consider defenses that change the communication and training protocol [11, 20].

3 Background

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be an input space, and let $h_\theta : \mathcal{X} \to \mathcal{Y}$ be a neural network with parameters θ classifying an input to a label in the label space $\mathcal{Y}$. We assume that inputs (x, y) are coming from a distribution $\mathcal{D}$ with a marginal distribution $p(x)$. In standard federated learning, there are n clients with loss functions $l_1, ..., l_n$, who are trying to collaboratively solve the following optimization problem:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[l_i(h_\theta(x), y) \right].$$

To ease the notation, we will assume a single client throughout the paper, but the same reasoning can be applied to the general n-client case. Additionally, each client could have a different distribution $\mathcal{D}$, but the approach is again easy to generalize to this case. In a single training step, each client i first computes $\nabla_\theta l_i(h_\theta(x_i), y_i)$ on a batch of data (x_i, y_i), then sends these to the central server that performs a gradient descent step to obtain the new parameters $\theta' = \theta - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla_\theta l_i(h_\theta(x_i), y_i)$, where α is a learning rate. We will consider a scenario where each client reports, instead of the true gradient $\nabla_\theta l_i(h_\theta(x_i), y_i)$, a noisy gradient g sampled from a distribution $p(g|x)$, which we call a defense mechanism. The purpose of the defense mechanism is to add enough noise to hide the sensitive user information from the gradients while retaining high enough informativeness so that it can be used for training. Thus, given some noisy gradient g, the central server would update the parameters as $\theta' = \theta - \alpha g$ (assuming n = 1 as mentioned above). Typical examples of defenses used in our experiments in Section 6, each inducing $p(g|x)$, include adding Gaussian or Laplacian noise to the original gradients, as well as randomly masking some components of the gradient. Naturally, $p(x)$ and $p(g|x)$ together induce a joint distribution $p(x, g)$. Note that a network that has no defense corresponds to the distribution $p(g|x)$, which is concentrated only on the true gradient at x.

4 Bayesian adversarial framework

Here we describe our theoretical framework for gradient leakage in federated learning.

Adversarial risk We first define the adversarial risk for gradient leakage and then derive the Bayes optimal adversary that minimizes this risk. The adversary can observe the gradient g and is trying to reconstruct the input x that produced this gradient. Formally, the adversary is a function $f : \mathbb{R}^k \to \mathcal{X}$ mapping gradients to inputs. Given some (x, g) sampled from the joint distribution $p(x, g)$, the adversary outputs the reconstruction $f(g)$ and incurs loss $\mathcal{L}(x, f(g))$, which is a function $\mathcal{L} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Typically, we will consider a binary loss that evaluates to 0 if the adversary's output is close to the original input, and 1 otherwise. If the adversary wants to reconstruct the *exact* input x, we can define the loss to be $\mathcal{L}(x, x') := 1_{x \neq x'}$, denoting with 1 the indicator function. If the adversary only wants to get to some δ-neighbourhood of the input x in the input space, a more appropriate definition of the loss is $\mathcal{L}(x, x') := 1_{d(x,x')>\delta}$. In this section, we will assume that the distance d is the ℓ_2-distance, but the approach can be generalized to other notions as well. This definition is well suited for image data, where ℓ_2 distance captures our perception of visual closeness, and for which the adversary can often obtain a reconstruction that is very close to the original image, both visually and in the ℓ_2 space. Note that if we let δ approach 0 in our second definition of the loss, we essentially recover the first loss. We can now define the risk $R(f)$ of the adversary f as

$$R(f) := \mathbb{E}_{x,g} \left[\mathcal{L}(x, f(g)) \right] = \mathbb{E}_{x\sim p(x)} \mathbb{E}_{g\sim p(g|x)} \left[\mathcal{L}(x, f(g)) \right].$$

Bayes optimal adversary We consider white-box adversary who knows the joint distribution $p(x, g)$, as opposed to the weaker alternative based on security through obscurity, where adversary does not know what defense is used, and therefore does not have a good estimate of $p(x, g)$. We also do not consider adversaries that can exploit other vulnerabilities in the system to obtain extra information. Let us consider the second definition of the loss for which $\mathcal{L}(x, f(g)) := 1_{||x-f(g)||_2>\delta}$. We can then rewrite the definition of the risk as follows:

$$\begin{aligned} R(f) &= \mathbb{E}_{x,g} \left[\mathcal{L}(x, f(g)) \right] \\ &= \mathbb{E}_g \mathbb{E}_{x|g} \left[1_{||x-f(g)||_2>\delta} \right] \\ &= \mathbb{E}_g \int_{\mathcal{X}} p(x|g) \cdot 1_{||x-f(g)||_2>\delta} \, dx \\ &= \mathbb{E}_g \int_{\mathcal{X} \setminus B(f(g),\delta)} p(x|g) \, dx \\ &= 1 - \mathbb{E}_g \int_{B(f(g),\delta)} p(x|g) \, dx. \end{aligned}$$

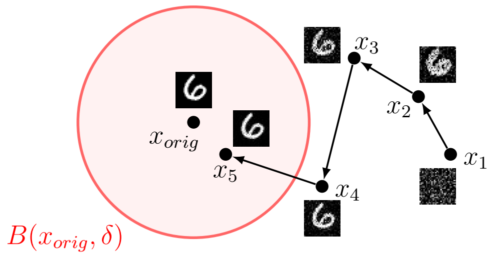


Figure 1: Example of a gradient leakage attack. Bayes optimal adversary randomly initializes image x_1 and then optimizes for the image with the highest $\log p(g|x) + \log p(x)$ in its δ-neighborhood. The adversary has loss 1 if the final reconstruction is outside the ball $B(x_{orig}, \delta)$, and 0 otherwise.

Here $B(f(g), \delta)$ denotes the ℓ_2-ball of radius δ around $f(g)$. Thus, an adversary which wants to minimize their risk has to maximize $\mathbb{E}_g \int_{B(f(g),\delta)} p(x|g)\, dx$, meaning that the adversarial function f can be defined as $f(g) := \arg\max_{x_0} \int_{B(x_0,\delta)} p(x|g)\, dx$. Intuitively, the adversary predicts x_0 which has the highest likelihood that one of the inputs in its δ-neighborhood was the original input that produced gradient g. Note that if we would let $\delta \to 0$, then $f(g) \to \arg\max_x p(x|g)$, which would be the solution for the loss that requires the recovered input to exactly match the original input. As we do not have the closed form for $p(x|g)$, it can be rewritten using Bayes' rule:

$$
\begin{aligned}
f(g) &= \arg\max_{x_0 \in \mathcal{X}} \int_{B(x_0,\delta)} p(x|g)\, dx \\
&= \arg\max_{x_0 \in \mathcal{X}} \int_{B(x_0,\delta)} \frac{p(g|x)p(x)}{p(g)}\, dx \\
&= \arg\max_{x_0 \in \mathcal{X}} \int_{B(x_0,\delta)} p(g|x)p(x)\, dx \\
&= \arg\max_{x_0 \in \mathcal{X}} \left[\log \int_{B(x_0,\delta)} p(g|x)p(x)\, dx \right]
\end{aligned}
\tag{2}
$$

The last steps follow from multiplying by the constant $p(g)$ and taking the logarithm, both not affecting the argmax. Computing the optimal reconstruction now requires evaluating both the input prior $p(x)$ and the conditional probability $p(g|x)$, which is determined by the used defense mechanism. Given these two ingredients, Eq. (2) then provides us with a way to compute the output of the Bayes optimal adversary by solving an optimization problem involving distributions $p(x)$ and $p(g|x)$.

Approximate Bayes optimal adversary While Eq. (2) provides a formula for the optimal adversary in the form of an optimization problem, using this adversary for practical reconstruction is difficult due to three main challenges: (i) we require knowledge of the prior distribution $p(x)$, (ii) computing the integral over the δ-ball around x_0 is generally not tractable, and (iii) we need to solve the optimization problem over $\mathcal{X}$. However, we can address each of these challenges by introducing appropriate approximations. We first apply Jensen's inequality to the logarithm function:

Algorithm 1 Approximate Bayes optimal adversary

```
x ← attack_init()
for i = 1 to m do
    Sample x_1, ..., x_k uniformly from B(x, δ)
    x ← x + α∇_x (1/k) Σ_{i=1}^{k} log p(g|x_i) + log p(x_i)
end for
return x
```

$$
\max_{x_0 \in \mathcal{X}} \left[\log \int_{B(x_0,\delta)} p(g|x)p(x)\, dx \right] \geq \max_{x_0 \in \mathcal{X}} \left[\int_{B(x_0,\delta)} \log p(g|x) + \log p(x) \right].
$$

For image data, we approximate the prior distribution using the total variation image prior, which has already worked well for gradient inversion [4]. Alternatively, we could estimate it from data using

Attack	Prior $p(x)$	Conditional $p(g|x)$
DLG [23]	Uniform	Gaussian
Inverting Gradients [4]	TV	Cosine
GradInversion [21]	TV + Gaussian + DeepInv	Gaussian

Table 1: Several existing attacks can be interpreted as instances of our Bayesian framework. We show prior and conditional distribution for corresponding losses that each attack uses.

density estimation models such as PixelCNN [18] or Glow [7]. We then approximate the integral over the δ-ball via Monte Carlo integration by sampling k points $x_1, ..., x_k$ uniformly inside the ball. Finally, as the objective is differentiable, we can use gradient-based optimizer such as Adam [6], and obtain the attack in Algorithm 1. Fig. 1 shows a single run of this adversary which is initialized randomly, and gets closer to the original image at every step.

Existing attacks as approximations of the Bayes optimal adversary We now describe how existing attacks can be viewed as different approximations of the Bayes optimal adversary. Recall that the optimal adversary f searches for the δ-ball with the maximum value of the integral of $\log p(g|x) + \log p(x)$. Previously proposed attacks can in fact be recovered by plugging in different approximations for $\log p(g|x)$ and $\log p(x)$ into Algorithm 1, with estimating the integral using $k = 1$ sample located at the center of the ball. For example, suppose that the defense mechanism adds Gaussian noise to the original gradient, which corresponds to the conditional probability $p(g|x) = \mathcal{N}(\nabla_\theta l(h_\theta(x), y), \sigma^2 I)$. This implies that $\log p(g|x) = C - \frac{1}{2\sigma^2}||g - \nabla_\theta l(h_\theta(x), y)||_2^2$, where C is a constant. Assuming a uniform prior (where $\log p(x)$ is constant), the optimization problem in Eq. (2) boils down to minimizing ℓ_2 distance between g and $\nabla_\theta l(h_\theta(x), y)$, which is exactly the optimization problem solved by Deep Leakage from Gradients (DLG) [23]. Inverting Gradients [4] uses a total variation (TV) image prior for $\log p(x)$ and cosine similarity instead of ℓ_2 to measure similarity between gradients. Cosine similarity corresponds to the distribution $\log p(g|x) = C - \frac{g^T \nabla_\theta l(h_\theta(x),y)}{||g||_2 ||\nabla_\theta l(h_\theta(x),y)||_2}$ which also requires the support of the distribution to be bounded (this happens, e.g., when gradients are clipped). GradInversion [21] introduces a more complex prior based on a combination of the total variation, ℓ_2 norm of the image, and a DeepInversion prior while using l_2 norm to measure distance between gradients, which corresponds to Gaussian noise, as observed before. We summarize these observations in Table 1.

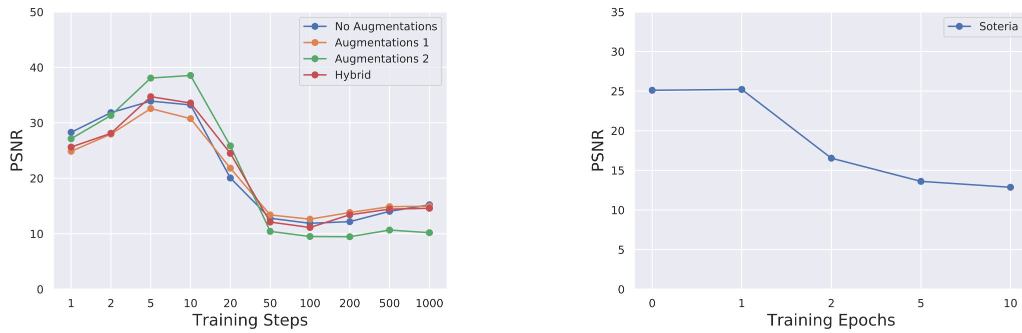
5 Attacking existing defenses

In this section we provide practical attacks against three recent defenses, showing that they are generally not able to withstand stronger adversaries early in training. While in Section 4 we have shown that Bayes optimal adversary is the optimal attack for any defense (each inducing different $p(g|x)$), computing it is not tractable for the three defenses we consider. Here, it is more efficient to create a custom attack tailored to each defense, which is enough to break their privacy promise.

Soteria The Soteria defense [17] perturbs the intermediate representation of the input at a chosen defended layer l of the attacked neural network $H : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ with L layers of size $n_1, \ldots, n_L$ and input size n_0. Let X and $X' \in \mathbb{R}^{n_0}$ denote the original and reconstructed images on H and $h_{i,j} : \mathbb{R}^{n_i} \to \mathbb{R}^{n_j}$ denote the function between the input of the i^{th} layer of H and the output of the j^{th}. For the chosen layer l, Sun et al. [17] denotes the inputs to that layer for the images X and X' with $r = h_{0,l-1}(X)$ and $r' = h_{0,l-1}(X')$, respectively, and aims to solve

$$\max_{r'} ||X - X'||_2 \text{ s.t. } ||r - r'||_0 \leq \epsilon. \tag{3}$$

Intuitively, Eq. (3) is searching for a minimal perturbation of the input to layer l that results in maximal perturbation of the respective reconstructed input X'. Despite the optimization being over the intermediate representation r', for an attacker who observes neither r nor r' the defense amounts to a perturbed gradient at layer l. In particular let $\nabla W = \{\nabla W_1, \nabla W_2, \ldots \nabla W_L\}$ be the set of gradients for the variables in the different layers of H. Sun et al. [17] first solves Eq. (3) to obtain r'. It afterwards uses this r' to generate a perturbed gradient at layer l denoted with $\nabla W'_l$. Notably r'



(a) ATS

(b) Soteria

Figure 2: PSNR obtained by reconstruction attacks on ATS and Soteria during the first 1000 steps and 10 epochs, respectively, demonstrating high gradient leakage early in training.

is not propagated further through the network and hence the gradient perturbation stays local to the defended layer l. Hence, to defend the data X, Soteria's clients send the perturbed set of gradients $\nabla W' = \{\nabla W_1, \ldots, \nabla W_{l-1}, \nabla W_l', \nabla W_{l+1}, \ldots \nabla W_L\}$ in place of ∇W. Sun et al. [17] show that Soteria is safe against several attacks from prior work [4, 23].

In this work, we propose to circumvent this limitation by dropping the perturbed gradients $\nabla W_l'$ from $\nabla W'$ to obtain $\nabla W^* = \{\nabla W_1, \ldots, \nabla W_{l-1}, \nabla W_{l+1}, \ldots \nabla W_L\}$. As long as ∇W^* contains enough gradients, this allows an attacker to compute an almost perfect reconstruction of X. Note that the attacker does not know which layer is defended, but can simply run the attack for all.

Automated Transformation Search The Automatic Transformation Search (ATS) [3] attempts to hide sensitive information from input images by augmenting the images during training. The key idea is to score sequences of roughly 3 to 6 augmentations from AutoAugment library [2] based on the ability of the trained network to withstand gradient-based reconstruction attacks and the overall network accuracy. Similarly to Soteria, Gao et al. [3] also demonstrate that ATS is safe against attacks proposed by Zhu et al. [23] and Geiping et al. [4]. In this work we show that, even though the ATS defense works well in later stages of training, in initial communication rounds we can easily reconstruct the input images using Geiping et al. [4]'s attack.

PRECODE PRECODE [16] is a proposed defense which inserts a variational bottleneck between two layers in the network. Given an input x, PRECODE first encodes the input into a representation $z = E(x)$, then samples bottleneck features $b \sim q(b|z)$ from a latent Gaussian distribution. Then, they compute new latent representation $\hat{z} = D(b)$, and finally obtain the output $\hat{y} = O(\hat{z})$. For this defense we focus on an MLP network, which [16] evaluates against several attacks and shows that images cannot be recovered. Our attack is based on the approach from [15], which shows that in most cases the inputs to an MLP can be perfectly reconstructed from the gradients.

6 Experimental evaluation

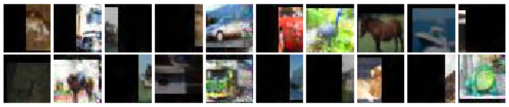
We now evaluate existing defenses and practical implementation of the Bayes optimal adversary.

Evaluating existing defenses In this experiment, we evaluate the three recently proposed defenses described in Section 5: Soteria [17], ATS [3], and PRECODE [16] on the CIFAR-10 dataset [10]. For ATS and Soteria, we use the code from their respective papers. In particular, we evaluated ATS on their ConvNet implementation, a network with 7 convolutional layers, batch-norm, and ReLU activations followed by a single linear layer. We consider 2 different augmentation strategies for ATS, as well as a hybrid strategy. For Soteria, we evaluate our own network architecture with 2 convolutional and 3 linear layers. The defense is applied on the largest linear layer. We provide additional details on the hyperparameters used for this experiment in Appendix A.2.

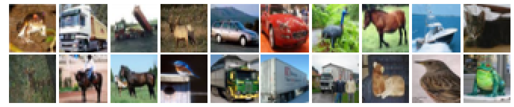
The results of this experiment are shown in Fig. 2. We attack ATS for the first 1000 steps of training and Soteria for the first 10 epochs, measuring Peak signal-to-noise ratio (PSNR) of the reconstruction

(a) Original

(b) Soteria



(c) ATS



(d) PRECODE

Figure 3: Images obtained by running attacks on Soteria, ATS, and PRECODE on the CIFAR-10 dataset after 10-th training step, showing that defenses do not protect privacy early in the training.

obtained using our attack at every step. In all of our PRECODE experiments we obtain perfect reconstruction with PSNR values > 150 so we do not show PRECODE on the plot. We can observe that, generally, each network becomes less susceptible to the attack with the increased number of training steps. However, early in training, networks are very vulnerable, and images can be reconstructed almost perfectly. Fig. 3 visualizes the first 40 reconstructed images obtained using our attacks on Soteria, ATS and PRECODE. We can see that for all defenses, our reconstructions are very close to their respective inputs. This indicates that proposed defenses do not reliably protect privacy under gradient leakage, especially in earlier stages of training. Our findings suggest that creating effective defenses and properly evaluating them remains a key challenge.

Approximations of Bayes optimal adversary In this experiment, we compare the Bayes optimal adversary with attacks that have suboptimal approximations of $p(g|x)$ and $p(x)$. As vision datasets have complex priors $p(x)$ which we cannot compute exactly, for this experiment, we consider a synthetic dataset where $p(x)$ is 20-dimensional unit Gaussian. We define the true label $y := \arg\max(Wx)$ where W is a fixed random matrix, and perform an attack on a 2-layer MLP defended by adding Laplacian noise with a 0.1 scale. Thus, $p(x)$ is a Gaussian, and $p(g|x)$ is Laplacian. In this study, we consider 4 different variants of the attack, obtained by choosing Laplacian or Gaussian for prior and conditional. In Fig. 4 for each attack, we show the distance from the original input for 200 steps. We can observe that Bayes optimal attack (with Gaussian prior and Laplacian conditional) converges significantly closer to the original input than the other attacks, as predicted by the theory in Section 4.

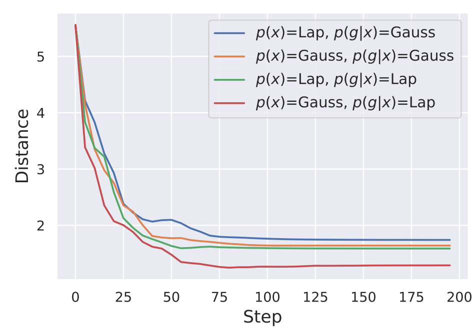


Figure 4: Ablation with the Bayes optimal attack.

7 Conclusion

We proposed a theoretical framework to formally analyze the problem of gradient leakage, which has recently emerged as an important privacy issue for federated learning. Our framework enables us to analyze the Bayes optimal adversary for this setting and phrase it as an optimization problem. We interpreted several previously proposed attacks as approximations of the Bayes optimal adversary, each approximation implicitly using different assumptions on the distribution over inputs and gradients. Our experimental evaluation shows that the Bayes optimal adversary is effective in scenarios in which it knows the underlying distribution. We also experimented with several proposed defenses based on heuristics and found that they do not offer effective protection against stronger attacks. Given our findings, we believe that formulating an effective defense that balances accuracy and protection against gradient leakage during all stages of training remains an exciting open challenge.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (ACM CCS), pp. 308-318, 2016*, 2016. doi: 10.1145/2976749.2978318.

[2] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pp. 113–123. Computer Vision Foundation / IEEE, 2019.

[3] Wei Gao, Shangwei Guo, Tianwei Zhang, Han Qiu, Yonggang Wen, and Yang Liu. Privacy-preserving collaborative learning with automatic transformation search. In *CVPR*, pp. 114–123. Computer Vision Foundation / IEEE, 2021.

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

[5] Jinwoo Jeon, Jaechang Kim, Kangwook Lee, Sewoong Oh, and Jungseul Ok. Gradient inversion with generative image prior. In *International Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2021.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[7] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

[8] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527, 2016.

[9] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492, 2016.

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[11] Hongkyu Lee, Jeehyeong Kim, Seyoung Ahn, Rasheed Hussain, Sunghyun Cho, and Junggab Son. Digestive neural networks: A novel defense strategy against inference attacks in federated learning. *Computers and Security*, 109:102378, 2021.

[12] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

[13] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, pp. 691–706. IEEE, 2019.

[14] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. Quantifying information leakage from gradients. *CoRR*, abs/2105.13929, 2021.

[15] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning: Revisited and enhanced. In *ATIS*, volume 719 of *Communications in Computer and Information Science*, pp. 100–110. Springer, 2017.

[16] Daniel Scheliga, Patrick Mäder, and Marco Seeland. Precode - a generic model extension to prevent deep gradient leakage, 2021.

[17] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *CVPR*, pp. 9311–9319. Computer Vision Foundation / IEEE, 2021.

[18] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

[19] Wenqi Wei, Ling Liu, Margaret Loper, Ka Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *ESORICS (1)*, volume 12308 of *Lecture Notes in Computer Science*, pp. 545–566. Springer, 2020.

[20] Wenqi Wei, Ling Liu, Yanzhao Wu, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. *CoRR*, abs/2107.01154, 2021.

[21] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *CVPR*. Computer Vision Foundation / IEEE, 2021.

[22] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients, 2020.

[23] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *NeurIPS*, 2019.