
Data Reconstruction Attacks and Defenses: A Systematic Evaluation

Sheng Liu*

Stanford University
shengl@stanford.edu

Zihan Wang*

New York University
zw3508@nyu.edu

Yuxiao Chen

Peking University
2100010776@stu.pku.edu.cn

Qi Lei

New York University
ql518@nyu.edu

Abstract

Reconstruction attacks and defenses are essential in understanding the data leakage problem in machine learning. However, prior work has centered around empirical observations of gradient inversion attacks, lacks theoretical grounding, and cannot disentangle the usefulness of defending methods from the computational limitation of attacking methods. In this work, we propose to view the problem as an inverse problem, enabling us to theoretically and systematically evaluate the data reconstruction attack. On various defense methods, we derived the algorithmic upper bound and the matching (in feature dimension and architecture dimension) information-theoretical lower bound on the reconstruction error for two-layer neural networks. To complement the theoretical results and investigate the utility-privacy trade-off, we defined a natural evaluation metric of the defense methods with similar utility loss among the strongest attacks. We further propose a strong reconstruction attack that helps update some previous understanding of the strength of defense methods under our proposed evaluation metric.

1 Introduction

Machine learning research has transformed the technical landscape across various domains but also raises privacy concerns potentially (Li et al., 2023; Papernot et al., 2016). Federated learning (Konečný et al., 2016; McMahan et al., 2017), a collaborative multi-site training framework, aims to uphold user privacy by keeping

user data local while only exchanging model parameters and updates between a central server and edge users. However, recent studies on reconstruction attacks (Yin et al., 2021; Huang et al., 2021) highlight privacy vulnerabilities even within federated learning. Attackers can eavesdrop on shared trained models and gradient information and reconstruct training data using them. Even worse, honest but curious servers can inadvertently expose training data by querying designed model parameters. Some defending methods are also proposed and analyzed.

However, previous research on reconstruction attacks and defenses centered around empirical studies and lacked theoretical guarantees. Moreover, those experiments over defenses usually focus only on one specific attack, making the evaluation of defenses’ strength untrustworthy. Empirical understanding of attacking and defending methods is affected by factors like optimization challenges of non-convex objectives. Under the heuristic attacking methods from prior work, it is hard to disentangle the following two possibilities: 1) the algorithmic or computational barrier of the specific attack and 2) the defense’s success. Thus, a systematic and information-theoretical evaluation of defenses that is independent to the attack method is needed.

A line of theoretical work analyzing privacy attacks is differential privacy (DP) (Dwork, 2006), which guarantees privacy in data re-identification or membership inference. However, there are scenarios when data identity information is not sensitive, but data itself is, and DP is unsuitable for interpreting data reconstruction attacks (Guo et al., 2022; Hayes et al., 2024). An algorithm can yield no DP guarantee but prevent the threat model from data reconstruction (Details deferred to Section 2). Although some previous works discussed the reconstruction attack with more tailored Renyi-DP (Guo et al., 2022; Stock et al., 2022), their setting is when all samples are known, and the attacker’s goal is to reconstruct the last data point, which doesn’t apply to the federated learning setting. We will discuss how (Renyi-)DP is too stringent for the data leakage problem under federated learning in Section 2. Some

*Equal contribution. Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

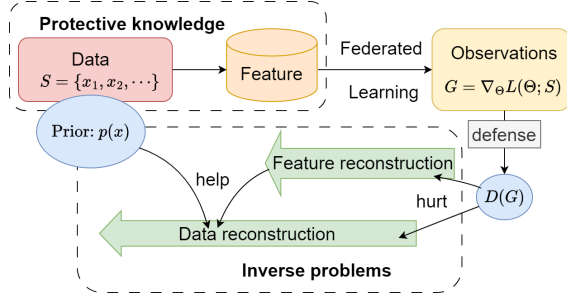


Figure 1: An illustration of the key components of data reconstruction studied in this paper.

underlying structures of the observation can yield no (Renyi-)DP guarantee but prevent the attacker from reconstructing the data.

In our work, we propose a framework for evaluating defenses that view the reconstruction attack as an inverse problem (Fig. 1): from the gradient G that is generated from the unknown data S and a known function (objective’s gradient), the adversary intends to reconstruct the input data, and a defender D is to prevent this from perturbing the observations. Our theoretical analysis focuses on 2-layer fully connected networks and gives both upper and lower bounds on the error of this inverse problem under various defenses. We are especially interested in obtaining a quantitative relationship between reconstruction error and the key factors of the learning pipeline, such as data dimension, model architecture (width), and defense strength, which are derived in the form of both algorithmic upper bounds and (matching) information-theoretic lower bound (that is independent of the attack algorithm). To generalize the analysis to real-world models and study the utility-privacy trade-offs, we empirically conduct a comprehensive evaluation of different defenses’ performances. Our comparisons are based on the strongest attack for each defense to exclude the case that some defenses are only effective for a specific attack.

Our paper is organized as follows. In Section 2, we introduce the exact setups of both theoretical and empirical analysis of defenses. We also explain why DP is not suitable for our setting. In Section 3, we summarize the defenses used in reconstruction attacks and analyze the error bounds theoretically for 2-layer networks. In Section 4.1, we introduce several previous attacking methods used in our empirical analysis and propose a strong attack based on an intermediate feature reconstruction method (Wang et al., 2023). In Section 5, we show the results of experiments and evaluate the defenses based on utility loss and best reconstruction error.

1.1 Related Work

In federated learning (Konečný et al., 2016; McMahan et al., 2017), a line of work in gradient inversion aims to reconstruct training data from gradients. Early work by Aono et al., 2017 theoretically showed that reconstruction is possible under a single neuron setting. Zhu et al., 2019 next proposed the gradient matching framework, where the training data and labels are recovered by minimizing the l_2 distance between gradients generated by dummy variables and real data. Instead of using l_2 distance, Wang et al., 2020 proposed a Gaussian kernel-based distance function, and Geiping et al., 2020 used cosine similarity.

Besides using different distance measures, recent works introduced image priors to gradient matching. Geiping et al., 2020 added a total variation regularization term since pixels of a real image are usually similar to their neighbor. Wei et al., 2020b introduced a label-based regularization to fit the prediction results. Balunovic et al., 2021 formulated the gradient inversion attack in a Bayesian framework. Yin et al., 2021 used a batch-normalization regularization to fit the BN statistics and a group consistency regularization to indicate the spatial equivariance of CNN. Instead of using regularization terms, Jeon et al., 2021; Xue et al., 2023; Li et al., 2022 represented dummy training data by a generative model to preserve the image prior.

Some recent works investigated new data reconstruction methods outside the framework of gradient matching. Wang et al., 2023 and Kariyappa et al., 2023 respectively use tensor-based method and independent component analysis to construct data from gradient information, where optimization is not needed. Haim et al., 2022 proposed a reconstruction method using the stationary property for trained neural networks induced by the implicit bias of training. Other works investigate partial data reconstructions with fishing parameters (Wen et al., 2022; Boenisch et al., 2023; Fowl et al., 2021).

Defense strategies in this setting, however, are relatively less studied. The original setting of federated learning proposed by McMahan et al., 2017 used local aggregation to update multiple steps instead of gradients directly. Bonawitz et al., 2016, 2017 proposed a secure aggregation that can collect the gradients from multiple clients before updating. Geyer et al., 2017; Wei et al., 2020a introduced differential privacy into federated learning, perturbing the gradients. Aono et al., 2017 demonstrated an encryption framework in the setting of federated learning. Moreover, many tricks designed to improve the performance of neural networks are effective in defending against privacy attacks, such as Dropout (Hinton et al., 2012), gradient

pruning (Sun et al., 2017), and MixUp (Zhang et al., 2017). The effect of defending strategies against gradient matching attacks are discussed in (Zhu et al., 2019; Geiping et al., 2020; Huang et al., 2021).

2 Preliminary

In reconstruction attacks, we observe the model iterations in training and recover data from the gradient $\nabla L(\Theta; S)$, where L is the loss function, Θ is the model and S is an unknown dataset. Then, it can be regarded as an inverse problem to identify unknown signal S from observation $G = \nabla L(\Theta; S)$, where ∇L is the forward function. We denote the data space by \mathcal{X} and the batch size of data by B . Then $S \in \mathcal{X}^B$. Under the inverse problem framework, the effectiveness of attacks and defenses becomes the feasibility and hardness of recovering S . Formally, we define a minimax risk of reconstruction error:

$$R_L = \left(\min_{\hat{S}=\hat{S}(G)} \max_{S \in \mathcal{X}^B} \min_{\pi} \mathbb{E} \left[d(S, \pi(\hat{S})) \right] \right)^{1/2}. \quad (1)$$

Here $d(S, \pi(\hat{S})) = \frac{1}{B} \sum_{i=1}^B \|S_i - \hat{S}_{\pi(i)}\|^2$, where π is a permutation of $[B]$. We will study in Section 3 the reconstruction error lower bounded in the setting of a two-layer network with random weights¹. For upper bounding reconstruction errors, we consider the error for specific algorithm A . We define the reconstruction error by

$$R_U = \max_{S \in \mathcal{X}^B} \min_{\pi} \left(d(S, \pi(\hat{S})) \right)^{1/2}. \quad (2)$$

Note that in both lower and upper bounds, we consider the minimum over permutation. It is because batched gradient descent adds gradient together in an unordered manner, so only identifying the set of data points without their correspondence is important.

Our studies are meant to complement the esteemed DP (Dwork, 2006) to analyze the reconstruction as DP is too strong and sometimes unnecessary for our studied settings. On one hand, scenarios exist when reconstruction is impossible, even when DP does not hold². In addition, the price of achieving DP in the data reconstruction setting is high. For DP-SGD (Abadi et al., 2016), the well-established Gaussian mechanism requires a large variance that will destroy the utility of gradient information:

¹Throughout the paper, we use a unified scaling for data and network weights similar to Mean-field view (Mei et al., 2018).

²For instance, one can easily verify that with linear net and l_2 loss, the observation becomes $\sum_{i=1}^B x_i$, where no DP is satisfied but individual recovery of x_i is impossible for $B > 1$ unless specific prior knowledge is assumed, for example natural images.

Proposition 2.1 (Short version of Proposition A.1). *For a two-layer neural network with m hidden nodes and random weights, we denote the gradient by G . Under mild assumptions, the randomized mechanism $\mathcal{M} = G + \mathcal{N}(0, \sigma^2 I)$ is (ϵ, δ) -DP for any $\epsilon, \delta > 0$ if $\sigma^2 = \Omega(\frac{m \log(1/\delta)}{\epsilon})$.*

In this work, we will theoretically and empirically evaluate defenses against reconstruction. For theoretical analysis, we bound the reconstruction error under various defenses D by estimating Eq. (1) and (2), where the observation is $D(G)$ instead of G . The theoretical bounds will connect the model architecture, defenses' strength, and data dimension. However, only theoretical analysis lacks an extension to general networks and utility-privacy trade-off. Therefore we introduce a systematic way to evaluate defenses by empirical analysis further. Within a set of defenses \mathcal{D}_U of the same level of utility loss \mathcal{U} , we will measure the following criterion for each $D \in \mathcal{D}_U$:

$$\mathcal{S}_D := \max_{A \in \mathcal{A}} d(S, A(D(G))), \quad (3)$$

which evaluates their strength against the most effective attacker in \mathcal{A} , the set of considered reconstruction attacks. In experiments, we conduct various attacks under different defenses to estimate Eq. (3). Then, we compare the best reconstruction error with the same utility loss for each defense.

3 Theoretical Analysis

In theoretical analysis, we mainly focus on two-layer neural networks $f(\mathbf{x}; \Theta) = \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x})$ with ℓ be the square loss. Here $a_j \sim \mathcal{N}(0, 1/m^2)$ and $\mathbf{w}_j \sim \mathcal{N}(0, I)$. For the upper bound, we examine a specific attack method proposed by (Wang et al., 2023). The attack method involves the computation of gradient-related tensor $\sum_{j=1}^m g(\mathbf{w}_j) H_p(\mathbf{w}_j)$, where $g(\mathbf{w}_j) := \frac{\partial \ell}{\partial a_j}$ and H_p is the p -th Hermite function. By Stein's lemma (Stein, 1981; Mamis, 2022), it is approximate to a tensor product $\sum_{i=1}^B c_i \mathbf{x}_i^{\otimes p}$ when \mathbf{w}_j 's are random Gaussian and we can conduct tensor decomposition to recover data, where B is the batch size.

Theorem 3.1 (Informal, Theorem 5.1 in (Wang et al., 2023)). *For a 2-layer network under mild assumptions, the reconstruction error of the tensor-based attack has an upper bound $R_U \leq \tilde{O}(B \sqrt{\frac{d}{m}})$ with high probability.*

For the lower bound, to avoid pure combinatorial analysis, we formulate data reconstruction as a statistical estimation problem: we observe the noisy gradient $\nabla_{\Theta} L(\Theta; S) + \epsilon$ and identify the input data S . Here $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random vector. We consider the minimax lower bound in Eq. (1), where the

	No defense	Local aggregation	Gradient noise	Gradient clipping	DP-SGD	Dropout	Gradient pruning
Upper bound	$B\sqrt{\frac{d}{m}}$	$B\sqrt{\frac{d}{m}}$	$(B + \sigma_0)\sqrt{\frac{d}{m}}$	$B\sqrt{\frac{d}{m}}$	$(B + \sigma_0 \max\{1, \frac{\ G\ }{C}\})\sqrt{\frac{d}{m}}$	$B\sqrt{\frac{d}{(1-p)m}}$	Unknown
Lower bound	$\sigma\sqrt{\frac{d}{m}}$	$\sigma\sqrt{\frac{d}{m}}$	$\sigma\sqrt{\frac{d}{m}}$	$\sigma \max\{1, \frac{\ G\ }{C}\}\sqrt{\frac{d}{m}}$	$\sigma \max\{1, \frac{\ G\ }{C}\}\sqrt{\frac{d}{m}}$	$\sigma\sqrt{\frac{d}{(1-p)m}}$	$\sigma\sqrt{\frac{d}{(1-p)m}}$

Table 1: The algorithmic upper bound and information-theoretic lower bound of the reconstruction error against different defenses. The bound here is the order with respect to different factors. Some parameters are defined later in the subsection of each defense.

expectation is over the random Gaussian noise and the high probability in the following proposition is derived with random weights.

Theorem 3.2 (Short version for Theorem D.7). *For a 2-layer network with noisy gradient, the statistical minimax risk has a lower bound $R_L \geq \tilde{\Omega}(\sigma\sqrt{\frac{d}{m}})$ with high probability.*

In the following, we analyze the upper and lower bound for the reconstruction under different defenses. The bound is shown in Table 1. Our analysis provides a tight and quantitative analysis of how each defense’s defending strength (such as pruning ratio or noise variance) affects the data reconstruction error (which is inversely proportional to the degree of privacy).

3.1 Local aggregation

In a realistic setting of federated learning, the observed local update can be multiple steps of gradient descent (McMahan et al., 2017). In our analysis, we mainly consider the most simple cases, where local devices train two steps and the observation is $\Theta^{(2)} - \Theta^{(0)}$. Since one step of training does not change the parameters too much, we can approximately assume that the two steps share a same set of parameters. Then the analysis is similar to that without defense.

Proposition 3.1 (Short version for Proposition C.1). *For a 2-layer network with mild assumptions, the reconstruction error of tensor based attack under local aggregation defense for 2 steps has an upper bound $R_U \leq \tilde{O}(B\sqrt{\frac{d}{m}})$ with high probability.*

Proposition 3.2 (Short version for Proposition E.2). *For a 2-layer network with noisy gradient and local aggregation defense with two steps, the statistical minimax risk has a lower bound $R_L \geq \tilde{\Omega}(\sigma\sqrt{\frac{d}{m}})$ with high probability.*

3.2 Differential privacy stochastic gradient descent

Differential privacy (Dwork, 2006) algorithms hides the exact local gradients by adding noise to gradient descent steps (Shokri and Shmatikov, 2015; Abadi et al.,

2016; Song et al., 2013). In differential private federated learning (Geyer et al., 2017; Wei et al., 2020a), each client clips the gradient and introduces a random Gaussian noise before updates to the cluster. In the setting of differential privacy stochastic gradient descent (DP-SGD), the update of a_j is

$$\tilde{G}_j = G_j / \max\left\{1, \frac{\|G\|}{C}\right\} + \epsilon_0, \text{ where } \epsilon_0 \sim \mathcal{N}(0, \sigma_0^2 I).$$

Here C is a constant threshold for gradient clipping. Note that under this setting, the information contained in the gradient is disrupted by gradient clipping and random noise. For the upper bound, the noisy gradient $G + \epsilon_0$ will lead to a $\tilde{O}((B + \sigma_0)\sqrt{\frac{d}{m}})$ error bound, which is the same order to original case if $\sigma_0 \leq O(\frac{B}{m})$. Gradient clipping will enhance the strength noise though itself has no defensive effect.

Proposition 3.3 (Short version for Proposition C.13). *For a 2-layer network with mild assumptions, the reconstruction error of tensor-based attack under defense DP-SGD with clipping threshold S and Gaussian noise with variance σ_0^2 has an upper bound $R_U \leq \tilde{O}((B + \sigma_0 \max\{1, \frac{\|G\|}{C}\})\sqrt{\frac{d}{m}})$ with high probability.*

In the analysis of statistical lower bound, there is already a Gaussian noise in the observed gradient so we only need to analyze the effect of gradient clipping. Similarly, the clipping changes the scaling of the random noise and the bound changes to $\tilde{\Omega}(\sigma \max\{1, \frac{\|G\|}{C}\}\sqrt{\frac{d}{m}})$. Therefore, gradient noise will be effective if the scaling of noise is large. However, this hurts the utility a lot.

3.3 Secure aggregation

In federated learning, local devices update gradients to the server individually. Secure aggregation (Bonawitz et al., 2016, 2017) is a method that clients can aggregate their gradients before the cluster, and the cluster can only access to aggregated gradient: $\tilde{G} = \frac{1}{B} \sum_{l=1}^L G_l B_l$, where G_l and B_l is the gradient update and batch size for l -th user respectively and $B = \sum_{l=1}^L B_l$. In this way, the global server is blocked from knowing each gradient, which prevents privacy leakage. Though there is no extra defensive strength compared with directly

reconstructing a large batch with size B , it cannot identify which user has the data.

3.4 Dropout

Another method to improve the defending effect is *dropout* (Hinton et al., 2012; Srivastava et al., 2014), a mechanism designed to prevent overfitting. It has been empirically discovered in (Zhu et al., 2019; Huang et al., 2021) that it can defend against reconstruction attack. It randomly drops nodes in a network with a certain probability of $1 - p$. In this way, some of the entries in the gradient turn to zero, introducing randomness to the gradient and prevent data leakage to some degree. A two-layer fully connected network with a dropout layer can be seen as a model with an effective width equal to the number of nodes $m' \approx (1 - p)m$ that has not been dropped. Then m changes into $(1 - p)m$ for both upper and lower bounds.

3.5 Gradient pruning

Gradient pruning (Sun et al., 2017; Ye et al., 2020) is a method that accelerates the computation in training by setting the small entries in gradient to zero. Thus, the gradient will be sparse, which is similar to dropout. However, the key difference is that the remaining entries of the gradient in dropout are chosen randomly yet gradient pruning drops small entries. With dropping nodes by the pruning rules, the distribution of effective weights is not Gaussian, which violates a key assumption in upper bound analysis. Therefore, there is no evidence showing that the tensor-based method can reconstruct input data with small errors under gradient pruning defense. In contrary, we can prove a lower bound with the same order when there is no defense.

Proposition 3.4 (Short version for Proposition E.3). *For a 2-layer network with noisy gradient and gradient pruning defense with pruning ratio p , the statistical minimax risk has a lower bound $R_L \geq \tilde{\Omega}(\sigma \sqrt{\frac{d}{(1-p)m}})$ with high probability. Let $J = \nabla_{\mathbf{x}_1, \dots, \mathbf{x}_B} \nabla_{\Theta} L$. $\hat{p} = \|J_{\setminus p}\|_F^2 / \|J\|_F^2 \in [0, 1]$, where $J_{\setminus p}$ is J dropping the columns corresponding to the pruned coordinates.*

No upper bound derived indicated gradient pruning is more effective for the tensor-based data reconstruction method. The lower bound also demonstrates its potential for better privacy-utility trade-off, at least compared to dropout. Note that gradient pruning only mildly hurts utility by pruning out the least important coordinates; it does not necessarily hurt reconstruction error in the same way.

4 Empirical Analysis

4.1 Attacks for Defense Methods Evaluation

To conduct a fair comparison between different defense D , we evaluate their strength in response to the strongest attack: $\mathcal{S}_D := \max_{A \in \mathcal{A}} d(S, A(D(G)))$, where \mathcal{A} is the set of attacking methods considered in this evaluation. The ideal case is for \mathcal{A} to contain the strongest possible attack; to complement existent attacks, we propose a new attack demonstrated strong for various defenses. We also include a range of attacks that are either related to our proposed attack method or widely acknowledged to be strong and robust to different defenses:

- **GradientInversion** (Geiping et al., 2020). We consider the attack proposed by (Geiping et al., 2020), which achieves good recovery results using gradient inversion with known BatchNorm statistics, as knowing such statistics often results in better recovery results and thus stronger attack (Huang et al., 2021).
- **Our proposed attack**. Our proposed attack is an improved version of GradientInversion attack. There are two modules we introduced to the original GradientInversion framework. a) a randomly initialized image prior network is introduced to generate the images, and b) the latent features of the fully connected layer are recovered first based on (Wang et al., 2023), and then inverting these features using feature inversion to recover the input images. More details will be introduced in Section 4.2.
- **CPA** (Kariyappa et al., 2023). We consider this attack method due to its similarity to our method: GradientInversion and feature inversion are both adopted to recover the input images. In contrast to our proposed attack, CPA forms the feature recovery problem as a blind source separation problem and recovers features by finding the unmixing matrix.
- **Robbing The Fed** (Fowl et al., 2021). As both the proposed method and CPA require minimal malicious modifications of the shared model, we are also interested in attacks on explicitly malicious servers.

We apply various defense techniques to the above attack algorithms, and for each defense, we consider the best attack performance across different attack algorithms following Eq. (3).

Batchsize	w/o defense	Gradprune			GradClipping			GradNoise			Local Aggregation	
		$p=0.3$	$p=0.5$	$p=0.7$	$C=2$	$C=4$	$C=8$	0.001	0.01	0.1	step=3	step=5
1	0.171	0.173 (0.002)	0.173 (0.002)	0.248 (0.006)	0.173 (0.002)	0.173 (0.002)	0.173 (0.002)	0.172 (0.003)	0.174 (0.002)	0.259 (0.012)	0.181 (0.067)	0.213 (0.071)
2	0.171	0.171 (0.003)	0.218 (0.011)	0.443 (0.120)	0.169 (0.003)	0.169 (0.003)	0.170 (0.003)	0.166 (0.003)	0.252 (0.082)	0.850 (0.108)	0.186 (0.127)	0.210 (0.087)
4	0.174	0.186 (0.114)	0.277 (0.106)	0.483 (0.075)	0.174 (0.116)	0.175 (0.127)	0.177 (0.113)	0.190 (0.021)	0.252 (0.106)	0.714 (0.102)	0.192 (0.079)	0.218 (0.081)
8	0.175	0.188 (0.044)	0.266 (0.071)	0.425 (0.043)	0.179 (0.041)	0.179 (0.041)	0.179 (0.041)	0.198 (0.114)	0.499 (0.104)	0.953 (0.108)	0.202 (0.073)	0.223 (0.069)

Table 2: Feature reconstruction error under different defenses. When using gradient pruning defense and gradient noise defense with large σ , the feature reconstruction quality degrades significantly.

4.2 Description of the proposed attack algorithm

Notice that the tensor-based reconstruction attack is provably strong, with a matching lower bound in the setting of two-layer neural networks. However, the original design in (Wang et al., 2023) was impractical, and we propose an attack that incorporates the strength of the tensor-based method and the generality of gradient inversion attacks for a comprehensive evaluation of different defenses. On top of the gradient matching term $\mathcal{L}_{\text{grad}}$ in (Geiping et al., 2020) and the prior knowledge utilized in (Yin et al., 2021) grounded in batch normalization, the key innovation of our method is an additional regularization that integrates feature reconstruction to gradient inversion. However, the integration is challenging due to different parameter requirements for gradient inversion attacks and tensor-based feature reconstruction.

Regularization on feature matching. By treating the intermediate (last but one) layer \mathbf{z}_i as the model input, we approximately reconstruct $\hat{\mathbf{z}}_i$ as a preprocessing step using the tensor-based method in (Wang et al., 2023). However, the reconstructed quality, the ordering of $\hat{\mathbf{z}}_i$, or whether it is unique is unclear. To address such challenges, we introduce a gradient-matching regularization term to align intermediate network features with reconstructed feature $\hat{\mathbf{z}}_i$. This regularization, denoted as $\mathcal{R}_{\text{feature}}(x)$, employs squared cosine similarity to mitigate the issue of sign ambiguity inherent in tensor-based reconstruction: $\mathcal{R}_{\text{feature}}(x) = 1 - \left(\frac{\langle f(x, \phi), \hat{\mathbf{z}} \rangle}{\|f(x, \phi)\| \|\hat{\mathbf{z}}\|} \right)^2$, where $f(x, \phi)$ are the intermediate features of input x and $\hat{\mathbf{z}}$ are the reconstructed features from the tensor-based method. This approach aims to refine gradient matching by ensuring the generated input x yields intermediate features similar to $\hat{\mathbf{z}}$, providing a more accurate and specific solution space. We notice the reconstruction quality of individual samples drops drastically when they fall into the same class. However, the reconstruction of their spanned space is still accurate. We further refine the regularize as $\mathcal{R}_{\text{feature}}(x) = \|P_{\text{span}(\hat{\mathbf{z}}_i)}^\perp f(x, \phi)\|^2$ when the sample size is large.

Other regularizations. Following (Geiping et al., 2020), we adopt total variation and regularization on

batch norm to exploit prior knowledge of natural images and training data. Moreover, inspired by deep image prior (Ulyanov et al., 2018), we employ an untrained Convolutional Neural Network (CNN) to generate an image, which can serve as a sufficient image prior. This idea is rather similar to (Jeon et al., 2021) where a network is pre-trained for image generation. The final objective of the introduced attack method is

$$\arg \min_x \mathcal{L}_{\text{grad}}(x; \theta, \nabla_{\theta} \mathcal{L}_{\theta}(x^*, y^*)) + \alpha_{TV} \mathcal{R}_{TV} + \alpha_{BN} \mathcal{R}_{BN} + \alpha_f \mathcal{R}_{\text{feature}}(x).$$

5 Experiments

5.1 Experiment setup

Key parameters of defenses. We evaluate defense methods on CIFAR-10 dataset (Krizhevsky et al., 2009) with a ResNet-18 (trained for one epoch), which is the default backbone model for federated learning. The details are in the Appendix F. For *GradPrune*: gradient pruning sets gradients of small magnitudes to zero. We vary the pruning ratio p in $\{0.3, 0.5, 0.7, 0.9, 0.99\}$. For *GradClipping*: gradient clipping set gradient to have a bounded norm, which is ensured by applying a clipping operation that shrinks individual model gradients when their norm exceeds a given threshold. We set this threshold as $\{2, 4, 8\}$. We also adopt defense by *adding noise to gradient*, the noise is generated from random Gaussian with different standard deviations $\{0.001, 0.01, 0.05, 0.1\}$. For *GradDropout*, we consider randomly dropout the entries of the gradient with probability in $\{0.3, 0.5, 0.7, 0.9\}$. Local gradient aggregation aggregates local gradient by performing a few steps of gradient descent to update the model. We locally aggregate 3 or 5 steps.

Key experimental setup for the attack. We use a stratified sampled subset of 50 CIFAR-10 images to evaluate the attack performance. For all attack algorithms, we assume the batch norm statistics are available following (Geiping et al., 2020). We observe the difficulty of optimization when using cosine-similarity as the gradient inversion loss for the proposed attack algorithm and therefore address this issue by reweighting the gradients by their ℓ_0 norm. See details in the Appendix F. Geiping et al., 2020; Zhu et al., 2019 have

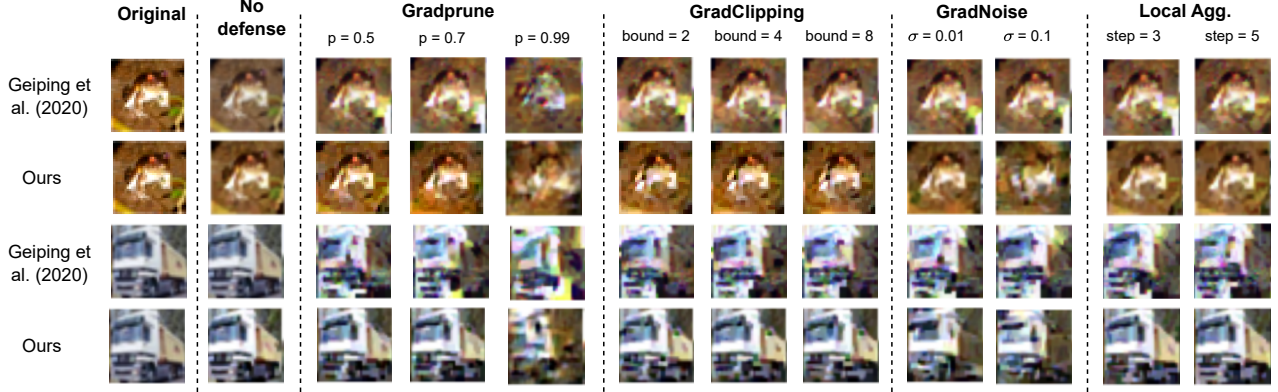


Figure 2: Comparison of the reconstruction results from the gradient inversion method (Geiping et al., 2020) and our proposed attack method on different defenses with batch size equal to 4. Our method achieves more robust reconstruction across various defenses. Gradient pruning ($p = 0.99$) makes reconstructions from both methods almost unrecognizable.

shown that small batch size is vital for the success of the attack. We intentionally evaluate the attack with two small batch sizes 4 and 8 and two small but realistic batch sizes, 16 and 32.

Metrics for reconstruction quality. We visualize reconstructions obtained under different defenses. Following (Yin et al., 2021), we also use the RMSE, PSNR, and learned perceptual image patch similarity (LPIPS) score (Zhang et al., 2018) to measure the mismatch between reconstruction and original images. For the evaluation of feature reconstruction, we use the average norm of projection from original features to the orthogonal complement of the space spanned by reconstructed features. This metric measures the difference between the two spaces spanned by original and reconstructed features. Compared with the average cosine similarity between features, this metric is more robust when some features are similar.

5.2 Experiment results

Verification of theoretical analysis. In Section 3, we analyze the performance of different defenses on 2-layer networks. In the experiments, our attack method involves intermediate feature matching, which requires reconstructing feature from a 2-layer network. This setting aligns with our theoretical analysis. Therefore, we can verify the theory by checking the feature reconstruction error in our attack. The results is shown in Table 2. The increase of the reconstruction error along with the batch size matches the theoretical bounds. Moreover, gradient pruning and gradient noise with large scaling has significantly large reconstruction error, which also aligns with our theoretical evaluation.

Batch Size	16			32		
Method	Geiping	Jeon	Ours	Geiping	Jeon	Ours
LPIPS ↓	0.41 (0.09)	0.17 (0.12)	0.14 (0.11)	0.45 (0.11)	0.24 (0.13)	0.15 (0.11)

Table 3: Comparison of our methods with other methods, Geiping, and Jeon refer to (Geiping et al., 2020) and (Jeon et al., 2021) respectively. We highlight the best performances in bold.

Our proposed attack. According to the framework of evaluation, we need to consider strong attacks against various defenses. A key motivation of our proposed attack is its good performance among different defense, which we will verify in the following results. Our method is based on gradient inversion and can be easily added to previous methods (Geiping et al., 2020; Zhu et al., 2019). In Table 3, we compare the state-of-the-art gradient inversion methods with the proposed attacking method, our method outperforms previous methods. We visualize reconstructed images in Figure 2. Without defense, both (Geiping et al., 2020) and our attack method can recover images well from the gradient, and our method produces higher-quality images. With defenses, our method shows better robustness – the recovered images are visually more similar to the original image. See Table 7 and Table 8 in Appendix I for summary of reconstruction results.

Systematic evaluation of defense. In our systematic evaluation of different defenses, we measure their strength with the strongest attack: $\mathcal{S}_D = \max_{A \in \mathcal{A}} d(S, A(D(G)))$. In order to estimate the maximum reconstruction error, we selected four different attacks with various parameters and batch sizes to conduct the experiment and record the reconstruction RMSE in Table 4. For each setting, we select the attack with the smallest RMSE as the strongest attack. Therefore, among all defenses, gradient pruning with

Data Reconstruction Attacks and Defenses: A Systematic Evaluation

		GradClip (C)			GradDrop (p)				GradNoise (σ_0)			GradPrune (p)					
Parameter		2	4	8	0.3	0.5	0.7	0.9	0.001	0.01	0.1	0.3	0.5	0.7	0.9	0.99	
$B = 2$	Ours	0.17	0.17	0.19	0.16	0.16	0.18	0.18	0.17	0.22	0.28	0.16	0.17	0.20	0.26	0.27	
	GradientInversion	0.19	0.23	0.25	0.19	0.20	0.20	0.21	0.30	0.32	0.35	0.19	0.20	0.24	0.28	0.27	
	Robbing The Fed	0.23	0.23	0.23	0.28	0.32	0.32	0.30	0.26	0.25	0.27	0.22	0.25	0.32	0.37	0.46	
	CPA	0.21	0.22	0.23	0.23	0.22	0.22	0.24	0.22	0.23	0.28	0.21	0.23	0.29	0.28	0.32	
$B = 4$	Ours	0.16	0.16	0.16	0.16	0.16	0.19	0.19	0.19	0.24	0.27	0.16	0.16	0.21	0.27	0.27	
	GradientInversion	0.21	0.22	0.23	0.21	0.21	0.21	0.22	0.31	0.31	0.31	0.19	0.18	0.23	0.29	0.28	
	Robbing The Fed	0.18	0.18	0.18	0.27	0.31	0.32	0.31	0.19	0.19	0.22	0.20	0.25	0.32	0.39	0.42	
	CPA	0.21	0.21	0.20	0.21	0.23	0.22	0.24	0.23	0.23	0.25	0.18	0.21	0.24	0.29	0.31	
$B = 8$	Ours	0.16	0.16	0.16	0.16	0.17	0.19	0.19	0.19	0.29	0.30	0.15	0.16	0.20	0.29	0.29	
	GradientInversion	0.22	0.21	0.21	0.22	0.22	0.22	0.23	0.30	0.31	0.31	0.20	0.21	0.25	0.30	0.30	
	Robbing The Fed	0.16	0.16	0.16	0.29	0.31	0.31	0.31	0.16	0.16	0.25	0.22	0.30	0.30	0.36	0.43	
	CPA	0.21	0.21	0.22	0.22	0.22	0.23	0.25	0.20	0.22	0.24	0.19	0.22	0.26	0.32	0.33	

Table 4: We run attacks under different defenses with various parameters and batch sizes. For each setting, we compared the attacks and select the one with lowest RMSE (the highlighted numbers). The best attack for each setting represents the degree of data leakage of that specific defense. Compared with the utility loss of this defense, we can systematically evaluate defenses.

	GradClip (C)			GradDrop (p)				GradNoise (σ_0)			GradPrune (p)				
Parameter	2	4	8	0.3	0.5	0.7	0.9	0.001	0.01	0.1	0.3	0.5	0.7	0.9	0.99
Attack batch size = 2															
Final training loss ↓	0.390	0.481	0.377	0.466	0.363	0.564	0.864	0.445	0.769	1.889	0.585	0.430	0.621	0.763	1.020
Attack batch size = 4															
Final training loss ↓	0.408	0.353	0.329	0.552	0.316	0.275	0.460	0.527	0.551	1.764	0.253	0.389	0.561	0.505	0.935
Attack batch size = 8															
Final training loss ↓	0.377	0.201	0.357	0.259	0.178	0.231	0.461	0.365	0.343	1.540	0.256	0.161	0.257	0.344	0.725

Table 5: The final training loss of the model with defenses, which measures the utility loss from defenses. Gradient noise produces much larger interference than gradient pruning.

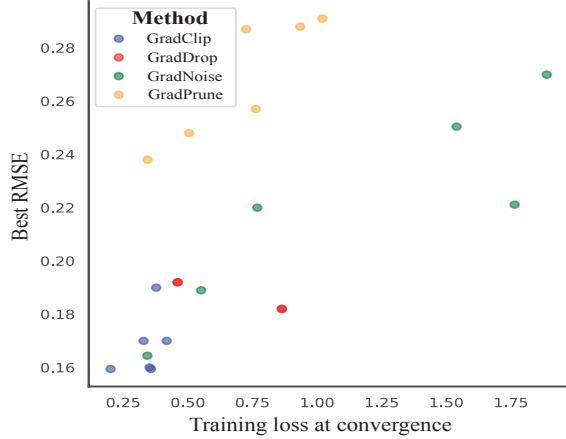


Figure 3: The relation between the strength of the defense and the utility loss, where we use the best RMSE and the final training loss of the original task respectively. Each dot represents a defense method with a different batch size and strength. For the same level of utility loss, gradient pruning has the best defending effect.

large p and gradient noise with large variance σ_0^2 has the best defensive effect.

In order to evaluate the defenses, another key criteria is the utility loss of the defense methods. Most of the defenses prevent models from data leakage by perturbing the gradients in training, which may hurt the training

task itself. In our experiment, we measure the utility by the final loss of the training task and we show it in Table 5. Gradient pruning, as one of the defenses having strongest effect towards attacks, causes smaller utility loss than gradient noise. This result indicates that gradient pruning is a better defense than gradient noise. In Figure 3, we plot the relation between the reconstruction error and the utility loss, where each dot represents a method under the setting of a specific batch size and parameter of the defense. With the same level of utility loss, gradient pruning has the largest RMSE, indicating it is the best defense to have $\min_{D \in \mathcal{D}_U} \mathcal{S}_D$.

Note that there are various of attacks to reconstruction input data and we cannot try all of them on defenses. However, within our framework of evaluation, newly proposed attacks can be added in the experiment and will improve the strength of defenses \mathcal{S}_D . Therefore, we call for more empirical analysis on top of our results with more attack methods, in order to produce a more accurate evaluation.

6 Conclusion

In this paper, we systematically evaluate various defense methods, both theoretically and empirically. We established the algorithmic upper bound and matching

information-theoretic lower bound of the reconstruction error for various defenses on two-layer random nets, thereby quantitatively analyzing the effect of defense strength, data dimension, and model width. To extend our theoretical insights to more general network architectures and explore the utility-privacy trade-off, we introduced a robust attack method capable of overcoming a broad spectrum of defense strategies in a setting with an honest but curious server. The method enhances the conventional gradient inversion attack by combining it with the powerful feature reconstruction attack that achieves our algorithmic upper bound. We propose to evaluate each attack based on their impact on utility and effectiveness against the strongest corresponding attack. Our evaluation indicates gradient pruning is the strongest among our considered defenses and against our considered attacks, which updates some previous evaluation results purely based on gradient inversion attacks (Huang et al., 2021). Our work establishes a more comprehensive understanding of the data reconstruction problem in federated learning. We anticipate that our proposed framework will encourage further research and additional evaluations building on our findings.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science Energy Earthshot Initiative as part of the project “Learning reduced models under extreme data conditions for design and rapid decision-making in complex systems” under Award #DE-SC0024721.

References

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.
- Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al. (2017). Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345.
- Balunovic, M., Dimitrov, D. I., Staab, R., and Vechev, M. (2021). Bayesian framework for gradient leakage. In *International Conference on Learning Representations*.
- Boenisch, F., Dziedzic, A., Schuster, R., Shamsabadi, A. S., Shumailov, I., and Papernot, N. (2023). When the curious abandon honesty: Federated learning is not private. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 175–199. IEEE.
- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2016). Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*.
- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191.
- Cramér, H. (1999). *Mathematical methods of statistics*, volume 26. Princeton university press.
- Dwork, C. (2006). Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings, Part II 33*, pages 1–12. Springer.
- Fowl, L., Geiping, J., Czaja, W., Goldblum, M., and Goldstein, T. (2021). Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*.
- Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947.
- Geyer, R. C., Klein, T., and Nabi, M. (2017). Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.
- Guo, C., Karrer, B., Chaudhuri, K., and van der Maaten, L. (2022). Bounding training data reconstruction in private (deep) learning. In *International Conference on Machine Learning*, pages 8056–8071. PMLR.
- Haim, N., Vardi, G., Yehudai, G., Shamir, O., and Irani, M. (2022). Reconstructing training data from trained neural networks. *arXiv preprint arXiv:2206.07758*.
- Hayes, J., Balle, B., and Mahloujifar, S. (2024). Bounding training data reconstruction in dp-sgd. *Advances in Neural Information Processing Systems*, 36.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. (2021). Evaluating gradient inversion attacks and

- defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241.
- Jeon, J., Lee, K., Oh, S., Ok, J., et al. (2021). Gradient inversion with generative image prior. *Advances in neural information processing systems*, 34:29898–29908.
- Kariyappa, S., Guo, C., Maeng, K., Xiong, W., Suh, G. E., Qureshi, M. K., and Lee, H.-H. S. (2023). Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis. In *International Conference on Machine Learning*, pages 15884–15899. PMLR.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *ICLR 2015*.
- Konečný, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*.
- Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.
- Kuleshov, V., Chaganty, A., and Liang, P. (2015). Tensor factorization via matrix factorization. In *Artificial Intelligence and Statistics*, pages 507–516. PMLR.
- Li, H., Guo, D., Fan, W., Xu, M., and Song, Y. (2023). Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*.
- Li, Z., Zhang, J., Liu, L., and Liu, J. (2022). Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10132–10142.
- Mamis, K. (2022). Extension of stein’s lemma derived by using an integration by differentiation technique. *Examples and Counterexamples*, 2:100077.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.
- Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671.
- Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. (2016). Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*.
- Rao, C. R. (1992). Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in Statistics: Foundations and basic theory*, pages 235–247. Springer.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.
- Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*.
- Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29.
- Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321.
- Song, S., Chaudhuri, K., and Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151.
- Stock, P., Shilov, I., Mironov, I., and Sablayrolles, A. (2022). Defending against reconstruction attacks with r\`enyi differential privacy. *arXiv preprint arXiv:2202.07623*.
- Sun, X., Ren, X., Ma, S., and Wang, H. (2017). meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In *International Conference on Machine Learning*, pages 3299–3308. PMLR.
- Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2018). Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454.
- Wang, Y., Deng, J., Guo, D., Wang, C., Meng, X., Liu, H., Ding, C., and Rajasekaran, S. (2020). Sapag: A self-adaptive privacy attack from gradients. *arXiv preprint arXiv:2009.06228*.
- Wang, Z., Lee, J., and Lei, Q. (2023). Reconstructing training data from model gradient, provably. In *International Conference on Artificial Intelligence and Statistics*, pages 6595–6612. PMLR.

- Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q., and Poor, H. V. (2020a). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469.
- Wei, W., Liu, L., Loper, M., Chow, K.-H., Gursoy, M. E., Truex, S., and Wu, Y. (2020b). A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397*.
- Wen, Y., Geiping, J., Fowl, L., Goldblum, M., and Goldstein, T. (2022). Fishing for user data in large-batch federated learning via gradient magnification. *arXiv preprint arXiv:2202.00580*.
- Wu, Y. and He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.
- Xue, D., Yang, H., Ge, M., Li, J., Xu, G., and Li, H. (2023). Fast generation-based gradient leakage attacks against highly compressed gradients. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE.
- Ye, X., Dai, P., Luo, J., Guo, X., Qi, Y., Yang, J., and Chen, Y. (2020). Accelerating cnn training by pruning activation gradients. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 322–338. Springer.
- Yin, H., Mallya, A., Vahdat, A., Alvarez, J. M., Kautz, J., and Molchanov, P. (2021). See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346.
- Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.
- Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.
- Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. (2017). Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pages 4140–4149. PMLR.
- Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable] Yes. We include the setting in Section 2 and the methods used in experiments in Section 4.2.
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable] Not Applicable. Our work does not focus on the analysis of computation.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable] Yes. We provide full set of assumptions in appendix.
 - (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable] Yes. We provide complete proofs in appendix.
 - (c) Clear explanations of any assumptions. [Yes/No/Not Applicable] Yes. We explain all the assumptions in Section 3 and appendix.
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable] Yes. We use benchmarking public dataset, and we will keep our code public accessible.
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable] Yes. We include all the implementation details in the appendix.
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable] Yes. We provide a standard error in the empirical results in appendix.
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable] Yes. We include all the computing and training configurations in the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable] Yes. We cite all existing assets we used.
 - (b) The license information of the assets, if applicable. [Yes/No/Not Applicable] Not Applicable.
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable] Yes. No new data is involved. We have documented our code well and will make it publicly accessible.
 - (d) Information about consent from data providers/curators. [Yes/No/Not Applicable] Not Applicable. All assets are public for research purpose.
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable] Not Applicable. This work does not involve sensible content.
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable] Not Applicable.
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable] Not Applicable.
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable] Not Applicable.

A Analysis of Differential Privacy

Differential privacy (DP) (Dwork, 2006) is a measure of privacy mainly in membership inference attack. A random algorithm $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ satisfies (ϵ, δ) -DP if for any $\mathcal{Q} \subset \mathcal{R}$ and any adjacent inputs $S, S' \in \mathcal{D}$ holds that

$$\mathbb{P}(\mathcal{M}(S) \in \mathcal{Q}) \leq e^\epsilon \mathbb{P}(\mathcal{M}(S') \in \mathcal{Q}) + \delta.$$

Here adjacent inputs means only one of the samples is different.

In reconstruction attack, DP is too strong that an algorithm may prevent data leakage without DP guarantees. Moreover, the price to guarantee DP is high. A popular method to guarantee DP is DP-SGD (Abadi et al., 2016). However, it requires extremely large Gaussian noise.

Proposition A.1 (Full version of Proposition 2.1). *We define a two-layer neural network $f(\mathbf{x}; \Theta) = \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x})$ with input dimension d , random Gaussian weights $a_j \sim \mathcal{N}(0, \frac{1}{m^2})$ and $\mathbf{w}_j \sim \mathcal{N}(0, I_d)$ and activation function σ is 1-Lipschitz. The loss function ℓ is square loss, the data $\|\mathbf{x}\| = 1$ and the label $y \in \{\pm 1\}$. We denote the gradient of ℓ by G . The randomized mechanism $\mathcal{M} = G + \mathcal{N}(0, \sigma^2 I)$ is (ϵ, δ) -DP for any $\epsilon, \delta > 0$ if $\sigma^2 = \Omega(\frac{m \log(1/\delta)}{\epsilon})$ with high probability with respect to random weights.*

In order to prove the proposition, we first define

$$\alpha_{\mathcal{M}}(\lambda) := \max_{S, S'} \log \mathbb{E}_{z \sim \mathcal{M}(S)} \left(\frac{p(\mathcal{M}(S) = z)}{p(\mathcal{M}(S') = z)} \right)^\lambda,$$

where p is the density function. In DP-SGD setting, $\mathcal{M}(S) = G(S) + \mathcal{M}(0, \sigma^2 I)$ so $\mathcal{M}(S) \sim \mathcal{N}(G(S), \sigma^2 I)$ for $S \in \mathbb{R}^{d \times B}$. The following lemma is a key step in the proof of Proposition A.1.

Lemma A.2 (Theorem 2 in (Abadi et al., 2016)). *For any $\epsilon > 0$, the mechanism \mathcal{M} is (ϵ, δ) -DP for $\delta = \min_\lambda (\alpha_{\mathcal{M}}(\lambda) - \lambda\epsilon)$.*

Proof of Proposition A.1. We denote $\Delta := \max_{\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d} \|G(\mathbf{x}) - G(\mathbf{x}')\|^2$. Then we compute $\alpha_{\mathcal{M}}(\lambda)$ with adjacent $S, S' \in \mathbb{R}^{d \times B}$. We denote the only different element by \mathbf{x} and \mathbf{x}' .

$$\begin{aligned} \alpha_{\mathcal{M}}(\lambda) &= \max_{S, S'} \log \mathbb{E}_{Z \sim \mathcal{N}(G(S), \sigma^2 I)} \left[\left(\frac{p_S(Z)}{p_{S'}(Z)} \right)^\lambda \right] \\ &= \max_{\mathbf{x}, \mathbf{x}'} \log \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(G(\mathbf{x}), I)} \exp \left(\frac{\lambda}{2\sigma^2} \sum_{i=1}^d \left((G(\mathbf{x})_i - z_i)^2 - (G(\mathbf{x}')_i - z_i)^2 \right) \right) \\ &= \max_{\mathbf{x}, \mathbf{x}'} \log \frac{1}{(\sqrt{2\pi}\sigma)^d} \int \exp \left(\frac{1}{2\sigma^2} \sum_{i=1}^d \left((\lambda + 1)(G(\mathbf{x})_i - z_i)^2 - \lambda(G(\mathbf{x}')_i - z_i)^2 \right) \right) \\ &= \max_{\mathbf{x}, \mathbf{x}'} \log \frac{1}{(\sqrt{2\pi}\sigma)^d} \int \exp \left(\frac{1}{2\sigma^2} \sum_{i=1}^d \left((z_i - ((\lambda + 1)G(\mathbf{x})_i - \lambda G(\mathbf{x}')_i))^2 - \lambda(\lambda + 1)(G(\mathbf{x})_i - G(\mathbf{x}')_i)^2 \right) \right) \\ &= \max_{\mathbf{x}, \mathbf{x}'} \log \exp \left(\frac{\lambda(\lambda + 1)}{2\sigma^2} \sum_{i=1}^d (G(\mathbf{x})_i - G(\mathbf{x}')_i)^2 \right) \\ &= \max_{\mathbf{x}, \mathbf{x}'} \frac{\lambda(\lambda + 1)}{2\sigma^2} \|G(\mathbf{x}) - G(\mathbf{x}')\|^2 = \frac{\lambda(\lambda + 1)}{w\sigma^2} \Delta. \end{aligned}$$

By Lemma A.2, for any

$$\begin{aligned} \delta &\geq \min_\lambda \exp \left(\frac{\lambda(\lambda + 1)}{2\sigma^2} \Delta - \lambda\epsilon \right) \\ &= \exp \left(-\frac{\Delta}{2\sigma^2} \left(\frac{\sigma^2 \epsilon}{\Delta} - \frac{1}{2} \right)^2 \right), \end{aligned}$$

\mathcal{M} is (ϵ, δ) -DP. Then for $\sigma^2 \geq \Omega(\frac{\Delta \log(1/\delta)}{\epsilon})$, \mathcal{M} is (ϵ, δ) -DP.

Now we only need to bound Δ . We denote $r = y - f(\mathbf{x})$ and $r' = y' - f(\mathbf{x}')$. We consider ∇_{a_j} and $\nabla_{\mathbf{w}_j}$ separately. For ∇_{a_j} , we have

$$\begin{aligned} |\nabla_{a_j} \ell(S) - \nabla_{a_j} \ell(S')|^2 &= |\nabla_{a_j} \ell(\mathbf{x}) - \nabla_{a_j} \ell(\mathbf{x}')|^2 \\ &= |r\sigma(\mathbf{w}_j^\top \mathbf{x}) - r'\sigma(\mathbf{w}_j^\top \mathbf{x}')|^2 \\ &\lesssim r^2 |\sigma(\mathbf{w}_j^\top \mathbf{x})|^2 + r'^2 |\sigma(\mathbf{w}_j^\top \mathbf{x}')|^2 \\ &= \tilde{O}(1) \end{aligned}$$

with high probability for any j . For $\nabla_{\mathbf{w}_j}$, we have

$$\begin{aligned} \|\nabla_{\mathbf{w}_j} \ell(S) - \nabla_{\mathbf{w}_j} \ell(S')\|^2 &= \|\nabla_{\mathbf{w}_j} \ell(\mathbf{x}) - \nabla_{\mathbf{w}_j} \ell(\mathbf{x}')\|^2 \\ &= \|r\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{x} - r'\sigma'(\mathbf{w}_j^\top \mathbf{x}')\mathbf{x}'\|^2 \\ &\lesssim r^2 \|\mathbf{x}\|^2 + r'^2 \|\mathbf{x}'\|^2 \\ &= \tilde{O}(1) \end{aligned}$$

with high probability for any j . Then

$$\begin{aligned} \Delta &= \max_{S, S'} \|G(S) - G(S')\|^2 \\ &= \max_{S, S'} \sum_{j=1}^m \left(|\nabla_{a_j} \ell(S) - \nabla_{a_j} \ell(S')|^2 + \|\nabla_{\mathbf{w}_j} \ell(S) - \nabla_{\mathbf{w}_j} \ell(S')\|^2 \right) \\ &\leq \tilde{O}(m) \end{aligned}$$

with high probability. Therefore, for $\sigma^2 \geq \tilde{\Omega}(\frac{m \log(1/\delta)}{\epsilon})$, \mathcal{M} is (ϵ, δ) -DP with high probability. \square

B Analysis of Reconstruction Upper Bound

B.1 Tensor Method

Various of noisy tensor decomposition can be used in tensor based attack (Wang et al., 2023). We select the method proposed by (Zhong et al., 2017) in our theoretical analysis since it can provably achieve a relatively small error with least assumptions.

Assumption B.1. *We make the following assumptions:*

- **Data:** Let data matrix $S := [\mathbf{x}_1, \dots, \mathbf{x}_B] \in \mathbb{R}^{d \times B}$, we denote the B -th singular value by $\pi_{\min} > 0$. Training samples are normalized: $\|\mathbf{x}_i\| = 1, \forall i \in [B]$.
- **Activation:** σ is 1-Lipschitz and $\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma''(z)] < \infty$. Let

$$k_2 = \min\{k \geq 2 : |\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma^{(k)}(z)]| \neq 0\}$$

and

$$k_3 = \min\{k \geq 3 : |\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma^{(k)}(z)]| \neq 0\}.$$

Then $\nu = |\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma^{(k_2)}(z)]|$ and $\lambda = |\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma^{(k_3)}(z)]|$ are not zero. We assume $k_2 \leq 3$ and $k_3 \leq 4$.

We first introduce the method in (Zhong et al., 2017) when the activation function satisfies $k_2 = 2$ and $k_3 = 3$. In addition to $\hat{\mathbf{T}} = \sum_{j=1}^m g(\mathbf{w}_j) H_3(\mathbf{w}_j)$, we also need the matrix $\hat{P} = \sum_{j=1}^m g(\mathbf{w}_j) H_2(\mathbf{w}_j)$ in the tensor decomposition, where H_p is the p -th Hermite function. We first conduct the power method to \hat{P} to estimate the orthogonal span U of training samples $\mathbf{x}_1, \dots, \mathbf{x}_B$ and denote it by V , where $V \in \mathbb{R}^{B \times d}$ is an orthogonal matrix. Then we conduct tensor decomposition with $\hat{\mathbf{T}}(V, V, V)$ instead of T and have the estimation of $\{V^\top \mathbf{x}_i\}_{i=1}^B$. By multiplying column orthogonal matrix V , we can reconstruct training data. The advantage of this method is that the dimension of $\hat{\mathbf{T}}(V, V, V)$ is $B < d$. Then the error will depend on B instead of d .

For activation functions such that $k_2 = 3$, we define $\hat{P} = \sum_{j=1}^m g(\mathbf{w}_j)H_3(\mathbf{w}_j)(I, I, \mathbf{a})$ and for activation functions such that $k_3 = 4$, we define $\hat{T} = \sum_{j=1}^m g(\mathbf{w}_j)H_4(\mathbf{w}_j)(I, I, I, \mathbf{a})$, where \mathbf{a} is any unit vector. In the proofs below, we only consider the case $k_2 = 2$ and $k_3 = 3$ while the proofs of other settings are similar.

Note that with this method only, we cannot identify the norm of recovered samples without the assumption of $\|\mathbf{x}_i\| = 1$ for all i . However, in our stronger reconstruction attack, the feature matching term is the cosine similarity between the reconstructed feature and the dummy feature, where knowing the norm is not necessary. Thus, our assumption on $\|b\mathbf{x}_i\| = 1$ is reasonable and can simplify the method and the proof.

B.2 Error Bound

For the noisy tensor decomposition introduced above, we have the following error bound:

Theorem B.1 (Adapted from the proof of Theorem 5.6 in (Zhong et al., 2017)). *Consider matrix $\hat{P} = P + S$ and tensor $\hat{T} = T + E$ with rank- B decomposition*

$$P = \sum_{i=1}^B \nu_i \mathbf{x}_i \mathbf{x}_i^\top, \quad T = \sum_{i=1}^B \lambda_i \mathbf{x}_i^{\otimes 3},$$

where $\mathbf{x}_i \in \mathbb{R}^d$ satisfying Assumption B.1. Let V be the output of Algorithm 3 in (Zhong et al., 2017) with input P and $\{s_i \mathbf{u}_i\}_{i=1}^B$ be the output of Algorithm 1 in (Kuleshov et al., 2015) with input $T(V, V, V)$, where $\{s_i\}$ are unknown signs. Suppose the perturbations satisfy

$$\|S\| \leq \mu \leq O(\nu_{\min} \pi_{\min}), \quad \|E(V, V, V)\| \leq \gamma.$$

Let $N = \Theta(\log \frac{1}{\epsilon})$ be the iteration numbers of Algorithm 3 in (Zhong et al., 2017), where $\epsilon = \frac{\mu}{\nu_{\min}}$. Then with high probability, we have

$$\|\mathbf{x}_i - s_i V \mathbf{u}_i\| \leq \tilde{O}\left(\frac{\mu}{\nu_{\min} \pi_{\min}}\right) + \tilde{O}\left(\frac{\kappa \gamma \sqrt{B}}{\lambda_{\min} \pi_{\min}^2}\right),$$

where $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$.

With Theorem B.1, we only need to bound the error $\|P - \hat{P}\|$ and $\|T - \hat{T}\|$, where the bound varies under different settings. Wang et al., 2023 proved the error bound under the square loss setting. In the following section, we will propose error bounds with cross-entropy loss, which is more widely used in classification problems. We will also give error bounds when defending strategies are used against privacy attacks.

B.3 Matrix Bernstein's Inequality

The following lemma is crucial to the concentration bounds $\|P - \hat{P}\|$ and $\|T - \hat{T}\|$.

Lemma B.2 (Matrix Bernstein for unbounded matrices; adapted from Lemma B.7 in (Zhong et al., 2017)). *Let \mathcal{Z} denote a distribution over $\mathbb{R}^{d_1 \times d_2}$. Let $d = d_1 + d_2$. Let Z_1, Z_2, \dots, Z_m be i.i.d. random matrices sampled from \mathcal{Z} . Let $\bar{Z} = \mathbb{E}_{Z \sim \mathcal{Z}}[Z]$ and $\hat{Z} = \frac{1}{m} \sum_{i=1}^m Z_i$. For parameters $\delta_0 \in (0, 1)$, $M = M(\delta_0, m) \geq 0$, $\nu > 0$, $L > 0$, if the distribution \mathcal{B} satisfies the following four properties,*

$$\begin{aligned} (I) \quad & \mathbb{P}_{Z \sim \mathcal{Z}} \{\|Z\| \leq M\} \geq 1 - \frac{\delta_0}{m} \\ (II) \quad & \max \left(\left\| \mathbb{E}_{Z \sim \mathcal{Z}} [ZZ^\top] \right\|, \left\| \mathbb{E}_{Z \sim \mathcal{Z}} [Z^\top Z] \right\| \right) \leq \nu \\ (III) \quad & \max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} \left(\mathbb{E}_{Z \sim \mathcal{Z}} \left[(\mathbf{a}^\top Z \mathbf{b})^2 \right] \right)^{1/2} \leq L \end{aligned}$$

Then we have for any $0 < \delta_1 < 1$, if $\delta_1 \leq \frac{1}{d}$ and $m \gtrsim M \log(1/\delta_1)$, with probability at least $1 - \delta_1 - \delta_0$,

$$\|\hat{Z} - \bar{Z}\| \lesssim \sqrt{\frac{\log(1/\delta_1)(\nu + \|\bar{Z}\|^2 + M + \delta_0 L^2)}{m}}$$

B.4 Reconstruction Upper Bounds

Wang et al., 2023 proposed a privacy attack based on tensor decomposition. We adopt the setting of that work. Let the input dimension be d . For a two layer neural network $f(\mathbf{x}; \Theta) = \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x})$ with m hidden nodes and σ is point-wise. Then square loss with B samples is $\ell(f(\mathbf{x}; \Theta), y) = \sum_{i=1}^B (y_i - f(\mathbf{x}_i; \Theta))^2$. Then we have the gradient

$$\frac{\partial \ell}{\partial \mathbf{w}_j} = \sum_{i=1}^B r_i a_j \sigma'(\mathbf{w}_j^\top \mathbf{x}_i) \mathbf{x}_i$$

and

$$\frac{\partial \ell}{\partial a_j} = \sum_{i=1}^B r_i \sigma(\mathbf{w}_j^\top \mathbf{x}_i),$$

where $r_i = 2(f(\mathbf{x}_i; \Theta) - y_i)$.

To reconstruct training data, we set the weights to be random Gaussian.

Assumption B.2. *The network $f(\mathbf{x}; \Theta) = \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x})$. The parameters of the network satisfy $a_j \sim \mathcal{N}(0, \frac{1}{m^2})$ and $\mathbf{w}_j \sim \mathcal{N}(0, I_d)$ and are independent.*

For each \mathbf{w}_j , we define $g(\mathbf{w}_j) = \sum_{i=1}^B r_i \sigma(\mathbf{w}_j^\top \mathbf{x}_i)$. We can use the same attacking method in (Wang et al., 2023) that conduct noisy tensor decomposition to $\sum_{j=1}^m g(\mathbf{w}_j) H_3(\mathbf{w}_j)$, where H_p is the p -th Hermite function. In our method, we will mainly use $H_2(\mathbf{w}) = \mathbf{w} \mathbf{w}^\top - I$ and $H_3(\mathbf{w}) = \mathbf{w}^{\otimes 3} - \mathbf{w} \hat{\otimes} I$, where $\mathbf{w} \hat{\otimes} I(i, j, k) = w_i \delta_{jk} + w_j \delta_{ki} + w_k \delta_{ij}$.

Lemma B.3 (Stein's Lemma). *Let X be a standard normal random variable. Then for any function g , we have*

$$\mathbb{E}[g(X) H_p(X)] = \mathbb{E}[g^{(p)}(X)], \quad (4)$$

if both sides of the equation exist. Here H_p is the p th Hermite function and $g^{(p)}$ is the p th derivative of g .

We define

$$\hat{P} = \frac{1}{m} \sum_{j=1}^m g(\mathbf{w}_j) H_2(\mathbf{w}_j) \quad (5)$$

$$P = \mathbb{E} \left[\sum_{i=1}^B r_i^* \sigma''(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i \mathbf{x}_i^\top \right] \quad (6)$$

$$\hat{T} = \frac{1}{m} \sum_{j=1}^m g(\mathbf{w}_j) H_3(\mathbf{w}_j) \quad (7)$$

$$T = \mathbb{E} \left[\sum_{i=1}^B r_i^* \sigma^{(3)}(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i^{\otimes 3} \right]. \quad (8)$$

Then with Lemma B.3 and concentration bounds, we have the following lemmas.

Lemma B.4 (Proposition 5.5 in (Wang et al., 2023)). *Under Assumption B.1 and B.2, $|y_i| \leq 1$, then for $\delta \leq \frac{2}{d}$ and $m \gtrsim \log(8/\delta)$, we have*

$$\|\hat{P} - P\| \leq \tilde{O}\left(\frac{B\sqrt{d}}{\sqrt{m}}\right)$$

with probability $1 - \delta$.

Lemma B.5 (Proposition 5.6 in (Wang et al., 2023)). *Under Assumption B.1 and B.2, $|y_i| \leq 1$, and $\|VV^\top - UU^\top\| \leq 1/4$, then for $\delta \leq \frac{2}{B}$ and $m \gtrsim \log(6/\delta)$ we have*

$$\|\bar{T}(V, V, V) - T(V, V, V)\| \leq \tilde{O}\left(\frac{B^{5/2}}{\sqrt{m}}\right)$$

with probability $1 - \delta$.

Combining with Theorem B.1, we have an upper bound for data reconstruction error.

Theorem B.6 (Theorem 5.1 in (Wang et al., 2023), full version of Theorem 3.1). *Under Assumption B.1 and B.2, $y_i \in \{\pm 1\}$, if we have $B \leq \tilde{O}(d^{1/4})$ and $m \geq \tilde{\Omega}(\frac{d}{\min\{\nu^2, \lambda^2\}\pi_{\min}^4})$, then for $\delta \leq \frac{2}{d}$ and $m \gtrsim \log(8/\delta)$, the output of tensor based reconstruction satisfies:*

$$\sqrt{\frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2} \leq \frac{1}{\min\{|\nu|, |\lambda|\}\pi_{\min}^2} \tilde{O}\left(\sqrt{\frac{d}{m}}\right)$$

with probability $1 - \delta$.

C Proofs of Upper Bounds with Defenses

C.1 Local Aggregation

For local aggregation, the observation is $\Theta^{(2)} - \Theta^{(0)}$. In our attack method, we conduct tensor decomposition to $\tilde{g} = -\sum_{j=1}^m (a_j^{(2)} - a_j^{(0)}) H_3(\mathbf{w}_j)$. Here a notation with superscript (k) means the corresponding value after k steps of iteration. Then we have the following reconstruction error bounds under the setting of two gradient descent steps with a same batch of input and with two different batches input.

Proposition C.1 (Full version of Proposition 3.1). *Under Assumption B.1 and B.2, $y_i \in \{\pm 1\}$, the observed update $\Theta^{(2)} - \Theta^{(0)}$ is the result of 2 gradient descent steps trained a same batch with size B , where $B^4 \leq \tilde{O}(d)$. If $m \geq \tilde{\Omega}(\frac{B^2 d}{\nu^2 \pi_{\min}^2})$ and we set learning rate for two layers with different scales $\eta_a = O(\frac{1}{m^2})$ and $\eta_w = O(1)$, then with appropriate tensor decomposition methods and proper weights, we can reconstruct input data with the error bound:*

$$\sqrt{\frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2} \leq \frac{1}{\min\{|\nu|, |\lambda|\}\pi_{\min}^2} \tilde{O}\left(B \sqrt{\frac{d}{m}}\right) \quad (9)$$

with high probability.

Proposition C.2. *Under Assumption B.1 and B.2, $y_i \in \{\pm 1\}$, the observed update $\Theta^{(2)} - \Theta^{(0)}$ is the result of 2 gradient descent steps, where the first step is trained with B data and the second step is trained with $N - B$ data. Here $B < N \leq 2B$ and $B^4 \leq \tilde{O}(d)$. If $m \geq \tilde{\Omega}(\frac{B^2 d}{\nu^2 \pi_{\min}^2})$ and we set learning rate for two layers with different scales $\eta_a = O(\frac{1}{m^2})$ and $\eta_w = O(1)$, then with appropriate tensor decomposition methods and proper weights, we can reconstruct input data with the error bound:*

$$\sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2} \leq \frac{1}{\min\{|\nu|, |\lambda|\}\pi_{\min}^2} \tilde{O}\left(N \sqrt{\frac{d}{m}}\right) \quad (10)$$

with high probability.

Remark C.3. *In these propositions we assume that the learning rate for the two layers are different since $a_j \sim \mathcal{N}(0, \frac{1}{m^2})$ but $\mathbf{w}_j \sim \mathcal{N}(0, I_d)$ have different scaling. If we use a single learning rate η for all parameters, the results still hold if we assume $\eta = O(\frac{1}{m^2})$.*

To prove the propositions, we have to verify that the parameters after a gradient descent step will not change too much from the original ones. We first introduce some lemmas.

The first lemma show that $\mathbf{w}^{(1)}$ has a similar effect to \mathbf{w} in the reconstruction.

Lemma C.4. *If the learning rate $\eta_w = O(1)$, we have*

$$|\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| \leq \tilde{O}\left(\frac{B}{m}\right) \quad (11)$$

for any i and j with probability $1 - \frac{\delta}{Bm}$.

Proof. Note that $\mathbf{w}_j^{(1)} = \mathbf{w}_j - \eta_w \sum_{i=1}^B a_j \sigma'(\mathbf{w}_j^\top \mathbf{x}_i) \mathbf{x}_i$. Then

$$\begin{aligned} & |\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| \\ &= \left| \sigma \left(\mathbf{w}_j^\top \mathbf{x}_i - \eta_w \sum_{i=1}^B a_j \sigma'(\mathbf{w}_j^\top \mathbf{x}_i) \right) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| \\ &\leq \eta_w \sum_{i=1}^B a_j \sigma'(\mathbf{w}_j^\top \mathbf{x}_i) \leq \tilde{O}\left(\frac{B}{m}\right) \end{aligned} \quad (12)$$

with probability $1 - \frac{\delta}{Bm}$ for all i, j . \square

Then we give lemmas showing that $r_i^{(1)}$ is very close to r_i and the difference only changes a little in the reconstruction.

Lemma C.5. *If the learning rate $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^2})$, we have*

$$|r_i^{(1)} - r_i| \leq \tilde{O}\left(\frac{B}{m}\right) \quad (13)$$

with probability $1 - \frac{\delta}{Bm}$ for any i .

Proof. We have $r_i = \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x}_i) - y_i$ and $r_i^{(1)} = \sum_{j=1}^m a_j^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - y_i$. Note that $a_{kj}^{(1)} = a_{kj} - \eta_a \sum_{i=1}^B \sigma(\mathbf{w}_j^\top \mathbf{x}_i)$. Then by Lemma C.4 we have

$$\begin{aligned} |r_i^{(1)} - r_i| &= \left| \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x}_i) - \sum_{j=1}^m \left(a_j - \eta_a \sum_{i=1}^B \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right) \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) \right| \\ &\leq \sum_{j=1}^m a_j \left| \sigma(\mathbf{w}_j^\top \mathbf{x}_i) - \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) \right| + \eta_a \sum_{j=1}^m \sum_{i=1}^B (\sigma(\mathbf{w}_j^\top \mathbf{x}_i))^2 \\ &\quad + \eta_a \sum_{j=1}^m \sum_{i=1}^B |\sigma(\mathbf{w}_j^\top \mathbf{x}_i)| |\sigma(\mathbf{w}_j^\top \mathbf{x}_i) - \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i)| \\ &\leq \tilde{O}\left(\frac{B}{m}\right) + \tilde{O}\left(\frac{B \log(2Bm^2/\delta)}{m}\right) + \tilde{O}\left(\frac{B \log(2Bm^2/\delta)}{m^2}\right) \leq \tilde{O}\left(\frac{B}{m}\right) \end{aligned} \quad (14)$$

with probability $1 - \frac{\delta}{Bm}$. \square

Corollary C.6. *If the learning rate $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^2})$, we have $r_i^{(1)} = \tilde{O}(1)$ with probability $1 - \frac{\delta}{Bm}$.*

Proof. We have $r_i = \tilde{O}(1)$ with probability $1 - \frac{\delta}{Bm}$. Then the result holds directly by Lemma C.5. \square

Lemma C.7. *If the learning rate $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^2})$, we have*

$$\left\| \frac{1}{m} \sum_{j=1}^m r_i^{(1)} \left| \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \leq \tilde{O}\left(\frac{Bd}{m}\right) \quad (15)$$

with probability $1 - \frac{\delta}{2B}$ for any i .

Proof. We define $Z_j = r_i^{(1)} |\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| (\mathbf{w}_j \mathbf{w}_j^\top - I)$, then we have

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j \right\| \leq \|\mathbb{E} Z_j\| + \left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\|.$$

We first bound $\|\mathbb{E} Z_j\|$. By Lemma C.4 and Corollary C.6 we have

$$\|\mathbb{E} Z_j\| \lesssim \mathbb{E} |\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| \|\mathbf{w}_j \mathbf{w}_j^\top - I\| \lesssim \frac{Bd}{m}. \quad (16)$$

Next we bound $\|\frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j\|$ with matrix Bernstein inequality. We first check the conditions of Theorem B.2 with Lemma C.4 and Corollary C.6

(I) We first bound the norm of Z_j . BWe have

$$\|Z_j\| \leq 2|\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)|(\|\mathbf{w}_j\|^2 + 1) \lesssim \frac{Bd \log(4Bm/\delta)}{m}$$

with probability $1 - \frac{\delta}{2Bm}$.

(II) We have

$$\begin{aligned} \max \{ \|\mathbb{E} [Z^\top Z]\|, \|\mathbb{E} [ZZ^\top]\| \} &= \|\mathbb{E} [Z^2]\| \\ &\lesssim \mathbb{E} |\sigma(\mathbf{w}^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}^\top \mathbf{x}_i)|^2 \|\mathbf{w} \mathbf{w}^\top - I\|^2 \lesssim \frac{B^2 d^2}{m^2}. \end{aligned}$$

(III) For $\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}(\mathbf{a}^\top Z \mathbf{b})^2)^{1/2}$, it reaches the maximal when $\mathbf{a} = \mathbf{b}$ since Z is a symmetric matrix. Thus, we have

$$\mathbb{E}(\mathbf{a}^\top Z \mathbf{a})^2 = \mathbb{E} |\sigma(\mathbf{w}^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}^\top \mathbf{x}_i)|^2 (\mathbf{a}^\top (\mathbf{w} \mathbf{w}^\top - I) \mathbf{a})^2 \lesssim \frac{B^2 d^2}{m^2}.$$

Then $\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}(\mathbf{a}^\top Z \mathbf{b})^2)^{1/2} \lesssim \frac{Bd}{m}$.

Moreover, $\|\mathbb{E} Z_j\| \lesssim \frac{Bd}{m}$ by Eq. equation 16. Then by Theorem B.2, we have

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\| \leq \tilde{O}\left(\frac{Bd}{m}\right)$$

with probability $1 - \frac{\delta}{2B}$. □

Lemma C.8. *If the learning rate $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^2})$, we have*

$$\left\| \frac{1}{m} \sum_{j=1}^m r_i^{(1)} \left| \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I)(V, V, V) \right\| \leq \tilde{O}\left(\frac{B^{5/2}}{m}\right) \quad (17)$$

with probability $1 - \frac{\delta}{2B}$ for any i .

Proof. We define $\tilde{Z}_j = r_i^{(1)} |\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I)(V, V, V)$ and Z_j be the flatten of \tilde{Z}_j along the first dimension. Then we have

$$\left\| \frac{1}{m} \sum_{j=1}^m \tilde{Z}_j \right\| \leq \left\| \frac{1}{m} \sum_{j=1}^m Z_j \right\| \leq \|\mathbb{E} Z_j\| + \left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\|.$$

We first bound $\mathbb{E} Z_j$. By Lemma C.4 and Corollary C.6, we have

$$\|\mathbb{E} Z_j\| \lesssim \mathbb{E} |\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| \|\mathbf{V}^\top \mathbf{w}_j\|^3 \lesssim \frac{B^{5/2}}{m}. \quad (18)$$

Next we bound $\|\frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j\|$ with matrix Bernstein inequality. We first check the conditions of Theorem B.2 with Lemma C.4 and Corollary C.6:

(I) We first bound the norm of Z_j . We have

$$\|Z_j\| \lesssim 2|\sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)| \|V^\top \mathbf{w}_j\|^3 \leq \tilde{O}\left(\frac{B^{5/2}}{m}\right)$$

with probability $1 - \frac{\delta}{2Bm}$.

(II) We have

$$\begin{aligned} \max \{ \|\mathbb{E}[Z^\top Z]\|, \|\mathbb{E}[ZZ^\top]\| \} &\leq \mathbb{E}[\|Z\|^2] \\ &\lesssim \mathbb{E}|\sigma(\mathbf{w}^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}^\top \mathbf{x}_i)|^2 \|V^\top \mathbf{w}\|^6 \lesssim \frac{B^5}{m^2}. \end{aligned}$$

(III) We have

$$\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}(\mathbf{a}^\top Z \mathbf{b})^2)^{1/2} \leq \left(\mathbb{E}[\|Z\|^2] \right)^{1/2} \lesssim \frac{B^{5/2}}{m}.$$

Moreover, $\|\mathbb{E} Z_j\| \lesssim \frac{B^{5/2}}{m}$ by Eq. equation 18. Then by Theorem B.2, we have

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\| \leq \tilde{O}\left(\frac{B^{5/2}}{m}\right)$$

with probability $1 - \frac{\delta}{2B}$. □

Lemma C.9. *If the learning rate $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^2})$, and $m \geq \tilde{\Omega}(B)$, we have*

$$\left\| \frac{1}{m} \sum_{j=1}^m \left(r_i^{(1)} - r_i \right) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \leq \tilde{O}\left(\frac{Bd}{m}\right) \quad (19)$$

with probability $1 - \frac{\delta}{2B}$ for any i .

Proof. We define $Z_j = (r_i^{(1)} - r_i) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I)$, then we have

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j \right\| \leq \|\mathbb{E} Z_j\| + \left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\|.$$

We first bound $\|\mathbb{E} Z_j\|$. By Lemma C.5 we have

$$\begin{aligned} \|\mathbb{E} Z_j\| &\leq \mathbb{E} |r_i^{(1)} - r_i| \|\sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I)\| \\ &\leq \left(\mathbb{E} [r_i^{(1)} - r_i]^2 \right)^{1/2} (\mathbb{E} \|\sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I)\|^2)^{1/2} \\ &\leq \tilde{O}\left(\frac{Bd}{m}\right). \end{aligned} \quad (20)$$

Next we bound $\left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\|$ with matrix Bernstein inequality. We first check the conditions of Theorem B.2:

(I) We first bound the norm of Z_j . By Lemma C.5, we have

$$\|Z_j\| \leq |r_i^{(1)} - r_i| |\sigma(\mathbf{w}_j^\top \mathbf{x}_i)| (\|\mathbf{w}_j\|^2 + 1) \leq \tilde{O}\left(\frac{Bd}{m}\right)$$

with probability $1 - \frac{\delta}{2Bm}$.

(II) By Lemma C.5 we have

$$\begin{aligned} \max \{ \|\mathbb{E}[Z^\top Z]\|, \|\mathbb{E}[ZZ^\top]\| \} &= \|\mathbb{E}[Z^2]\| \\ &\leq \left(\mathbb{E}[|r_i^{(1)} - r_i|^4] \right)^{1/2} \left(\mathbb{E} \|\sigma(\mathbf{w}^\top \mathbf{x}_i)(\mathbf{w}\mathbf{w}^\top - I)\|^4 \right)^{1/2} \\ &\leq \tilde{O}\left(\frac{B^2 d^2}{m^2}\right). \end{aligned}$$

(III) For $\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}(\mathbf{a}^\top Z \mathbf{b})^2)^{1/2}$, it reaches the maximal when $\mathbf{a} = \mathbf{b}$ since Z is a symmetric matrix. Thus, we have

$$\begin{aligned} \mathbb{E}(\mathbf{a}^\top Z \mathbf{a})^2 &= \mathbb{E}|r_i^{(1)} - r_i|^2 (\mathbf{a}^\top \sigma(\mathbf{w}^\top \mathbf{x}_i)(\mathbf{w}\mathbf{w}^\top - I)\mathbf{a})^2 \\ &\leq \left(\mathbb{E}[|r_i^{(1)} - r_i|^4] \right)^{1/2} \left(\mathbb{E} \|\sigma(\mathbf{w}^\top \mathbf{x}_i)(\mathbf{w}\mathbf{w}^\top - I)\|^4 \right)^{1/2} \\ &\lesssim \frac{B^2 d^2}{m^2}. \end{aligned}$$

Then $\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}(\mathbf{a}^\top Z \mathbf{b})^2)^{1/2} \lesssim \frac{Bd}{m}$.

Moreover, $\|\mathbb{E} Z_j\| \lesssim \frac{Bd}{m}$ by Eq. equation 20. Then by Theorem B.2, we have

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\| \leq \tilde{O}\left(\frac{Bd}{m}\right)$$

with probability $1 - \frac{\delta}{2B}$. □

Lemma C.10. *If the learning rate $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^2})$, and $m \geq \tilde{\Omega}(B)$, we have*

$$\left\| \frac{1}{m} \sum_{j=1}^m \left(r_i^{(1)} - r_i \right) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I)(V, V, V) \right\| \leq \tilde{O}\left(\frac{B^{5/2}}{m}\right) \quad (21)$$

with probability $1 - \frac{\delta}{2B}$ for any i .

Proof. We define $\tilde{\mathbf{Z}}_j = (r_i^{(1)} - r_i) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I)(V, V, V)$ and Z_j be the flatten of $\tilde{\mathbf{Z}}_j$ along the first dimension. Then we have

$$\left\| \frac{1}{m} \sum_{j=1}^m \tilde{\mathbf{Z}}_j \right\| \leq \left\| \frac{1}{m} \sum_{j=1}^m Z_j \right\| \leq \|\mathbb{E} Z_j\| + \left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\|.$$

We first bound $\|\mathbb{E} Z_j\|$. By Lemma C.5 we have

$$\begin{aligned} \|\mathbb{E} Z_j\| &\lesssim \mathbb{E} |r_i^{(1)} - r_i| \|\sigma(\mathbf{w}_j^\top \mathbf{x}_i)\| \|V^\top \mathbf{w}_j\|^3 \\ &\leq \left(\mathbb{E}[|r_i^{(1)} - r_i|^2] \right)^{1/2} \left(\mathbb{E} \|\sigma(\mathbf{w}_j^\top \mathbf{x}_i)\|^2 \|V^\top \mathbf{w}_j\|^6 \right)^{1/2} \\ &\leq \tilde{O}\left(\frac{B^{5/2}}{m}\right). \end{aligned} \quad (22)$$

Next we bound $\left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\|$ with matrix Bernstein inequality. We first check the conditions of Theorem B.2:

(I) We first bound the norm of Z_j . By Lemma C.5, we have

$$\|Z_j\| \leq |r_i^{(1)} - r_i| \|\sigma(\mathbf{w}_j^\top \mathbf{x}_i)\| \|V^\top \mathbf{w}_j\|^3 \leq \tilde{O}\left(\frac{B^{5/2}}{m}\right)$$

with probability $1 - \frac{\delta}{2Bm}$.

(II) By Lemma C.5 we have

$$\begin{aligned} & \max \{ \|\mathbb{E}[Z^\top Z]\|, \|\mathbb{E}[ZZ^\top]\| \} \\ & \lesssim \left(\mathbb{E}[|r_i^{(1)} - r_i|^4] \right)^{1/2} \left(\mathbb{E}[\sigma(\mathbf{w}^\top \mathbf{x}_i)]^4 \|V^\top \mathbf{w}\|^{12} \right)^{1/2} \\ & \leq \tilde{O}\left(\frac{B^5}{m^2}\right). \end{aligned}$$

(III) We have

$$\begin{aligned} \max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}(\mathbf{a}^\top Z \mathbf{b})^2)^{1/2} & \lesssim \left(\mathbb{E}(r_i^{(1)} - r_i)^2 [\sigma(\mathbf{w}^\top \mathbf{x}_i)]^2 \|V^\top \mathbf{w}\|^6 \right)^{1/2} \\ & \leq \left(\mathbb{E}[|r_i^{(1)} - r_i|^4] \right)^{1/2} \left(\mathbb{E}[\sigma(\mathbf{w}^\top \mathbf{x}_i)]^4 \|V^\top \mathbf{w}\|^{12} \right)^{1/2} \\ & \lesssim \frac{B^{5/2}}{m}. \end{aligned}$$

Moreover, $\|\mathbb{E} Z_j\| \lesssim \frac{B^{5/2}}{m}$ by Eq. equation 22. Then by Theorem B.2, we have

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j - \mathbb{E} Z_j \right\| \leq \tilde{O}\left(\frac{B^{5/2}}{m}\right)$$

with probability $1 - \frac{\delta}{2B}$. □

Now we are ready to prove Proposition C.1 and Proposition C.2.

Proof of Proposition C.1. We define $g_j^{(0)} = \frac{1}{m} \sum_{i=1}^B r_i^{(0)} \sigma(\mathbf{w}_j^{(0)} \cdot \mathbf{x}_i)$, $g_j^{(1)} = \frac{1}{m} \sum_{i=1}^B r_i^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i)$ and $\tilde{g}_j = g_j^{(0)} + g_j^{(1)}$, then the target vector $\tilde{g} = \sum_{j=1}^m \tilde{g}_j$. We also define $\hat{P}_0 = \sum_{j=1}^m g_j^{(0)} (\mathbf{w}_j \mathbf{w}_j^\top - I)$, $\hat{P}_1 = \sum_{j=1}^m g_j^{(1)} (\mathbf{w}_j \mathbf{w}_j^\top - I)$, $\hat{T}_0 = \sum_{j=1}^m g_j^{(0)} (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$, $\hat{T}_1 = \sum_{j=1}^m g_j^{(1)} (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$, $\hat{P} = \hat{P}_0 + \hat{P}_1$ and $\hat{T} = \hat{T}_0 + \hat{T}_1$. Let

$$P = \mathbb{E} \left[\sum_{i=1}^B r_i^* \sigma''(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i \mathbf{x}_i^\top \right]$$

and

$$\mathbf{T} = \mathbb{E} \left[\sum_{i=1}^B r_i^* \sigma^{(3)}(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i^{\otimes 3} \right].$$

We will bound $\|\hat{P}_0 + \hat{P}_1 - 2P\|$ and $\|(\hat{T}_0 + \hat{T}_1 - 2\mathbf{T})(V, V, V)\|$.

Error bound of P . We have

$$\begin{aligned} \|\hat{P}_0 + \hat{P}_1 - 2P\| & \leq 2\|\hat{P}_0 - P\| + \|\hat{P}_1 - \hat{P}_0\| \\ & \leq \tilde{O}(B\sqrt{\frac{d}{m}}) + \|\hat{P}_1 - \hat{P}_0\| \end{aligned}$$

with probability $1 - \frac{\delta}{2}$, where the last inequality is by Lemma B.4. Now we only need to bound $\|\hat{P}_1 - \hat{P}_0\|$. By Lemma C.7 and Lemma C.9,

$$\|\hat{P}_1 - \hat{P}_0\| = \left\| \frac{1}{m} \sum_{i=1}^B \sum_{j=1}^m r_i^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) - \frac{1}{m} \sum_{i=1}^B \sum_{j=1}^m r_i \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\|$$

$$\begin{aligned}
 &\leq \sum_{i=1}^B \left\| \frac{1}{m} \sum_{j=1}^m r_i^{(1)} \left| \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \\
 &\quad + \sum_{i=1}^B \left\| \frac{1}{m} \sum_{j=1}^m (r_i^{(1)} - r_i) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \\
 &\leq \tilde{O}\left(\frac{B^2 d}{m}\right) + \tilde{O}\left(\frac{B^2 d}{m}\right)
 \end{aligned}$$

with probability $1 - \frac{\delta}{2}$. Since $m \geq \tilde{\Omega}(B^2 d)$, $\|\hat{P}_1 - \hat{P}_0\| \leq O(B\sqrt{\frac{d}{m}})$. Thus,

$$\|\hat{P} - 2P\| = \|\hat{P}_0 + \hat{P}_1 - 2P\| \leq \tilde{O}\left(B\sqrt{\frac{d}{m}}\right)$$

with probability $1 - \delta$.

Error bound of $\mathbf{T}(V, V, V)$. We have

$$\begin{aligned}
 \|(\hat{\mathbf{T}}_0 + \hat{\mathbf{T}}_1 - 2\mathbf{T})(V, V, V)\| &\leq 2\|\hat{\mathbf{T}}_0(V, V, V) - \mathbf{T}(V, V, V)\| + \|\hat{\mathbf{T}}_1(V, V, V) - \hat{\mathbf{T}}_0(V, V, V)\| \\
 &\leq \tilde{O}\left(\frac{B^{5/2}}{m}\right) + \|\hat{\mathbf{T}}_1(V, V, V) - \hat{\mathbf{T}}_0(V, V, V)\|
 \end{aligned}$$

with probability $1 - \frac{\delta}{2}$, where the last inequality is by Lemma B.5. Now we only need to bound $\|\hat{\mathbf{T}}(V, V, V)_1 - \hat{\mathbf{T}}_0(V, V, V)\|$. By Lemma C.8 and Lemma C.10,

$$\begin{aligned}
 &\|\hat{\mathbf{T}}_1(V, V, V) - \hat{\mathbf{T}}_0(V, V, V)\| \\
 &= \left\| \frac{1}{m} \sum_{i=1}^B \sum_{j=1}^m r_i^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right. \\
 &\quad \left. - \frac{1}{m} \sum_{i=1}^B \sum_{j=1}^m r_i \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right\| \\
 &\leq \sum_{i=1}^B \left\| \frac{1}{m} \sum_{j=1}^m r_i^{(1)} \left| \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right\| \\
 &\quad + \sum_{i=1}^B \left\| \frac{1}{m} \sum_{j=1}^m (r_i^{(1)} - r_i) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right\| \\
 &\leq \tilde{O}\left(\frac{B^{7/2}}{m}\right) + \tilde{O}\left(\frac{B^{7/2}}{m}\right)
 \end{aligned}$$

with probability $1 - \frac{\delta}{2}$. Since $m \geq \tilde{\Omega}(B^2 d)$, $\|\hat{\mathbf{T}}_1 - \hat{\mathbf{T}}_0\| \leq O\left(\frac{B^{5/2}}{m}\right)$. Thus,

$$\|\hat{\mathbf{T}} - 2\eta\mathbf{T}\| = \eta\|\hat{\mathbf{T}}_0 + \hat{\mathbf{T}}_1 - 2\mathbf{T}\| \leq \eta\tilde{O}\left(\frac{B^{5/2}}{m}\right)$$

with probability $1 - \delta$.

We have that P 's smallest component $\nu_{\min} \geq |\nu|$, \mathbf{T} 's smallest component $\lambda_{\min} \geq |\nu|$ and $\kappa = \frac{\max |r_i^*|}{\min |r_i^*|} = 1$. Since $0 < |y_i| \leq 1$, then $|r_i^*|$ is lower bounded. By Theorem B.1,

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \leq \frac{1}{\min\{|\nu|, |\lambda|\} \pi_{\min}^2} \tilde{O}\left(B\sqrt{\frac{d}{m}}\right)$$

for all $i = 1, \dots, B$ with high probability. \square

Proof of Proposition C.2. We define $g_j^{(0)} = \frac{1}{m} \sum_{i=1}^B r_i^{(0)} \sigma(\mathbf{w}_j^{(0)} \cdot \mathbf{x}_i)$, $g_j^{(1)} = \frac{1}{m} \sum_{i=B+1}^N r_i^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i)$, $\tilde{g}_j^{(1)} = \frac{1}{m} \sum_{i=B+1}^N r_i^{(0)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i)$ and $\tilde{g}_j = g_j^{(0)} + \tilde{g}_j^{(1)}$, then the target vector $\tilde{g} = \sum_{j=1}^m \tilde{g}_j$. We also define $\hat{P}_0 = \sum_{j=1}^m g_j^{(0)} (\mathbf{w}_j \mathbf{w}_j^\top - I)$, $\hat{P}_1 = \sum_{j=1}^m g_j^{(1)} (\mathbf{w}_j \mathbf{w}_j^\top - I)$, $\tilde{P}_1 = \sum_{j=1}^m \tilde{g}_j^{(1)} (\mathbf{w}_j \mathbf{w}_j^\top - I)$, $\hat{\mathbf{T}}_0 = \sum_{j=1}^m g_j^{(0)} (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$, $\hat{\mathbf{T}}_1 = \sum_{j=1}^m g_j^{(1)} (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$, $\tilde{\mathbf{T}}_1 = \sum_{j=1}^m \tilde{g}_j^{(1)} (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$, $\hat{P} = \hat{P}_0 + \hat{P}_1$ and $\hat{\mathbf{T}} = \hat{\mathbf{T}}_0 + \hat{\mathbf{T}}_1$. Let

$$P = \mathbb{E} \left[\sum_{i=1}^N r_i^* \sigma''(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i \mathbf{x}_i^\top \right]$$

and

$$\mathbf{T} = \mathbb{E} \left[\sum_{i=1}^N r_i^* \sigma^{(3)}(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i^{\otimes 3} \right].$$

We will bound $\|\hat{P}_0 + \hat{P}_1 - 2P\|$ and $\|(\hat{\mathbf{T}}_0 + \hat{\mathbf{T}}_1 - 2\mathbf{T})(V, V, V)\|$.

Error bound of P . We have

$$\begin{aligned} \|\hat{P}_0 + \hat{P}_1 - P\| &\leq \|\hat{P}_0 + \tilde{P}_1 - P\| + \|\hat{P}_1 - \tilde{P}_1\| \\ &\leq \tilde{O}(N\sqrt{\frac{d}{m}}) + \|\hat{P}_1 - \tilde{P}_1\| \end{aligned}$$

with probability $1 - \frac{\delta}{2}$, where the last inequality is by Lemma B.4. Now we only need to bound $\|\hat{P}_1 - \tilde{P}_1\|$. By Lemma C.7 and Lemma C.9,

$$\begin{aligned} \|\hat{P}_1 - \tilde{P}_1\| &= \left\| \frac{1}{m} \sum_{i=B+1}^N \sum_{j=1}^m r_i^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) \right. \\ &\quad \left. - \frac{1}{m} \sum_{i=B+1}^N \sum_{j=1}^m r_i \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \\ &\leq \sum_{i=B+1}^N \left\| \frac{1}{m} \sum_{j=1}^m r_i^{(1)} \left| \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \\ &\quad + \sum_{i=B+1}^N \left\| \frac{1}{m} \sum_{j=1}^m (r_i^{(1)} - r_i) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j \mathbf{w}_j^\top - I) \right\| \\ &\leq \tilde{O}\left(\frac{(N-B)^2 d}{m}\right) + \tilde{O}\left(\frac{(N-B)^2 d}{m}\right) \end{aligned}$$

with probability $1 - \frac{\delta}{2}$. Since $m \geq \tilde{\Omega}(B^2 d)$, $\|\hat{P}_1 - \tilde{P}_1\| \leq O(N\sqrt{\frac{d}{m}})$. Thus,

$$\|\hat{P} - P\| = \|\hat{P}_0 + \hat{P}_1 - P\| \leq \tilde{O}(N\sqrt{\frac{d}{m}})$$

with probability $1 - \delta$.

Error bound of $\mathbf{T}(V, V, V)$. We have

$$\begin{aligned} \|(\hat{\mathbf{T}}_0 + \hat{\mathbf{T}}_1 - \mathbf{T})(V, V, V)\| &\leq \|(\hat{\mathbf{T}}_0 + \tilde{\mathbf{T}} - \mathbf{T})(V, V, V)\| + \|\hat{\mathbf{T}}_1(V, V, V) - \tilde{\mathbf{T}}(V, V, V)\| \\ &\leq \tilde{O}\left(\frac{N^{5/2}}{m}\right) + \|\hat{\mathbf{T}}_1(V, V, V) - \tilde{\mathbf{T}}(V, V, V)\| \end{aligned}$$

with probability $1 - \frac{\delta}{2}$, where the last inequality is by Lemma B.5. Now we only need to bound $\|\hat{\mathbf{T}}(V, V, V)_1 - \tilde{\mathbf{T}}(V, V, V)\|$. By Lemma C.8 and Lemma C.10,

$$\|\hat{\mathbf{T}}_1(V, V, V) - \tilde{\mathbf{T}}(V, V, V)\|$$

$$\begin{aligned}
 &= \left\| \frac{1}{m} \sum_{i=B+1}^N \sum_{j=1}^m r_i^{(1)} \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right. \\
 &\quad \left. - \frac{1}{m} \sum_{i=B+1}^N \sum_{j=1}^m r_i \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right\| \\
 &\leq \sum_{i=B+1}^N \left\| \frac{1}{m} \sum_{j=1}^m r_i^{(1)} \left| \sigma(\mathbf{w}_j^{(1)} \cdot \mathbf{x}_i) - \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \right| (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right\| \\
 &\quad + \sum_{i=B+1}^N \left\| \frac{1}{m} \sum_{j=1}^m (r_i^{(1)} - r_i) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I) (V, V, V) \right\| \\
 &\leq \tilde{O}\left(\frac{(N-B)^{7/2}}{m}\right) + \tilde{O}\left(\frac{(N-B)^{7/2}}{m}\right)
 \end{aligned}$$

with probability $1 - \frac{\delta}{2}$. Since $m \geq \tilde{\Omega}(B^2 d)$, $\|\hat{\mathbf{T}}_1 - \hat{\mathbf{T}}_0\| \leq O(\frac{N^{5/2}}{m})$. Thus,

$$\|\hat{\mathbf{T}} - 2\mathbf{T}\| = \|\hat{\mathbf{T}}_0 + \hat{\mathbf{T}}_1 - 2\mathbf{T}\| \leq \tilde{O}\left(\frac{N^{5/2}}{m}\right)$$

with probability $1 - \delta$.

We have that P 's smallest component $\nu_{\min} \geq |\nu|$, \mathbf{T} 's smallest component $\lambda_{\min} \geq |\nu|$ and $\kappa = \frac{\max |r_i^*|}{\min |r_i^*|} = 1$. Since $0 < |y_i| \leq 1$, then $|r_i^*|$ is lower bounded. By Theorem B.1,

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \leq \frac{1}{\min\{|\nu|, |\lambda|\} \pi_{\min}^2} \tilde{O}(N \sqrt{\frac{d}{m}})$$

for all $i = 1, \dots, N$ with high probability. \square

C.2 Differential Privacy

In differential private federated learning, the gradient update for any parameter is

$$\tilde{G} = G / \max \left\{ 1, \frac{\|G\|}{C} \right\} + \mathcal{E}_G,$$

where $\|G\|$ is the norm of the gradient of all parameters and $\mathcal{E}_g \sim \mathcal{N}(0, \sigma_0^2 C^2 I_d)$, d is the dimension of parameter. First, we give the proof of the case with no random noise that only gradient clipping is used, i.e. $\tilde{G} = G / \max \left\{ 1, \frac{\|G\|}{C} \right\}$. The error bound is the same as the case with no defense.

Proposition C.11. *Under Assumption B.1 and Assumption B.2, $y_i \in \{\pm 1\}$, we observe clipped gradient descent steps \tilde{G} with batch size B , where $B^4 \leq \tilde{O}(d)$ and clipping threshold C . If $m \geq \tilde{\Omega}(\frac{B^2 d}{\nu^2 \pi_{\min}^2})$, then with appropriate tensor decomposition methods and proper weights, we can reconstruct input data with the error bound*

$$\sqrt{\frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2} \leq \frac{1}{\min\{|\nu|, |\lambda|\} \pi_{\min}^2} \tilde{O}(B \sqrt{\frac{d}{m}})$$

with high probability.

Proof. For any j , the observed gradient updates for a_j are $\tilde{g}(\mathbf{w}_j) = g(\mathbf{w}_j) / \max \left\{ 1, \frac{\|G\|}{C} \right\}$. We define $\tilde{P} = \frac{1}{m} \sum_{j=1}^m \tilde{g}(\mathbf{w}_j) (\mathbf{w}_j \mathbf{w}_j^\top - I)$ and $\tilde{\mathbf{T}} = \frac{1}{m} \sum_{j=1}^m \tilde{g}(\mathbf{w}_j) (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \tilde{\otimes} I)$. \hat{P} , $\hat{\mathbf{T}}$, P and \mathbf{T} are defined in Eq. (5) to (8). Then let $R = \min\{1, \frac{C}{\|G\|}\}$, we have $\tilde{P} = R\hat{P}$ and $\tilde{\mathbf{T}} = R\hat{\mathbf{T}}$. Then by Lemma B.4 and B.5 we have

$$\|\tilde{P} - RP\| = R\|\hat{P} - P\| \leq \tilde{O}\left(\frac{RB\sqrt{d}}{\sqrt{m}}\right)$$

and

$$\|\tilde{\mathbf{T}}(V, V, V) - R\mathbf{T}(V, V, V)\| = R\|\hat{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\| \leq \tilde{O}\left(\frac{RB^{5/2}}{\sqrt{m}}\right)$$

with high probability.

On the other hand, the smallest components of RP and $R\mathbf{T}$ are $R\nu_{\min}$ and $R\lambda_{\min}$ respectively. We have $\nu_{\min} \geq |\nu|$, $\lambda_{\min} \geq |\lambda|$ and $\kappa = 1$. Then by Theorem B.1,

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \leq \tilde{O}\left(\frac{RB\sqrt{d/m}}{R|\mu|\pi_{\min}^2}\right) + \tilde{O}\left(\frac{R\sqrt{B^6/m}}{R|\lambda|\pi_{\min}^2}\right) \leq \frac{1}{\min\{|\nu|, |\lambda|\}\pi_{\min}^2} \tilde{O}\left(B\sqrt{\frac{d}{m}}\right)$$

for all $i = 1, \dots, B$ with high probability. \square

Then we give the formal statement and proof of the error bound of the case without gradient clipping that the only defense is random noise, i.e. $\tilde{G} = G + \mathcal{E}_G$, where $\mathcal{E}_G \sim \mathcal{N}(0, \sigma_0^2 I)$.

Proposition C.12. *Under Assumption B.1 and Assumption B.2, $y_i \in \{\pm 1\}$, we observe noisy gradient descent steps $G + \epsilon_0$ with batch size B , where $K^2 B^4 \leq \tilde{O}(d)$ and $\epsilon \sim \mathcal{N}(0, \sigma_0^2 I)$. If $m \geq \tilde{\Omega}(\frac{B^2 d}{\nu^2 \pi_{\min}^2})$, then with appropriate tensor decomposition methods and proper weights, we can reconstruct input data with the error bound*

$$\sqrt{\frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2} \leq \frac{1}{\min\{|\nu|, |\lambda|\}\pi_{\min}^2} \tilde{O}((B + \sigma_0)\sqrt{\frac{d}{m}})$$

with high probability.

Proof. For any j , the observed gradient updates for a_j are $\tilde{g}(\mathbf{w}_j) = g(\mathbf{w}_j) + \epsilon_j$, where $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$. We define $\tilde{P} = \frac{1}{m} \sum_{j=1}^m \tilde{g}(\mathbf{w}_j)(\mathbf{w}_j \mathbf{w}_j^\top - I)$ and $\tilde{\mathbf{T}} = \frac{1}{m} \sum_{j=1}^m \tilde{g}(\mathbf{w}_j)(\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$. \hat{P} , $\hat{\mathbf{T}}$, P and \mathbf{T} are defined in Eq. (5) to (8). Then we have $\tilde{P} = \hat{P} + \frac{1}{m} \sum_{j=1}^m \epsilon_j(\mathbf{w}_j \mathbf{w}_j^\top - I)$ and $\tilde{\mathbf{T}} = \hat{\mathbf{T}} + \frac{1}{m} \sum_{j=1}^m \epsilon_j(\mathbf{w}_j^{\otimes 3} + \mathbf{w}_j \otimes I)$. Now we bound $\|\tilde{P} - P\|$ and $\|\tilde{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\|$.

Error bound of \tilde{P} . We have $\|\tilde{P} - P\| \leq \|\hat{P} - P\| + \|P_\epsilon\|$, where $P_\epsilon = \frac{1}{m} \sum_{j=1}^m \epsilon_j(\mathbf{w}_j \mathbf{w}_j^\top - I)$. Since $\|\hat{P} - P\| \leq \tilde{O}(\frac{B\sqrt{d}}{\sqrt{m}})$ with high probability by Lemma B.4, we only have to bound $\|P_\epsilon\|$. We let $Z_j = (\epsilon_{1j} + \epsilon_{2j})(\mathbf{w}_j \mathbf{w}_j^\top - I)$ and check the conditions of Theorem B.2:

(I) We first bound the norm of Z_j :

$$\|Z_j\| \leq |\epsilon_j| \|(\mathbf{w}_j \mathbf{w}_j^\top - I)\| \lesssim \sigma_0 d (\log(16m/\delta))^2$$

with probability $1 - \frac{\delta}{4m}$.

(II) We have

$$\max\{\|\mathbb{E}[Z^\top Z]\|, \|\mathbb{E}[ZZ^\top]\|\} = \|\mathbb{E}\epsilon^2 \mathbb{E}[(\mathbf{w}\mathbf{w}^\top - I)^2]\|.$$

Let $Q = (\mathbf{w}\mathbf{w}^\top - I)^2$. Since $Q_{ij} = \sum_{k \neq i, j} w_i w_j w_k^2 + (w_i^2 + w_j^2 - 2)w_i w_j$ for $i \neq j$ and $Q_{ii} = \sum_{k \neq i} w_i^2 w_k^2 + (w_i^2 - 1)^2$, $\mathbb{E}(\mathbf{w}\mathbf{w}^\top - I)^2 = (d+1)I$. Then $\max\{\|\mathbb{E}[Z^\top Z]\|, \|\mathbb{E}[ZZ^\top]\|\} \leq O(\sigma_0^2 d)$.

(III) For $\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}[(\mathbf{a}^\top Z \mathbf{b})^2])^{1/2}$, it reaches the maximal when $\mathbf{a} = \mathbf{b}$ since Z is a symmetric matrix.

Thus, we have

$$\begin{aligned}
 \mathbb{E}(\mathbf{a}^\top (\mathbf{w}\mathbf{w}^\top - I)Z\mathbf{a})^2 &= \mathbb{E} \left(\sum_{i=1}^d a_i^2 (w_i^2 - 1) + \sum_{i \neq j} a_i a_j w_i w_j \right)^2 \\
 &= \mathbb{E} \left[\sum_{i=1}^d a_i^4 (w_i^2 - 1)^2 + \sum_{i \neq j} a_i^2 a_j^2 w_i^2 w_j^2 \right] \\
 &= 2 \sum_{i=1}^d a_i^4 + \sum_{i \neq j} a_i^2 a_j^2 \\
 &\leq 2 \left(\sum_{i=1}^d a_i^2 \right)^2 = 2.
 \end{aligned} \tag{23}$$

Then, $\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} (\mathbb{E}[(\mathbf{a}^\top Z\mathbf{b})^2])^{1/2} \leq O(\sigma_0^2)$.

Moreover, $\mathbb{E}[Z] = 0$. Then by Theorem B.2,

$$\left\| \frac{1}{m} \sum_{j=1}^m Z_j \right\| \leq \log(16m/\delta) \sqrt{\frac{\sigma_0^2 d \log(4/\delta)}{m}} \leq \tilde{O}(\sigma_0 \sqrt{\frac{d}{m}})$$

with probability $1 - \frac{\delta}{2}$. Thus,

$$\|\tilde{P} - P\| \leq \tilde{O}((B + \sigma_0) \sqrt{\frac{d}{m}})$$

with probability $1 - \delta$.

Error bound of $\tilde{\mathbf{T}}(V, V, V)$. We have $\|\tilde{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\| \leq \|\hat{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\| + \|\mathbf{T}_\epsilon(V, V, V)\|$, where $\mathbf{T}_\epsilon = \frac{1}{m} \sum_{j=1}^m \epsilon_j (\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$. Since $\|\hat{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\| \leq \tilde{O}(\frac{B^{5/2}}{\sqrt{m}})$ with high probability by Lemma B.5, we only need to bound $\|\mathbf{T}_\epsilon(V, V, V)\|$. Note that $\|\mathbf{T}_\epsilon\| \leq \|T_\epsilon^{(1)}\|$, where $T_\epsilon^{(1)}$ is the flatten of \mathbf{T}_ϵ along the first dimension, so we can bound $\|T_\epsilon^{(1)}(V, V, V)\|$ instead. We check the conditions of Theorem B.2. For any \mathbf{x}

(I) We have

$$\|Z_j\| \lesssim |\epsilon_j| \|V^\top \mathbf{w}_j\|^3 \leq \tilde{O}(\frac{\sigma_0 B^{3/2}}{\sqrt{m}})$$

with probability $1 - \frac{\delta}{4m}$.

(II) We have

$$\max \{ \|\mathbb{E}[Z^\top Z]\|, \|\mathbb{E}[ZZ^\top]\| \} \leq \mathbb{E}[\|Z\|^2] \lesssim \mathbb{E}[\epsilon^2] \mathbb{E}[\|V^\top \mathbf{w}_j\|^6] \lesssim \sigma_0^2 B^3.$$

(III) We have

$$\max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} \left(\mathbb{E}[(\mathbf{a}^\top Z\mathbf{b})^2] \right)^{1/2} \leq \left(\mathbb{E}[\|Z\|^2] \right)^{1/2} \lesssim \sigma_0 B^{3/2}.$$

Moreover, $\mathbb{E}[Z] = 0$. Then by Theorem B.2, we have for any i that

$$\left\| \frac{1}{m} \sum_{j=1}^m \mathbf{T}_\epsilon^{(1)}(V, V, V) \right\| \leq \tilde{O}(\frac{\sigma_0 B^{3/2}}{\sqrt{m}})$$

with probability $1 - \frac{\delta}{2}$. Thus,

$$\|\tilde{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\| \leq \tilde{O}((B + \sigma_0) \sqrt{\frac{B^3}{m}})$$

with probability $1 - \delta$.

Since ν_{\min} , λ_{\min} and κ are not changed, by Theorem B.1,

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \leq \frac{K}{\min\{|\nu|, |\lambda|\} \pi_{\min}^2} \tilde{O}((B + \sigma_0) \sqrt{\frac{d}{m}})$$

for all $i = 1, \dots, B$ with high probability. \square

Then we consider the case where both gradient clipping and gradient noise are used in the training. In this case, we found that gradient clipping will increase the error caused by gradient noise though clipping itself has no effect on the error bound. Here we reparameterize the observed gradient as $\tilde{G} = G / \max\{1, \frac{\|G\|}{C}\} + \epsilon_0$, where $\epsilon_0 \sim \mathcal{N}(0, \sigma^2 I)$.

Proposition C.13 (Full version of Proposition 3.3). *Under Assumption B.1 and Assumption B.2, $y_i \in \{\pm 1\}$, we observe clipped noisy gradient descent steps $G / \max\{1, \frac{\|G\|}{C}\} + \epsilon$ with batch size B , where $B^4 \leq \tilde{O}(d)$ and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. If $m \geq \tilde{\Omega}(\frac{B^2 d}{\nu^2 \pi_{\min}^2})$, then with appropriate tensor decomposition methods and proper weights, we can reconstruct input data with the error bound*

$$\sqrt{\frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2} \leq \frac{1}{\min\{|\nu|, |\lambda|\} \pi_{\min}^2} \tilde{O}((B + \sigma_0 \max\{1, \frac{\|G\|}{C}\}) \sqrt{\frac{d}{m}})$$

with high probability.

Proof. For any j , the observed gradient updates for a_j are $\tilde{g}(\mathbf{w}_j) = g(\mathbf{w}_j) / \max\{1, \frac{\|g\|}{C}\} + \epsilon_j$, where $\epsilon_j \sim \mathcal{N}(0, \sigma_0^2)$. We define $\tilde{P} = \frac{1}{m} \sum_{j=1}^m \tilde{g}(\mathbf{w}_j)(\mathbf{w}_j \mathbf{w}_j^\top - I)$ and $\tilde{\mathbf{T}} = \frac{1}{m} \sum_{j=1}^m \tilde{g}(\mathbf{w}_j)(\mathbf{w}_j^{\otimes 3} - \mathbf{w}_j \otimes I)$. \hat{P} , $\hat{\mathbf{T}}$, P and \mathbf{T} are defined in Eq. (5) to (8). Let $R = \min\{1, \frac{C}{\|G\|}\}$. Then we have $\tilde{P} = R\hat{P} + \frac{1}{m} \sum_{j=1}^m \epsilon_j(\mathbf{w}_j \mathbf{w}_j^\top - I)$ and $\tilde{\mathbf{T}} = R\hat{\mathbf{T}} + \frac{1}{m} \sum_{j=1}^m \epsilon_j(\mathbf{w}_j^{\otimes 3} + \mathbf{w}_j \otimes I)$. By the proof of Proposition C.11 and Proposition C.12, we have

$$\|\tilde{P} - RP\| \leq R\|\hat{P} - P\| + \|P_\epsilon\| \leq \tilde{O}((RB + \sigma_0) \frac{d}{m}) \quad (24)$$

and

$$\begin{aligned} \|\tilde{\mathbf{T}}(V, V, V) - R\mathbf{T}(V, V, V)\| &\leq R\|\hat{\mathbf{T}}(V, V, V) - \mathbf{T}(V, V, V)\| + \|\mathbf{T}_\epsilon(V, V, V)\| \\ &\leq \tilde{O}((RB + \sigma_0) \sqrt{\frac{B^3}{m}}). \end{aligned} \quad (25)$$

Note that the smallest components of RP and $R\mathbf{T}$ are $R\nu_{\min}$ and $R\lambda_{\min}$ respectively. Then by Theorem B.1,

$$\begin{aligned} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| &\leq \tilde{O}\left(\frac{(RB + \sigma_0) \sqrt{d/m}}{R|\mu| \pi_{\min}^2 / K}\right) + \tilde{O}\left(\frac{(RB + \sigma_0) K \sqrt{B^4/m}}{R|\lambda| \pi_{\min}^2 / K}\right) \\ &\leq \frac{K}{\min\{|\nu|, |\lambda|\} \pi_{\min}^2} \tilde{O}((B + \sigma_0 \max\{1, \frac{\|G\|}{C}\}) \sqrt{\frac{d}{m}}) \end{aligned}$$

for all $i = 1, \dots, B$ with high probability. \square

D Analysis of Reconstruction Lower Bound

D.1 The Minimax Risk

We start with the following definition of an information-theoretical minimax risk:

Definition D.1. *The minimax risk with batch size n is defined as*

$$R_L = \left(\min_{\hat{S}=\hat{S}(O)} \max_{S \subset \mathcal{X}^B} \min_{\pi} \mathbb{E} \left[d(S, \pi(\hat{S})) \right] \right)^{1/2}. \quad (26)$$

Here $d(S, \pi(\hat{S})) = \frac{1}{B} \sum_{i=1}^B \|S_i - \hat{S}_{\pi(i)}\|^2$, where π is a permutation of $[B]$. $\hat{S} = \hat{S}(O)$ is ranged over all algorithms represented as functions of O , $O = O(S)$ is a random variable related with S , S is ranged over all possible input data and $\pi \in S_n$ is ranged over all permutations with rank n .

The risk defined above provides a lower bound for the minimax expected risk of all attacking algorithms. As an information-theoretical or statistical minimax risk, we model the observation with a random noise. In our following analysis, O is interpreted as model gradients with a random Gaussian noise $O(S) = \nabla L(S; \Theta) + \epsilon$. The expectation in 26 is taken over the random noise ϵ .

D.2 Analysis of 2-Layer Neural Networks

In our setup of data reconstruction, we consider reconstructing training data based on the model gradients. Then for a 2-layer neural network $f(\mathbf{x}; \Theta)$ with data and labels $\{(\mathbf{x}_i, y_i)\}_{i=1}^B$ and square loss function ℓ , $S = \{\mathbf{x}_1, \dots, \mathbf{x}_B\}$ is the input data and $O(S) = \sum_{i=1}^B \nabla_{\Theta} \ell(f(\mathbf{x}_i), y_i)$ is the gradient. We further assume we already have information about the learning rate and the response variables.

Since we view data reconstruction as a statistical problem to estimate S from $O(S)$, the expectation of $d(S, \hat{S})$ can be lower bounded by Cramer-Rao lower bound (Cramér, 1999; Rao, 1992). In this problem, we define the Jacobian of the observation O by $J(S) = \nabla_S O(S)$. Then the minimax risk

$$R_L^2 \geq \text{tr}((J(S)J(S)^\top)^{-1})\sigma^2. \quad (27)$$

To analyze the lower bound of minimax risk in data reconstruction, we first introduce some lemmas. For ease of calculation, we loosen the minimax risk as below:

Lemma D.2. For positive-definite symmetric matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$, we have $\text{tr}(\mathbf{M}^{-1}) \geq \frac{d^2}{\text{tr}(\mathbf{M})}$

Proof. Let $\lambda_1, \dots, \lambda_d$ be the eigenvalues of \mathbf{M} . Since \mathbf{M} is positive definite, the eigenvalues are real and positive. Notice that $\text{tr}(\mathbf{M}) = \sum_{i=1}^d \lambda_i$ and $\text{tr}(\mathbf{M}^{-1}) = \sum_{i=1}^d \frac{1}{\lambda_i}$, by Cauchy's inequality we have

$$\text{tr}(\mathbf{M})\text{tr}(\mathbf{M}^{-1}) = \sum_{i=1}^d \lambda_i \sum_{i=1}^d \frac{1}{\lambda_i} \leq d^2.$$

Rearranging the inequality yields the desired result. \square

Remark D.3. Applying the lower bound to R_L in Eq. (27), the bound is now

$$R_L^2 \geq \frac{d^2}{\text{tr}(J(S)J(S)^\top)}\sigma^2. \quad (28)$$

To analyze 2-layer neural networks, we make some assumptions about the model and how the model parameters are generated for analyzing scaling.

Assumption D.1. Let σ be an activation function $\mathbb{R} \rightarrow \mathbb{R}$ such that $|\sigma'(x)| < C$ and $|\sigma''(x)| < C$ for any x for some constant C . Let the 2-layer network $f(\mathbf{x})$ be defined as $f(\mathbf{x}) = \sum_{j=1}^m a_j \sigma(\mathbf{w}_j^\top \mathbf{x})$, where $a_j \in \mathbb{R}$ are chosen from $\mathcal{N}(0, \frac{1}{m^2})$ and $\mathbf{w}_j \in \mathbb{R}^d$ are chosen from $\mathcal{N}(0, I_d)$. Let data points (\mathbf{x}_i, y_i) satisfy $\|\mathbf{x}_i\| = 1$ and $|y_i| \in \{\pm 1\}$.

Remark D.4. Popular activation functions including ReLU, Sigmoid, LeakyReLU and Tanh all satisfy the assumptions about the activation function.

Lemma D.5. With high probability we have the following scaling for all $i = 1, \dots, B$ and $j = 1, \dots, m$:

- $a_j = \tilde{O}(\frac{1}{m})$
- $\|\mathbf{w}_j\| = \tilde{O}(\sqrt{d})$
- $|\mathbf{x}_i^\top \mathbf{w}_j| = \tilde{O}(1)$
- $f(\mathbf{x}_i) = \tilde{O}(1)$.

Here the log terms are omitted.

The proof is trivial with concentration bounds.

We first prove the result using batch size 1 and then generalize to batch size higher than 1.

Lemma D.6. *Under Assumption D.1 and with high probability, the lower bound of minimax risk R_L with the observation*

$$O(S) = \nabla_\theta(y - f(\mathbf{x}))^2 + \epsilon$$

is of the scale $R_L \geq \tilde{\Omega}(\sigma\sqrt{\frac{d}{m}})$, where ϵ follows $\mathcal{N}(0, \sigma^2 \mathbf{I}_{md+m})$.

Proof. We first calculate the model gradients:

$$\begin{cases} \nabla_{a_j}(y - f(\mathbf{x}))^2 = 2(y - f(\mathbf{x}))\sigma(\mathbf{w}_j^\top \mathbf{x}) \\ \nabla_{\mathbf{w}_j}(y - f(\mathbf{x}))^2 = 2(y - f(\mathbf{x}))a_j\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{x}. \end{cases}$$

We then calculate the Jacobian of the gradients on x :

$$\begin{cases} \nabla_{\mathbf{x}} \nabla_{a_j}(y - f(\mathbf{x}))^2 = 2(y - f(\mathbf{x}))\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j - 2\sigma(\mathbf{w}_j^\top \mathbf{x})\mathbf{h} \\ \nabla_{\mathbf{x}} \nabla_{\mathbf{w}_j}(y - f(\mathbf{x}))^2 = 2(y - f(\mathbf{x}))a_j(\sigma''(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j\mathbf{x}^\top + \sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{I}) - 2a_j\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{h}\mathbf{x}^\top, \end{cases}$$

where $\mathbf{h} = \nabla_{\mathbf{x}} f(\mathbf{x}) = \sum_{j=1}^m (a_j\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j)$. Since $a_j = \tilde{O}(\frac{1}{m})$ with high probability and $\mathbf{w}_j = \tilde{O}(\sqrt{d})$ with high probability, we have with high probability that

$$\|\mathbf{h}\| \leq C \sum_{j=1}^m a_j \|\mathbf{w}_j\| = \tilde{O}(\sqrt{d}).$$

Denote $J(\mathbf{x})$ as the Jacobian of model gradients on the input data. Notice that $y - f(\mathbf{x}) = \tilde{O}(1)$, to calculate the lower bound, we first calculate $\text{tr}(J(\mathbf{x})^\top J(\mathbf{x}))$:

$$\begin{aligned} \text{tr}(J(\mathbf{x})^\top J(\mathbf{x})) &= \sum_{j=1}^m \|\nabla_{\mathbf{x}} \nabla_{a_j}(y - f(\mathbf{x}))^2\|^2 + \sum_{j=1}^m \|\nabla_{\mathbf{x}} \nabla_{\mathbf{w}_j}(y - f(\mathbf{x}))^2\|_F^2 \\ &\leq 8(y - f(\mathbf{x}))^2 \sum_{j=1}^m \left[\|\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j\|^2 + \|a_j(\sigma''(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j\mathbf{x}^\top + \sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{I})\|_F^2 \right] \\ &\quad + 8 \sum_{j=1}^m \left[\sigma^2(\mathbf{w}_j^\top \mathbf{x}) \|\mathbf{h}\|^2 + a_j^2 \sigma'^2(\mathbf{w}_j^\top \mathbf{x}) \|\mathbf{h}\|^2 \|\mathbf{x}\|^2 \right] \end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm. Note that the inequality used the fact that $\|A + B\| \leq 2(\|A\| + \|B\|)$ for any A and B .

For the first part, we have with high probability that

$$\begin{aligned} A_j &:= \|\sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j\|^2 + \|a_j(\sigma''(\mathbf{w}_j^\top \mathbf{x})\mathbf{w}_j\mathbf{x}^\top + \sigma'(\mathbf{w}_j^\top \mathbf{x})\mathbf{I})\|_F^2 \\ &= \sigma'^2(\mathbf{w}_j^\top \mathbf{x}) \|\mathbf{w}_j\|^2 + a_j^2 \sigma''^2(\mathbf{w}_j^\top \mathbf{x}) \|\mathbf{w}_j\|^2 \|\mathbf{x}\|^2 \\ &\quad + 2a_j^2 \sigma'(\mathbf{w}_j^\top \mathbf{x}) \sigma''(\mathbf{w}_j^\top \mathbf{x}) \mathbf{w}_j^\top \mathbf{x} + da_j^2 \sigma'^2(\mathbf{w}_j^\top \mathbf{x}) \\ &\leq C^2 \left[\|\mathbf{w}_j\|^2 + a_j^2 \|\mathbf{w}_j\|^2 + 2a_j^2 |\mathbf{w}_j^\top \mathbf{x}| + da_j^2 \right] \\ &= C^2 \left[\tilde{O}(d) + \tilde{O}(\frac{1}{m^2})\tilde{O}(d) + 2\tilde{O}(\frac{1}{m^2}) + d\tilde{O}(\frac{1}{m^2}) \right] \\ &= \tilde{O}(d). \end{aligned}$$

For the second part, we have with high probability that

$$B_j := \sigma^2(\mathbf{w}_j^\top \mathbf{x}) \|\mathbf{h}\|^2 + a_j^2 \sigma'^2(\mathbf{w}_j^\top \mathbf{x}) \|\mathbf{h}\|^2 \|\mathbf{x}\|^2$$

$$\begin{aligned} &\leq C^2 \tilde{O}(d) + C^2 \tilde{O}\left(\frac{1}{m^2}\right) \tilde{O}(d) \\ &= \tilde{O}(d). \end{aligned}$$

Therefore with a high probability,

$$\begin{aligned} \text{tr}(J(\mathbf{x})^\top J(\mathbf{x})) &\leq 8(y - f(\mathbf{x}))^2 \sum_{j=1}^m A_j + 8 \sum_{j=1}^m B_j \\ &= 8O(1) \sum_{j=1}^m O(d) \\ &= O(md) \end{aligned}$$

By Remark D.3, we have $R_L^2 \geq \frac{d^2}{\text{tr}(J(\mathbf{x})^\top J(\mathbf{x}))} \sigma^2 = \tilde{\Omega}\left(\frac{d}{m}\right) \sigma^2$. Then $R_L \geq \tilde{\Omega}\left(\sigma \sqrt{\frac{d}{m}}\right)$. \square

We now analyze the minimax risk of gradients summed over batch size B : $\mathbf{g}(\mathbf{x}_{1:B}) = \sum_{i=1}^B \mathbf{g}(\mathbf{x}_i)$.

Theorem D.7 (Full version of Theorem 3.2). *Under Assumption D.1 and with high probability, the lower bound of minimax risk R_L with the observation with B input samples*

$$O(S) = \sum_{i=1}^B \nabla_\theta (y - f(\mathbf{x}_i))^2 + \epsilon$$

is of the scale $R_L \geq \tilde{\Omega}\left(\sigma \sqrt{\frac{d}{m}}\right)$, where ϵ follows $\mathcal{N}(0, \sigma^2 \mathbf{I}_{md+m})$.

Proof. We first notice that the Jacobian $J_{1:B}(\mathbf{x}_{1:B})$ equals a concatenate of $J_1(\mathbf{x}_1), \dots, J_B(\mathbf{x}_B)$:

$$J_{1:B}(\mathbf{x}_{1:B}) = (J_1(\mathbf{x}_1), \dots, J_B(\mathbf{x}_B)).$$

Therefore

$$\begin{aligned} \text{tr}(J_{1:B}(\mathbf{x}_{1:B})^\top J_{1:B}(\mathbf{x}_{1:B})) &= \sum_{i=1}^B \text{tr}(J_i(\mathbf{x}_i)^\top J_i(\mathbf{x}_i)) \\ &= O(mdB), \end{aligned}$$

Note that in Definition D.1, a scaling of $\frac{1}{B}$ was added to the sum of risks. By Remark D.3, we have the lower bound $R_L^2 \geq \frac{1}{B} \frac{B^2 d^2}{\text{tr}(J_{1:B}(\mathbf{x}_{1:B})^\top J_{1:B}(\mathbf{x}_{1:B}))} \sigma^2 = \tilde{\Omega}\left(\frac{d}{m}\right) \sigma^2$. Then $R_L \geq \tilde{\Omega}\left(\sigma \sqrt{\frac{d}{m}}\right)$. \square

E Proofs with Lower Bounds with Defenses

E.1 Dropout

Dropout refers to randomly choosing certain entries of the model gradient and setting them to 0. Suppose the gradient is dropped out with a certain probability p , then we only obtain $1 - p$ of gradient information. In other words each row of $J(\mathbf{x})$ is similar to the Jacobian matrix in Theorem D.7 but removed with probability p . Therefore our lower bound under high probability is

$$R_{full} \geq \sqrt{\frac{1}{B} \frac{\sigma^2 d^2 B^2}{(1-p) \text{tr}(J(\mathbf{x})^\top J(\mathbf{x}))}} = \tilde{\Omega}\left(\sigma \sqrt{\frac{d}{(1-p)m}}\right)$$

where $J(\mathbf{x})$ is the complete Jacobian matrix without deleted rows.

E.2 Gradient Clipping

Clipping reduces the size of the gradient when the gradient size is too large. The noisy gradient could be written as

$$\min \left(\left\| \nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2 \right\|, C \right) \frac{\nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2}{\left\| \nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2 \right\|} + \epsilon,$$

where θ stands for the model parameters and ϵ is random noise with distribution $N(0, \frac{\sigma^2}{m^2} \mathbf{I}_{md+m})$.

When the norm of the gradient is smaller than the constant C , the lower bound remains unchanged. When the norm is higher than the constant C , the lower bound is of scale $\Omega(\frac{d \left\| \nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2 \right\|^2}{mC^2}) \sigma^2$. (When the number of model parameters is high we could ignore the effect of changes in the norm on the bound.)

Lemma E.1. *Under Assumption D.1, and for some constant C , when the norm of the model gradients $\|G\| \geq C$, with high probability the lower bound of minimax risk R_L with the observation*

$$O(S) = C \frac{G}{\|G\|} + \epsilon = C \frac{\nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2}{\left\| \nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2 \right\|} + \epsilon$$

is of the scale $R_L \geq \tilde{\Omega}(\sigma \sqrt{\frac{d}{m}} \frac{\|G\|}{C})$, where ϵ follows $\mathcal{N}(0, \sigma^2 \mathbf{I}_{md+m})$ and $G = \nabla_{\theta} \sum_{i=1}^B (y - f(\mathbf{x}_i))^2$.

Proof. Denote $\mathbf{g}_{\theta} = C \frac{G}{\|G\|}$, then for each \mathbf{x}_k ,

$$\begin{aligned} \nabla_{\mathbf{x}_k} \mathbf{g}_{\theta} &= C \frac{\nabla_{\mathbf{x}_k} G}{\|G\|} + C \left(\nabla_{\mathbf{x}_k} \frac{1}{\|G\|} \right) G^{\top} \\ &= C \frac{\nabla_{\mathbf{x}_k} G}{\|G\|} - C \left(\frac{\nabla_{\mathbf{x}_k} G}{\|G\|^3} G \right) G^{\top} \end{aligned}$$

Using the sub-multiplicative property of the Frobenius norm and scaling property with high probability of $\|\nabla_{\mathbf{x}_k} G\| = \|\nabla_{\mathbf{x}_k} \nabla_{\theta} (y - f(\mathbf{x}_k))^2\|_F^2 = \tilde{O}(md)$ proved in Lemma D.6, we have that

$$\begin{aligned} \|\nabla_{\mathbf{x}_k} \mathbf{g}_{\theta}\|_F^2 &\leq 2C^2 \left\| \frac{\nabla_{\mathbf{x}_k} G}{\|G\|} \right\|_F^2 + 2C^2 \left\| \frac{\nabla_{\mathbf{x}_k} G}{\|G\|^3} \right\|_F^2 \|GG^{\top}\|_F^2 \\ &\leq 2C^2 \frac{\tilde{O}(md)}{\|G\|^2} + 2C^2 \frac{\tilde{O}(md)}{\|G\|^6} \|GG^{\top}\|_F^2 \\ &= \frac{\tilde{O}(C^2 md)}{\|G\|^2} (1 + \|G\|^{-4} \|G\|^4) \\ &= \frac{\tilde{O}(C^2 md)}{\|G\|^2}. \end{aligned}$$

Therefore we have with high probability that

$$\begin{aligned} \|\nabla_{\mathbf{x}} \mathbf{g}_{\theta}\|_F^2 &= \sum_{i=1}^B \|\nabla_{\mathbf{x}_i} \mathbf{g}_{\theta}\|_F^2 \\ &= \sum_{i=1}^B \frac{\tilde{O}(C^2 md)}{\|G\|^2} \\ &= \frac{\tilde{O}(C^2 mdB)}{\|G\|^2} \end{aligned}$$

Therefore our lower bound is given as

$$R_L \geq \tilde{\Omega} \left(\sqrt{\frac{1}{B} \frac{B^2 d^2}{\|\nabla_{\mathbf{x}} \mathbf{g}_{\theta}\|_F^2}} \sigma^2 \right) = \tilde{\Omega} \left(\sqrt{\frac{d}{m}} \frac{\|G\|}{C} \right),$$

which is the desired result. \square

E.3 Local Aggregation

In local aggregation, we calculate two updates of model parameters using two different batches of size B . In local aggregation, we use different learning rates for a_j and \mathbf{w}_j since their scalings are different. Specifically, we take $\eta_a = O(\frac{1}{m^2d})$ and $\eta_w = O(\frac{1}{d})$. Then $a_j^{(1)} = a_j - \eta_a \sum_{i=1}^B \nabla_{a_j} \ell$ and $\mathbf{w}_j^{(1)} = \mathbf{w}_j - \eta_w \sum_{i=1}^B \nabla_{\mathbf{w}_j} \ell$ for all j . Denote $\theta^{(1)} = (a^{(1)}, \mathbf{w}^{(1)})$ and we can similarly define $a^{(2)}$, $\mathbf{w}^{(2)}$ and $\theta^{(2)}$. Denote $\ell(\mathbf{x}_i, \theta, y_i) = (y_i - f_\theta(\mathbf{x}_i))$, the direct observation from the local aggregation gradients is (with y_j omitted)

$$a_j^{(2)} - a_j = -\eta_a \left(\sum_{i=1}^B \nabla_{a_j} \ell(\mathbf{x}_i, \theta) + \sum_{i=B+1}^{2B} \nabla_{a_j} \ell(\mathbf{x}_i, \theta) \right)$$

and

$$\mathbf{w}_j^{(2)} - \mathbf{w}_j = -\eta_w \left(\sum_{i=1}^B \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta) + \sum_{i=B+1}^{2B} \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta) \right).$$

In the setting of reconstruction attack, both of the learning rates are known so we can directly observe the gradients for two steps with a Gaussian noise.

$$O(S) = \sum_{i=1}^B \nabla_\theta \ell(\mathbf{x}_i, \theta) + \sum_{i=B+1}^{2B} \nabla_{\theta^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) + \epsilon, \quad (29)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I_{md+d})$. We want to reconstruct all $2B$ data points from the update. Intuitively, this is approximately equivalent to a regular update with batch size $2B$ and learning rate 2η , so the lower bound should remain unchanged. We now prove the bound rigorously.

Proposition E.2 (Full version of Proposition 3.2). *If satisfy assumption D.1, $m > B$ and the learning rate for the two layers $\eta_w = O(1)$ and $\eta_a = O(\frac{1}{m^{3/2}})$, then with high probability the lower bound of minimax risk R_L with the observation Eq. (29) is of the scale $R_L \geq \tilde{\Omega}(\sigma \sqrt{\frac{d}{m}})$.*

Proof. Define

$$g(\mathbf{x}_{1:2B}) = \sum_{i=1}^B \nabla_\theta \ell(\mathbf{x}_i, \theta) + \sum_{i=B+1}^{2B} \nabla_{\theta^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}),$$

our lower bound is given as

$$\frac{1}{2B} \frac{B^2 d^2}{\sum_{i=1}^{2B} \|\nabla_{\mathbf{x}_i} g(\mathbf{x}_{1:2B})\|^2} \sigma^2.$$

Notice that all entries of $\nabla_a \ell(\mathbf{x}_i, \theta)$ are $\tilde{O}(1)$ and $\nabla_w \ell(\mathbf{x}_i, \theta)$ are $\tilde{O}(\frac{1}{m})$:

$$\begin{cases} \nabla_{a_j} \ell = 2 \sum_{i=1}^B (y - f(\mathbf{x}_i)) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) \\ \nabla_{\mathbf{w}_j} \ell = 2 \sum_{i=1}^B (y - f(\mathbf{x}_i)) a_j \sigma'(\mathbf{w}_j^\top \mathbf{x}_i) \mathbf{x}_i. \end{cases}$$

Therefore taking $\eta_a \leq O(\frac{1}{B})$ and $\eta_w \leq O(\frac{m}{B})$ would make the updated $a_j^{(1)} = a_j - \eta_a \nabla_{a_j} \ell$ and $\mathbf{w}_j^{(1)} = \mathbf{w}_j - \eta_w \nabla_{\mathbf{w}_j} \ell$ still satisfy scaling in Lemma D.5.

Therefore for $i \geq n+1$, with high probability

$$\|\nabla_{\mathbf{x}_i} g(\mathbf{x}_{1:2B})\|_F^2 = \|\nabla_{\mathbf{x}_i} \nabla_{\theta^{(1)}} \ell(\mathbf{x}_i, \theta_1)\|_F^2 = \tilde{O}(md).$$

For $i \leq n$, with high probability

$$\|\nabla_{\mathbf{x}_i} \nabla_{a_j} \ell(\mathbf{x}_i, \theta)\|^2 = \tilde{O}(d), \quad (30)$$

$$\|\nabla_{\mathbf{x}_i} \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta)\|_F^2 = \tilde{O}(\frac{d}{m^2}), \quad (31)$$

$$|\nabla_{a_k} \nabla_{a_j} \ell(\mathbf{x}_i, \theta)| = \tilde{O}(1) \quad (32)$$

for any j, k ,

$$\|\nabla_{a_j} \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta)\|^2 \leq \tilde{O}(1), \quad (33)$$

$$\|\nabla_{a_k} \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta)\|^2 \leq \tilde{O}(\frac{1}{m^2}) \quad (34)$$

for $j \neq k$,

$$\|\nabla_{\mathbf{w}_j} \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta)\|_F^2 \leq \tilde{O}(\frac{1}{m^2}) \quad (35)$$

and

$$\|\nabla_{\mathbf{w}_k} \nabla_{\mathbf{w}_j} \ell(\mathbf{x}_i, \theta)\|_F^2 \leq \tilde{O}(\frac{1}{m^4}) \quad (36)$$

for $j \neq k$. Therefore, by Eq. (30) to (36) we have

$$\begin{aligned} \|\nabla_{\mathbf{x}_i} g(\mathbf{x}_{1:2B})\|^2 &= \left\| \nabla_{\mathbf{x}_i} \nabla_{\theta^{(1)}} \ell(\mathbf{x}_i, \theta) - \sum_{i=B+1}^{2B} \left(\eta_w \nabla_{\mathbf{x}_i} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{\mathbf{w}^{(1)}} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right. \right. \\ &\quad \left. \left. + \eta_w \nabla_{\mathbf{x}_i} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{\mathbf{w}^{(1)}} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right. \right. \\ &\quad \left. \left. + \eta_a \nabla_{\mathbf{x}_i} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{a^{(1)}} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right. \right. \\ &\quad \left. \left. + \eta_a \nabla_{\mathbf{x}_i} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{a^{(1)}} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right) \right\|_F^2 \\ &\lesssim \|\nabla_{\mathbf{x}_i} \nabla_{\theta^{(1)}} \ell(\mathbf{x}_i, \theta)\|_F^2 + \sum_{i=1}^B \left(\eta_w \left\| \nabla_{\mathbf{x}_i} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{\mathbf{w}^{(1)}} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right\|_F^2 \right. \\ &\quad \left. + \eta_w \left\| \nabla_{\mathbf{x}_i} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{\mathbf{w}^{(1)}} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right\|_F^2 \right. \\ &\quad \left. + \eta_a \left\| \nabla_{\mathbf{x}_i} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{a^{(1)}} \nabla_{\mathbf{w}^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right\|_F^2 \right. \\ &\quad \left. + \eta_a \left\| \nabla_{\mathbf{x}_i} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta) \nabla_{a^{(1)}} \nabla_{a^{(1)}} \ell(\mathbf{x}_i, \theta^{(1)}) \right\|_F^2 \right) \\ &\lesssim \tilde{O}(md) + \sum_{i=1}^B \left(\eta_w^2 m \tilde{O}(\frac{d}{m^2}) (m \tilde{O}(1) + m(m-1) \tilde{O}(\frac{1}{m^2})) \right. \\ &\quad \left. + \eta_w^2 m \tilde{O}(\frac{d}{m^2}) (m \tilde{O}(\frac{1}{m^2}) + m(m-1) \tilde{O}(\frac{1}{m^4})) \right. \\ &\quad \left. + \eta_a^2 m \tilde{O}(d) (m \tilde{O}(1) + m(m-1) \tilde{O}(\frac{1}{m^2})) \right. \\ &\quad \left. + \eta_a^2 m \tilde{O}(d) m^2 \tilde{O}(1) \right) \\ &\leq \tilde{O}(md) + \eta_w^2 \tilde{O}(Bd) + \eta_a^2 \tilde{O}(Bm^3d) \\ &= \tilde{O}(md) \end{aligned}$$

if $\eta_w = O(\sqrt{m}B)$ and $\eta_a = O(\frac{1}{m\sqrt{B}})$.

Therefore the lower bound is

$$R_L^2 \geq \frac{1}{B} \frac{d^2 B^2}{\sum_{i=1}^{2B} \|\nabla_{\mathbf{x}_i} g(\mathbf{x}_{1:2B})\|^2} \frac{1}{m^2} \sigma^2 = \tilde{\Omega}(\frac{d}{m}) \sigma^2.$$

Thus, we have $R_L \geq \tilde{\Omega}(\sigma \sqrt{\frac{d}{m}})$. □

E.4 Gradient Pruning

Gradient pruning refers to changing small entries of the model gradient to zero before updating the model. If a model gradient was pruned, the entries corresponding to this term in the Jacobian matrix would become 0. Denote the Jacobian of pruned parameters' gradient on the input \mathbf{x} the matrix $J_0(\mathbf{x})$ (which is part of the full Jacobian $\text{tr}(J(\mathbf{x}))$), we will demonstrate in this section that gradient pruning has a lower bound at least as good as the regular bound in Proposition D.7.

Lemma E.3. *Under Assumption D.1, and for any constant γ , with high probability the lower bound of minimax risk R_L with the observation*

$$P\left(\sum_{i=1}^B \nabla_{\theta}(y - f(\mathbf{x}_i))^2\right) + \epsilon$$

is of scale $R_L \geq \tilde{\Omega}(\sigma \sqrt{\frac{d}{m(1-\hat{p})}})$, where $P(\mathbf{a})$ is the function that for all $i = 1, \dots, \dim(\mathbf{a})$,

$$P_i(\mathbf{a}) = \begin{cases} a_i & \text{if } |a_i| \geq \gamma \\ 0 & \text{if } |a_i| < \gamma \end{cases},$$

and

$$\hat{p} = \frac{\|J_0(\mathbf{x})\|_F^2}{\|J(\mathbf{x})\|_F^2}.$$

Proof. As the points that are not differentiable after applying this defense method are of zero measure, our analysis could ignore these cases.

The Jacobian $\nabla_{\mathbf{x}} P\left(\sum_{i=1}^B \nabla_{\theta}(y - f(\mathbf{x}_i))^2\right)$ is $\nabla_{\mathbf{x}} \sum_{i=1}^B \nabla_{\theta}(y - f(\mathbf{x}_i))^2$ with the columns corresponding to columns in J_0 being set to 0. Our lower bound is given as

$$R_L \geq \tilde{\Omega}(\sigma \sqrt{\frac{1}{B} \frac{d^2 B^2}{\|J(\mathbf{x})\|_F^2 - \|J_0 \mathbf{x}\|_F^2}}) = \tilde{\Omega}(\sigma \sqrt{\frac{1}{B} \frac{d^2 B^2}{\|J(\mathbf{x})\|_F^2} \frac{1}{1 - \hat{p}}}).$$

Which gives us the desired result. \square

Remark E.4. *Since entries of $G_{\mathbf{w}_j}$ scale smaller compared to $G_{\mathbf{a}_j}$, gradients from the first layer easier to be pruned. Therefore until the ratio of pruned gradients reaches $\frac{d}{d+1}$, the pruned gradients gives a small \hat{p} and does not change the lower bound much. But higher than this ratio we start to prune gradients of \mathbf{a}_j , so we predict that the bound will greatly increase after the ratio reaches $\frac{d}{d+1}$.*

F Additional experimental details for our proposed attack

Image Prior Network The architecture of our deep image prior network is structured on the foundation of PixelCNN++ as per (Salimans et al., 2017), and it utilizes a U-Net as depicted by (Ronneberger et al., 2015), which is built upon a Wide-ResNet according to (Zagoruyko and Komodakis, 2016). To simplify the implementation, we substituted weight normalization (Salimans and Kingma, 2016) with group normalization (Wu and He, 2018). The models operating on 32×32 utilize four distinct feature map resolutions, ranging from 32×32 to 4×4 . Each resolution level of the U-Net comprises two convolutional residual blocks along with self-attention blocks situated at the 16×16 resolution between the convolutional blocks. When available, the reconstructed features are incorporated into each residual block.

Backbone network for federated learning We adopt ResNet-18 (He et al., 2016) architecture as our backbone network. The network is pre-trained for one epoch for better results of gradient inversion. Note that this architecture results in a more challenging setting given its minimal information retention capacity and is observed to cause a slight drop in quantitative performance and large variance (Jeon et al., 2021). We further add an extra linear layer before the classification head for feature reconstruction. In order to satisfy the concentration bound $\tilde{O}(B \sqrt{\frac{d}{m}})$ of feature reconstruction, this linear layer has $512 \times 32 = 16384$ neurons.

We adopt a special design for the last but one layer in favor of both the gradient matching procedure as well as intermediate feature reconstruction step. Specifically, we find out that a pure random net will degrade the performance of gradient matching compared to a moderately trained network. Therefore we split the neurons into two sets; we set the first set (first 4096 neurons) as trained parameters from ResNet-18 and set the weights associated with the rest neurons as random weights as required for the feature reconstruction step.

Hyper-parameters for training For all experiments, we train the backbone ResNet-18 for 200 epochs, with a batch size of 64. We use SGD with a momentum of 0.9 as the optimizer. The initial learning rate is set to 0.1 by default. We decay the learning rate by a factor of 0.1 every 50 epochs.

Hyper-parameters of the attack The attack minimize the objective function given in Eq. ?? . We search α_{TV} in $\{0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ and α_{BN} in $\{0, 0.0005, 0.001, 0.01, 0.05, 0.1\}$ for attack without defense, and apply the best choices $\alpha_{TV} = 0.01$ and $\alpha_{BN} = 0.001$ for all defenses (we did not tune α_{TV} and α_{BN} for each defense as we observe the proposed attack method is robust in terms of these two hyper-parameters). We optimize the attack for 10,000 iterations using Adam (Kingma and Ba, 2014), with the initial learning rate set to 0.00001. We decay the learning rate by a factor of 0.1 at 3/8, 5/8, and 7/8 of the optimization. For feature matching loss strength α_f , we search in $\{0.01, 0.05, 0.1, 0.5\}$ for each defense and find that 0.1 is the optimal choice.

Batch size of the attack Zhu et al., 2019; Geiping et al., 2020 suggests that small batch size is vital for the success of the attack. We therefore intentionally evaluate the attack with three small batch sizes 2, 4, and 8 to test the upper bound of privacy leakage, and the minimum (and unrealistic) batch size 1, and two small but realistic batch sizes 16 and 32.

Gradient Reweighting The original gradient inversion loss, which calculates the cosine similarity of the gradients, can be expressed as

$$\sum_{i=1}^p \frac{\langle \nabla_{\theta_i} \mathcal{L}_{\theta_i}(\hat{x}, y), \nabla_{\theta_i} \mathcal{L}_{\theta_i}(x^*, y^*) \rangle}{\|\nabla_{\theta_i} \mathcal{L}_{\theta_i}(\hat{x}, y)\| \|\nabla_{\theta_i} \mathcal{L}_{\theta_i}(x^*, y^*)\|},$$

Here, p represents the number of parameters in the federated neural network. To make the optimization of this loss easier, since some parameters can have much larger gradients and therefore overshadow the effect of others, we reweight the loss:

$$\sum_{i=1}^p w_i \frac{\langle \nabla_{\theta_i} \mathcal{L}_{\theta_i}(\hat{x}, y), \nabla_{\theta_i} \mathcal{L}_{\theta_i}(x^*, y^*) \rangle}{\|\nabla_{\theta_i} \mathcal{L}_{\theta_i}(\hat{x}, y)\| \|\nabla_{\theta_i} \mathcal{L}_{\theta_i}(x^*, y^*)\|},$$

where $w_i = \|\theta_i\|_0$.

Technical details on feature reconstruction The feature reconstruction method suggested by Wang et al., 2023 always produces features that are normalized and may have a sign opposite to the actual features. To compare the similarity between the original and the reconstructed features, we use cosine similarity, as it doesn't change with feature magnitude. We square this similarity to avoid any issues with mismatched signs.

In addressing the challenge of integrating a tensor-based method, which requires randomly initialized networks, into gradient inversion processes that typically see improved performance with a few steps of preliminary training, we have designed a strategic approach for weight initialization. Since tensor-based method is only applied to the final fully connected layers, denoted as $A\sigma(Wx)$, where $A \in \mathbb{R}^{K \times m}$, $W \in \mathbb{R}^{m \times d}$, with K representing the number of classes, m the number of hidden nodes, and d the input dimension. We consider the original pre-trained weights to be partitioned into two sets: $W = [W_1, W_2]$ and $A = [A_1; A_2]$. Here, $W_2 \in \mathbb{R}^{v \times d}$ is initialized as a random Gaussian matrix, essential for the tensor-based method to function effectively. Concurrently, each row i of $A_2 \in \mathbb{R}^{K \times v}$ is initialized to a constant value of i/v , as suggested by the tensor-based approach. The remaining dimensions of the weight matrices, $W_1 \in \mathbb{R}^{(m-v) \times d}$ and $A_1 \in \mathbb{R}^{K \times (m-v)}$, retain their pre-trained weights, aligning with the requirements for successful gradient matching. For all of our experiments, the dimension allocated for gradient matching, $m - v$, is set as 2048. This strategic initialization approach not only accommodates the prerequisites of the tensor-based method but also maintains the efficacy of the gradient-matching process.

When deploying the tensor-based method for feature recovery, a challenge arises due to the algorithm's inability to precisely maintain the order and sign of features when dealing with batches larger than a single input. This challenge complicates the task of accurately matching the recovered features to their corresponding inputs within

a batch. In theory, one approach to circumvent this issue is to consider every possible permutation of the recovered features against the batch inputs, aiming to identify the correct alignment. However, this method is computationally intensive and impractical for larger batches. To streamline the process, we adopt a more efficient strategy that involves sequentially comparing each recovered feature against the actual input features using cosine similarity. This comparison continues until the most accurate order of features is determined based on their similarity scores. Although a single recovered feature might have high cosine similarity with several features corresponding to different inputs, we empirically observe that this greedy matching method achieves similar performance with the method that considers all possible pairs.

G Additional experimental details for other methods

For GradientInversion (Geiping et al., 2020), we adopted the code from (Huang et al., 2021), and the hyperparameters provided³. For Rob The Fed (Fowl et al., 2021), we follow strictly the setting mentioned in their paper and the code⁴, and introduced defenses to the gradients. For CPA (Kariyappa et al., 2023), since the settings and model architectures used in their paper are very different from ours, we incorporate their algorithm for feature recovery into our settings. i.e. using the latent features recovered by CPA rather than our proposed method. We perform hyperparameter selection through grid searches, but due to limited time, the hyperparameter searching process may not be comprehensive. But this should not significantly deviate our main goal – comparison of the effectiveness of different defense methods, (not the attack algorithms).

H Hardware and software

We run most of the experiments on clusters using NVIDIA V100s. All experiments are implemented using PyTorch library. Overall we estimated that a total of 800 GPU hours were consumed.

I Additional experimental results

Effectiveness of gradient reweighting. As observed in Table 6, when dealing with a large batch-size, adjusting the gradients notably enhances the quality of the final image reconstruction. We hypothesize that this improvement is due to the adjustment potentially modifying the loss landscape, aiding in the optimization process.

Effectiveness of feature matching. From Table 6, it’s evident that feature matching offers a minor enhancement in final image reconstruction, particularly with smaller batch sizes. This indicates that feature matching plays a more crucial role when dealing with larger batch sizes. Additionally, feature matching proves to be beneficial when defenses are in place. As suggested by Figure 4, feature matching can bring a notable improvement in reconstruction quality when applying defenses like additive noise. The comparison between Table 7 and Table 8 also indicates the effectiveness of feature matching in improving the robustness under defenses. For the defenses invalidating feature reconstruction and matching, for example, gradient pruning and large gradient noise, the performance has little improvement. On the contrary, for the defenses with no effect on feature reconstruction like gradient clipping, our method with feature matching improves a lot.

Effectiveness of random initialization. We conduct additional experiments by attacking the model at different time steps. Specifically, the RMSE for attacks after 1, 2, and 3 epochs of training are as follows: 0.15 (0.04), 0.17 (0.09), and 0.22 (0.08), respectively. These results indicate that the attack becomes progressively more challenging as the training continues. We use the attack at random initialization in all other experiments, which is the strongest setting.

³<https://github.com/Princeton-SysML/GradAttack/>

⁴https://github.com/lhfowl/robbing_the_fed

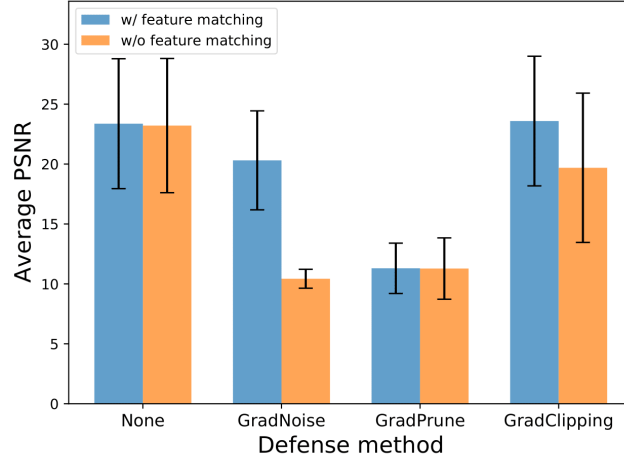


Figure 4: PSNR of reconstructed data using the method with and without feature matching. The difference indicates that feature matching improves robustness against defenses.

batch size	4			8			16			32		
Method	w/o GR and FM	w/o FM	Ours	w/o GR and FM	w/o FM	Ours	w/o GR and FM	w/o FM	Ours	w/o GR and FM	w/o FM	Ours
PSNR \uparrow	21.25 (6.02)	22.41 (5.84)	23.59 (5.39)	20.71 (5.96)	23.21 (5.60)	23.37 (5.42)	17.68 (5.61)	20.01 (5.33)	21.90 (5.18)	15.05 (4.05)	19.84 (5.30)	20.69 (4.88)
RMSE \downarrow	0.16 (0.05)	0.16 (0.04)	0.15 (0.03)	0.18 (0.05)	0.16 (0.04)	0.15 (0.04)	0.21 (0.06)	0.18 (0.06)	0.16 (0.04)	0.23 (0.06)	0.18 (0.05)	0.17 (0.05)
LPIPS \downarrow	0.12 (0.08)	0.09 (0.08)	0.09 (0.07)	0.15 (0.12)	0.11 (0.09)	0.10 (0.09)	0.24 (0.15)	0.16 (0.13)	0.14 (0.11)	0.30 (0.12)	0.18 (0.12)	0.15 (0.11)

Table 6: Results of ablation study on gradient reweighting (GR) and feature matching (FM). Both feature matching and gradient reweighting contribute the the final performance boost. When batch size is large (e.g. = 16/32), gradient reweighting significantly improves the performance.

	None		GradNoise (σ)				GradPrune (p)				GradClipping (C)				Local Aggregation (steps)	
Parameter	-	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.99	2	4	8	3	5	
Attack batch size = 2																
RMSE \downarrow	0.15 (0.04)	0.17 (0.05)	0.22 (0.07)	0.31 (0.06)	0.28 (0.08)	0.16 (0.08)	0.17 (0.07)	0.20 (0.09)	0.26 (0.09)	0.27 (0.10)	0.17 (0.08)	0.17 (0.08)	0.19 (0.08)	0.23 (0.09)	0.25 (0.10)	
PSNR \uparrow	23.69 (5.69)	23.31 (3.04)	21.32 (3.52)	14.58 (1.59)	13.11 (4.26)	22.63 (6.15)	23.33 (6.03)	19.40 (6.02)	16.39 (6.56)	14.72 (6.45)	23.26 (6.10)	23.40 (6.00)	23.01 (6.28)	18.65 (6.75)	18.77 (7.33)	
Attack batch size = 4																
RMSE \downarrow	0.15 (0.03)	0.19 (0.07)	0.24 (0.08)	0.29 (0.03)	0.27 (0.07)	0.16 (0.04)	0.16 (0.03)	0.21 (0.04)	0.27 (0.07)	0.27 (0.08)	0.16 (0.03)	0.16 (0.04)	0.16 (0.04)	0.28 (0.08)	0.26 (0.08)	
PSNR \uparrow	23.59 (5.39)	20.12 (3.58)	16.41 (4.64)	11.27 (0.33)	12.28 (4.08)	23.16 (5.48)	24.08 (5.38)	18.73 (5.63)	13.90 (3.77)	12.43 (2.73)	22.88 (5.31)	23.93 (5.55)	24.04 (5.58)	13.84 (5.38)	14.21 (5.41)	
Attack batch size = 8																
RMSE \downarrow	0.15 (0.04)	0.19 (0.05)	0.29 (0.05)	0.30 (0.03)	0.30 (0.06)	0.15 (0.03)	0.16 (0.04)	0.20 (0.05)	0.29 (0.05)	0.29 (0.05)	0.16 (0.03)	0.16 (0.04)	0.16 (0.04)	0.29 (0.04)	0.30 (0.04)	
PSNR \uparrow	23.37 (5.42)	20.31 (4.13)	14.56 (0.88)	11.27 (0.89)	11.27 (1.83)	23.74 (5.21)	23.32 (5.13)	18.25 (5.22)	11.30 (2.10)	11.13 (2.15)	23.36 (5.14)	23.59 (5.41)	23.47 (5.23)	11.25 (2.56)	10.82 (2.20)	

Table 7: Our method evaluated with different defense methods. With feature matching, our method performs well against most defenses. Specifically, gradient pruning shows its effectiveness under this stronger attack as well.

	None		GradNoise (σ)				GradPrune (p)				GradClipping (C)				Local Aggregation (steps)	
Parameter	-	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.99	2	4	8	3	5	
Attack batch size = 2																
RMSE ↓	0.16 (0.05)	0.30 (0.05)	0.32 (0.07)	0.32 (0.06)	0.35 (0.07)	0.19 (0.08)	0.20 (0.07)	0.24 (0.09)	0.28 (0.09)	0.27 (0.11)	0.19 (0.07)	0.23 (0.09)	0.25 (0.08)	0.25 (0.10)	0.25 (0.10)	
PSNR ↑	22.51 (4.72)	12.27 (3.27)	12.12 (3.45)	11.48 (1.53)	11.77 (2.00)	20.31 (4.35)	20.32 (7.03)	19.23 (7.02)	15.73 (6.70)	14.01 (5.95)	21.25 (4.10)	21.40 (6.00)	21.01 (4.28)	17.52 (6.88)	16.24 (6.58)	
Attack batch size = 4																
RMSE ↓	0.16 (0.04)	0.31 (0.01)	0.31 (0.03)	0.31 (0.03)	0.31 (0.07)	0.19 (0.04)	0.18 (0.03)	0.23 (0.04)	0.29 (0.07)	0.28 (0.07)	0.21 (0.03)	0.22 (0.04)	0.23 (0.03)	0.28 (0.078)	0.25 (0.09)	
PSNR ↑	22.41 (5.84)	12.12 (3.58)	10.53 (1.03)	10.51 (1.03)	9.98 (1.35)	20.16 (4.26)	21.08 (5.28)	17.03 (4.59)	11.88 (2.66)	12.64 (4.04)	19.02 (4.26)	19.64 (4.33)	21.38 (4.27)	13.08 (5.16)	14.31 (6.02)	
Attack batch size = 8																
RMSE ↓	0.16 (0.04)	0.30 (0.02)	0.31 (0.02)	0.31 (0.02)	0.31 (0.71)	0.20 (0.03)	0.21 (0.04)	0.25 (0.04)	0.30 (0.05)	0.30 (0.05)	0.22 (0.04)	0.21 (0.04)	0.21 (0.04)	0.31 (0.04)	0.31 (0.04)	
PSNR ↑	23.21 (5.60)	10.43 (0.79)	10.15 (0.80)	10.46 (0.87)	10.54 (1.38)	21.56 (4.32)	21.46 (6.23)	17.25 (4.59)	11.28 (2.56)	11.17 (2.19)	19.36 (5.14)	19.69 (6.23)	20.43 (4.27)	10.01 (2.22)	10.26 (2.03)	

Table 8: Results of the proposed method without feature matching loss with different defenses. Under defenses having no effect on feature reconstruction, the performance of the method without feature matching is much worse than that with feature matching.