Just a Simple Transformation is Enough for Data Protection in Vertical Federated Learning

Anonymous submission

Abstract

Vertical Federated Learning (VFL) enables collaborative training of deep learning models while maintaining privacy protection. However, the VFL procedure still has components that are vulnerable to attacks by malicious parties. In our work, we consider feature reconstruction attacks – a common risk targeting input data compromise. We theoretically claim that feature reconstruction attacks cannot succeed without knowledge of the prior distribution on data. Consequently, we demonstrate that even simple model architecture transformations can significantly impact the protection of input data during VFL. Confirming these findings with experimental results, we show that MLP-based models are resistant to SOTA feature reconstruction attacks.

Code —
https://github.com/anon43534/simple-tranformations

1 Introduction

Federated Learning (FL) (Kairouz et al. 2021; McMahan et al. 2023) introduces a revolutionary paradigm for collaborative machine learning, in which multiple clients participate in cross-device model training on decentralized private data. The key idea is to train global model without sharing the raw data among participants. Generally, FL can be divided into two types (Yang et al. 2019): horizontal (HFL)(Konečný et al. 2017; McMahan et al. 2023), when data is partitioned among clients by samples, and vertical (VFL)(Khan, ten Thij, and Wilbik 2023; Liu et al. 2024b; Wei et al. 2022; Yang et al. 2023) when features of data samples are distributed across clients. Since clients in HFL hold the same feature space, the global model is also the same for each participant. Consequently, the FL orchestrator (often reffered to as the server) can receive the parameter updates from each client. In contrast, VFL implies that different models are used for clients since their feature spaces differ. In this way, the participants communicate by intermediate outputs called activations.

The focus of this paper is on the privacy concepts of Vertical Federated Learning(Rodríguez-Barroso et al. 2023; Yu et al. 2024; Liu et al. 2024b), namely in Two Party Split Learning (simply, SL) (Gupta and Raskar 2018; Thapa et al. 2022), where the parties split model in such a way that the first several layers belongs to client, and the rest are processed at the master server. In SL the client shares its last layer (called Cut Layer) activations, instead of the raw data. As a canonical use case (Sun et al. 2022) of SL, one can think of advertising platform A and advertiser company B. Both parties own different features for each visitor: party A can record the viewing history, while B has the visitor's conversion rate. Since each participant has its own private information and they do not exchange it directly, the process of training a recommender system with data from A and B can be considered as Split Learning. We deeply discuss SL setting in Section 3.1.

With regard to practice, the types of attacks from an adversary party are divided into: label inference (Li et al. 2022b; Sun et al. 2022; Liu et al. 2024a; Erdoğan, Küpçü, and Çiçek 2022; Kariyappa and Qureshi 2022), feature reconstruction (Luo et al. 2021; Qiu et al. 2024a; Jin et al. 2022; Geiping et al. 2020; Gupta et al. 2022; Ye et al. 2022; Hu et al. 2022) and model reconstruction (Li et al. 2023; Gao and Zhang 2023; Fredrikson, Jha, and Ristenpart 2015; Shokri et al. 2017; Driouich et al. 2023; Ganju et al. 2018; Erdoğan, Küpçü, and Çiçek 2022). In particular, among all feature reconstruction attacks in Split Learning, we are interested in Model Inversion attacks(Erdoğan, Küpçü, and Çiçek 2022; Fredrikson, Jha, and Ristenpart 2015; He, Zhang, and Lee 2019, 2021; Nguyen et al. 2023a,b): one that aims to infer and reconstruct private data by abusing access to the model; and Hijacking attacks(Pasquini, Ateniese, and Bernaschi 2021; Fu et al. 2023): when the malicious party with labels holds an auxiliary dataset from the same domain of the training data of the defending parties, thus, the adversary has prior knowledge of the data distribution.

After revisiting **all the attacks**, we highlight that SOTA MI and Hijacking attacks (without the White-Box assumption) (Erdoğan, Küpçü, and Çiçek 2022; Pasquini, Ateniese, and Bernaschi 2021) acquire a knowledge of prior on data distribution (Section 2). Additionally, these attacks are validated **only on CNN-based** models, bypassing MLPs, which are also show promise in the same domains. This leads to further questions:

1. Is it that simple to attack features, or does the data prior's knowledge give a lot?
2. Does the architectural design plays a crucial role in effectiveness of the latter attacks?

3. Can we develop a theoretical intuition that MLPs might be more privacy-preserving?

In this work, we answer these question affirmatively. Following our theoretical justification from Section 3.3, by experimentally validating the proposed Hypothesis 1, we reveal that MI (Erdoğan, Küpçü, and Çiçek 2022) and Hijacking (Pasquini, Ateniese, and Bernaschi 2021) attacks fail on MLP-based client-side model. Thus, we neither consider a specific defense framework nor propose a novel method. In contrast, **we demonstrate the failure of feature reconstruction attakcs when the architecture is MLP**. We summarize our contributions as follows:

(Contribution 1) We prove that without additional information about the prior distribution on the data, the feature reconstruction attack in Split Learning cannot be performed even on a one-layer (dense) client-side model. For MLPs we state the server's inability to reconstruct the activations in the hidden-space. Furthermore, we provably guarantee that (semi)orthogonal transformations in the client data and weights initialization do not change the transmitted activations during training under the `GD`-like algorithms (see Section 3.3 and Appendix A.4), and also do not affect convergence for `Adam`-like algorithms.

(Contribution 2) We show that Hijacking and Model Inversion attacks fail on MLP models **without any additional assumptions**. We show the effectiveness of our approach against the UnSplit (Erdoğan, Küpçü, and Çiçek 2022) and Feature-space Hijacking (Pasquini, Ateniese, and Bernaschi 2021) attacks on popular community datasets (Krizhevsky 2009; Lecun et al. 1998; Xiao, Rasul, and Vollgraf 2017) and argue that feature reconstruction attacks can be prevented without resorting to any of the defenses, while preserving the model accuracy on the main task. Also, our findings **can be combined with any of the defense frameworks** covered in Section B.

(Contribution 3) We reconsider the perception of defense quality from a human-side perspective and evaluate the resistance against an attacker using the Fréchet inception distance (FID) (Heusel et al. 2017) between the true data and the reconstructed ones. And report the comparison with commonly used MSE in Sections 4 and 5.

2 Background and Related Work

Recent feature reconstruction attacks show promising results. Meanwhile, these attacks sometimes require strong assumptions about the capabilities of the attacking side. For example, methods from (Qiu et al. 2024a; Jin et al. 2022) assume access not only to the architecture, but also to the client-side model parameters during each step of optimization process (White-Box). The above assumptions rarely occur in real-world applications, as such knowledge is not naturally aligned with the SL paradigm. Nevertheless, an adaptive obfuscation framework from (Gu et al. 2023) successfully mitigates the (Jin et al. 2022) attack. Moreover, the attacker's setup from these works is more valid for the HFL case (see (Geiping et al. 2020)), where the model is shared among clients and can be trained with (McMahan et al. 2023; Li et al. 2020) algorithms, rather than for VFL. Therefore, such a strong settings are not considered in our work.

2.1 Model Inversion attacks

Model Inversion attack (MI)(Fredrikson, Jha, and Ristenpart 2015; He, Zhang, and Lee 2021, 2019; Zhao, Mopuri, and Bilen 2020; Zhu, Liu, and Han 2019; Wu et al. 2016) is a common approach in machine learning, where an adversary party (server in our case) trains a clone of the client-side model to reconstruct raw data given the client activations. Recent works (Erdoğan, Küpçü, and Çiçek 2022; Li et al. 2022a; Fredrikson, Jha, and Ristenpart 2015) demonstrate that Split Learning is also vulnerable to MI attacks. Meanwhile, the most popular defense frameworks (Li et al. 2022a; Sun et al. 2021), aiming to protect data from MI attack, are effective against the adversary with White-Box access, which does not hold in real-world, and require imitation of the attacker (called attacker-aware training) using client-side inversion models, which leads to a 27% floating point operations (FLOPs) computational overhead(see Li et al. (2022a) Table 6).

Next, we come to Unsplit, proposed in Erdoğan, Küpçü, and Çiçek (2022), the main MI attack aiming to reconstruct input image data by exploiting an extended variant of coordinate descent (Wright 2015). Given the client model f_{c}, its clone $\tilde{f}_{\mathrm{c}}$ (i.e., the randomly initialized model with the same architecture), the adversary server attempts to solve the two-step optimization problem:

$$\tilde{X}^* = \arg\min_{\tilde{X}} \mathcal{L}_{\mathrm{MSE}}\left(\tilde{f}_{\mathrm{c}}(\tilde{W}_{\mathrm{c}}, \tilde{X}),\ f_{\mathrm{c}}(W_{\mathrm{c}}, X)\right) + \lambda \mathrm{TV}(\tilde{X}),$$

$$\tilde{W_{\mathrm{c}}}^* = \arg\min_{\tilde{W}_{\mathrm{c}}} \mathcal{L}_{\mathrm{MSE}}\left(\tilde{f}_{\mathrm{c}}(\tilde{W}_{\mathrm{c}}, \tilde{X}),\ f_{\mathrm{c}}(W_{\mathrm{c}}, X)\right).$$

In this context, X, W_{c} represent the client model's private inputs and parameters; TV denotes the total variation distance (Rudin, Osher, and Fatemi 1992) for image pixels (this term allows the attacker to use prior on data distribution); and $\tilde{X}^*$, $\tilde{W_{\mathrm{c}}}^*$ are the desired variables for the attacker's reconstructed output and parameters, respectively. Whereas, λ is the coefficient to modify the impact of the total variation, e.g., minimizing $\mathrm{TV}(\tilde{X})$ results in smoother images. At the beginning of the Unsplit attack , "mock" features $\tilde{X}$ initializes as a constant matrix.

It should be noted that this optimization process can be applied both before and after training f_{c}. The latter corresponds to feature reconstruction during the inference stage. The authors assume that the server is only aware of the architecture of the client model f_{c}. See Section 3 for the theoretical results and Section 4 for the experimental justification.

2.2 Hijacking attacks

The Feature-space Hijacking Attack (FSHA) was initially proposed in Pasquini, Ateniese, and Bernaschi (2021), for simplicity, we call attacks of this type as "Hijacking" and the attack from this work we will also call FSHA. The authors mention that the server's ability to control the learning process is the most pervasive vulnerability of SL. Which is not used in UnSplit setting. Indeed, since the server is able to guide the client model f_{c} towards the required functional

states, it has the capacity to reconstruct the private features X. In hijacking attacks (Fu et al. 2023; Pasquini, Ateniese, and Bernaschi 2021; Yu et al. 2023), the malicious server exploits an access to a public dataset X_{pub} of the same domain as X to subdue the training protocol.

Specifically, in FSHA, the server initializes three additional models: encoder ψ_{E}, decoder ψ_{D} and discriminator D. While the client-side model $f_{\text{c}} : \mathcal{X} \to \mathcal{Z}$ is initialized as a mapping between the data distribution $\mathcal{X}$ and a hidden-space $\mathcal{Z}$, the encoder network $\psi_{\text{E}} : \mathcal{X} \to \tilde{\mathcal{Z}}$ dynamically defines a function to certain subset $\tilde{\mathcal{Z}} \subset \mathcal{Z}$. Since the goal is to recover $X \in \mathcal{X}$, to ensure the invertibility of ψ_{E}, the server trains the decoder model $\psi_{\text{D}} : \mathcal{Z} \to \mathcal{X}$. To guide f_{c} towards learning $\tilde{\mathcal{Z}}$, server uses a discriminator network trained to assign high probability to the $\psi_{\text{E}}(X_{\text{pub}})$ and low to the $f_{\text{c}}(X)$.

The general scheme of the attack is the following:

$$\psi_{\text{E}}^*,\ \psi_{\text{D}}^* = \arg\min_{\text{E,D}} \mathcal{L}_{\text{MSE}}\left(\psi_{\text{D}}(\psi_{\text{E}}(X_{\text{pub}})),\ X_{\text{pub}}\right),$$

$$D^* = \arg\min_{\text{D}} \left[\log(1 - D(\psi_{\text{E}}(X_{\text{pub}}))) + \log(D(f_{\text{c}}(X)))\right],$$

$$f_{\text{c}}^* = \arg\min_{f_{\text{c}}} \left[\log\left(1 - D(f_{\text{c}}(X))\right)\right].$$

And, finally, server recovers features with:

$$\tilde{X} = \psi_{\text{D}}\left(f_{\text{c}}(X)\right).$$

This paper has led to the creation of other works that study FSHA. Erdogan, Küpçü, and Cicek (2022) propose a defense method SplitGuard in which the client sends fake batches with mixed labels with a certain probability. Then, the client analyzes the gradients corresponding to the real and fake labels and computes SplitGuard score to assess whether the server is conducting a Hijacking Attack and potentially halt the training. In response to the SplitGuard defense, Fu et al. (2023) proposed SplitSpy: where it is observed that samples from the batch with the lowest prediction score are likely to correspond to the fake labels and should be removed during this round of FSHA. Therefore, SplitSpy computes gradients from discriminator D only for survived samples. We would like to outline that this attack uniformly weaker compared to the original FSHA (Pasquini, Ateniese, and Bernaschi 2021) in the absence of the SplitGuard defense. Thus, we will only consider this attack later.

2.3 Quality of the defense

In (Sun et al. 2023), authors study the faithfulness of different privacy leakage metrics to human perception. Crowdsourcing revealed that hand-crafted metrics (Sara, Akter, and Uddin 2019; Pedersen and Hardeberg 2012; Zhang et al. 2018; Wang and Bovik 2002) have a weak correlation and contradict with human awareness and similar methods(Zhang et al. 2018; Huynh-Thu and Ghanbari 2008). From this point of view, we reconsider the usage of the MSE metric for the evaluation of the defense against feature reconstruction attacks, i.e., the quality of reconstruction. Given that the main datasets contain images, we suggest to rely on Frechet Inception Distance (FID) (Heusel et al. 2017). Besides the fact that MSE metric is implied into the attacker algorithms (Sections 2.1 and 2.2), most of works on evaluation of the images quality rely on FID. From the privacy perspective, the goal of the successful defense evaluation is to compare privacy risks of a classification model under the reconstruction attack. This process can be formalized for Split Learning in the following way: let the attack mechanism $\mathcal{M}$ aiming to reconstruct client model f_{c} data X given the Cut Layer outputs H, depending on the setup, $\mathcal{M}$ can access the client model architecture (in other settings this assumption may differ), then the privacy leakage is represented as

$$\text{PrivacyLeak} = \text{InfoLeak}\left(X, \mathcal{M}(H, f_{\text{c}})\right),$$

where InfoLeak stands for the amount of information leakage in reconstructed images $X_{\text{rec}} = \mathcal{M}(H, f_{\text{c}})$. Note that, $\mathcal{M}$ receives the Cut Layer outputs H at every iteration; then, the PrivacyLeak can also be measured during every iteration of the attack. Generally, information leakage can be represented through the hand-crafted metric ρ: $\text{InfoLeak} = \rho(X, X_{\text{rec}})$.

3 Problem Statement and Theoretical Motivation

In this section we will:
1. Outline the (Two Party) Split Learning setting. (Section 3.1)
2. Demonstrate that (semi)orthogonally transformed data and weights result in an identical training process from the server's perspective. (Lemma 1)
3. Prove that in this scenario, even a malicious server cannot reconstruct features without prior knowledge of the data distribution. (Lemma 2)
4. Show that similar reasoning applies to the distribution of activations before the Cut Layer. (Lemma 3)
5. Propose Hypothesis 1 explaining why SOTA feature reconstruction attacks achieve significant success and suggest potential remedies.

3.1 Problem Statement

Notation. We denote the client's model in SL as f_{c}, with the weights W. Under the X_{c}, we consider a design matrix of shape $\mathbb{R}^{\text{n} \times \text{d}_{\text{c}}}$. We denote activations that client transmits to the server as $H \in \mathbb{R}^{\text{n} \times \text{d}}$, while $\mathcal{Z}$ and $\mathcal{X}$ are the hidden-space and the data distribution, respectively. n corresponds to the number of samples in the dataset X_{c}, while d_{c} stands for the features belonging to the client and d is a hidden-size of the model. f denotes the **loss function of the entire model** (both server and client). Next, we provide a detailed description of our setup.

Setup. From the perspective of one client, it cannot rely on any information about the other parties during VFL. Then, to simplify the analysis, we consider the Two Party Split Learning process. The server s (label-party) holds a vector of labels, while the other data is located at the client-side c matrix X_{c}. Server and client have their own neural networks. Server's part of the model produces the final predictions. In each iteration, the non-label party computes activations $H = f_{\text{c}}(X_{\text{c}}, W)$ and sends it to the server. Then, the

remaining forward computation is performed only by server, which leads to predictions and, consequently, to the loss of f. In the backward phase, client receives $\frac{\partial f}{\partial H}$, and computes the gradient with respect to their parameters $\frac{\partial f}{\partial W} = \frac{\partial f}{\partial H}\frac{\partial H}{\partial W}$.

3.2 Motivation: Orthogonal transformation of data and weights stops the attack

In this section we consider client f_c as **one-layer** linear model $f_c = X_c W$ with $W \in \mathbb{R}^{d_c \times d}$. Note that **(semi)orthogonal** transformations $X_c \to X_c U$, $W_0 \to U^\top W_0$ preserve the outputs of f_c at the initialization. The following lemma states, that it also holds for subsequent iteration of (Stochastic) Gradient Descent:

Lemma 1. *For a one-layer linear model trained using* `GD` *or* `SGD`*, there exist continually many pairs of client data and weights initialization that produce the same Split Learning protocol.*

The complete proof of this lemma is presented in appendix A.1. These pair has the form $\{\tilde{X}, \tilde{W}_0\} = \{XU, U^\top W_0\}$, where U - arbitrary orthogonal matrix. With such orthogonal transformations, the client produces **the same activations at each step**, as if we had left X_c and W_0 unchanged. The server cannot distinguish between the different data distributions that produce identical activations; therefore, the true data also cannot be obtained. This results in:

Remark 1. *Under the conditions of Lemma 1, if the server has no prior information about the distribution of X_c, the label party cannot reconstruct initial data X_c (only up to an arbitrary orthogonal transformation).*

Recent work (Ye et al. 2022) states similar considerations, but their remark about `Adam` (Kingma and Ba 2017) and `RMSprop` (Graves 2014) not changing the Split Learning protocol is false. In fact, Lemma 1 holds only for algorithms whose update step is linear with respect to the gradient history (see Appendix A for details). However, while `Adam` and `RMSProp` do not preserve the SL protocol in terms of full matching of transmitted activations, we can relax the conditions and consider the properties of these algorithms from the perspective of "protocol preservation" in the sense of maintaining convergence to the same value. To begin with, let us note the following:

Remark 2. *The model's optimal value f^* after Split Learning is the same for any orthogonal data transformation. Indeed, $\forall \tilde{X}_c = X_c U \; \exists \tilde{W}^* = U^\top W^* : \; f(\tilde{X}_c, \tilde{W}^*) = f^* = f(X_c, W^*)$. Thus, f^* remains the same if we correspondingly rotate the optimal weights.*

In the case of a convex or strongly-convex function (entire model) $f(X_c, W)$, the optimal value f^* is unique, and therefore any algorithm is guaranteed to converge to f^* for any data transformation. Meanwhile, for general non-convex functions, convergence behavior becomes more nuanced: in fact, in the Example 1, we present a function on which the `Adam` algorithm converges before data and weight transformation and diverges after the transformation. However, the situation changes when we turn to functions satisfying the Polyak-Łojasiewicz-condition (PL), which is used as a canonical description of the neural network[1]. We, then, provably claim that SL protocol is preserved for PL functions with orthogonal transformations of data and weights. We show that `Adam`'s preconditioning matrices can be bounded regardless of the W initialization, and derive a Descent Lemma 5 with the modification of bounded gradient Assumption 3 similar to prior works (Sadiev et al. 2024; Défossez et al. 2022). The converges guarantees are covered in Lemma 6 and the theoretical evidence can be found in Appendix A.4.

Compared to Lemma 1, even knowledge of weights does not help the attacker:

Corollary 1. *Under the conditions of Lemma 1, assume that server knows the first layer W_0 of f_c, and let this layer be an invertible matrix. Then, the label party cannot reconstruct initial data X_c (only up to an arbitrary orthogonal transformation).*

Indeed, the activations send to the server in the first step: $H_1 = X_c W_0$, but if the client performs an orthogonal transformation leading to $\tilde{X}_c$, then server can recover only the $\tilde{H}_1 W_0^{-1}$, where $\tilde{H}_1 = \tilde{X}_c W_0$. Meanwhile, the difference between X_c and $\tilde{X}_c$ affects only the initialization of weights, and thus should not change the final model performance much.

Next, we conclude that **even a malicious server** cannot reconstruct the client's data without the additional prior on X_c.

Lemma 2. *Under the conditions of Lemma 1, assume training with the malicious server sending arbitrary vectors instead of real gradients $G = \partial f / \partial H$. In addition, the server knows the initialization of the weight matrix W_0. Then, if the client applies a non-trainable orthogonal matrix before W_0, the malicious server cannot reconstruct initial data X_c (only up to an arbitrary orthogonal transformation).*

Remark 3. *With the same reasons as for the Lemma 1, if even the malicious server from Lemma 2 has no prior information about the distribution of X_c, it is impossible for the label party to reconstruct initial data X_c.*

3.3 Motivation: You cannot attack the activations before "Cut Layer"

Up until now, we considered the client-side model with one linear layer W and proved that orthogonal transformation of data X_c and weights W lead to the same training protocol. The intuition behind Lemma 1 and 2 suggests that in the client model, one should look for layers whose inputs cannot be given the prior distribution. This brings us to the consideration of Cut Layer, since this is a "bridge" between the client and server. The closer Cut Layer to the first layer of the client's model f_c, the easier it is to steal data (Erdoğan, Küpçü, and Çiçek 2022; Li et al. 2022a); the complexity of attack increase with the "distance" between Cut Layer and data. We pose a question: "Does our intuition from Section

[1]Note that PL-condition does not imply convexity, see footnote 1 from (Li et al. 2021).

3.2 apply for the activations before Cut Layer?" and we answer in the affirmative:

Lemma 3. *[Cut Layer Lemma] There exist continually many distributions of the activations before the linear Cut Layer that produce the same Split Learning protocol.*

The results of Lemma 3 lead to a promising remark. While server might have prior on original data distribution, acquiring a prior on the distribution of the activations before Cut Layer is, generally, much more challenging. The absence of knowledge regarding the prior distribution of activations, combined with the assertion in Lemma 3, yields a result for activations analogous to Remark 1. Specifically, even with knowledge of a certain prior on the data, the server can, at best, reconstruct **activations** only up to an orthogonal transformation[2].

3.4 Should we use dense layers against feature reconstruction attacks?

The findings from Section 3.3 indicate that the reconstruction of activations poses significant challenges for the server. However, many feature reconstruction attacks achieve considerable success. This raises the question: "Does the server's inability to reconstruct activations before the Cut Layer not impede its capacity to reconstruct data features?" Alternatively, "Could it be that the conditions of Lemma 3 do not hold in practical scenarios?"

To investigate this matter more thoroughly, we examined outlined in Sections 2.1 and 2.2 feature reconstruction attacks. Specifically, we focused on UnSplit (Erdoğan, Küpçü, and Çiçek 2022) and FSHA (Pasquini, Ateniese, and Bernaschi 2021), which are SOTA representatives (to the best of our knowledge) of the Model Inversion and Feature Space Hijacking attack categories, respectively. UnSplit requires knowledge of the client-side model architecture, while FSHA should know the dataset X_{pub} of the same distribution as the original X. Assumptions are quite strong in the general case, but, we, in turn, argue that their attacks can be mitigated **without any additional modifications** in UnSplit (Erdoğan, Küpçü, and Çiçek 2022) and FSHA (Pasquini, Ateniese, and Bernaschi 2021) assumptions (see Section 4).

Both of these attacks are validated exclusively on image datasets, utilizing CNN architectures. Consequently, the client-side model architectures **lack fully connected (dense) layers before Cut Layer and the conditions of Lemma 3 do not hold**.

While a convolutional layer is inherently a linear operation and can be represented as matrix multiplication — where the inputs and weights can be flattened into 2D tensors — the resulting matrix typically has a very specific structure. In particular, all elements except for $(kernel\ size) \cdot (kernel\ size)$ entries in each row are zero. Therefore, an inverse transform does not exist in a general sense — meaning not every matrix multiplication can be expressed as a convolution, as the resultant matrix generally contains significantly more non-zero elements. As a result, "merging" an orthogonal matrix into a convolutional layer by multiplying the convolution weights with an orthogonal matrix is impossible, since this would result in a matrix with an excess of non-zero elements.

Based on this observation we propose:

Hypothesis 1. *Could it be that the attacks are successful due to the lack of dense layers in the client architecture? Will usage of MLP-based architectures for f_c, instead of CNNs, be more privacy preserving against Model Inversion attack and FSHA?*

We intend to experimentally test this conjecture in the following section.

4 Experiments

This section is dedicated to the experimental validation of the concepts introduced earlier. To test our Hypothesis 1, we evaluate the effectiveness of UnSplit and FSHA on MNIST (Lecun et al. 1998) and F-MNIST (Xiao, Rasul, and Vollgraf 2017) in setting where at least one dense layer is present on the client side.

Figure 1: Results of UnSplit attack on MNIST. (**Top**): Original images. (**Middle**): CNN-based client model. (**Bottom**): MLP-based client model.

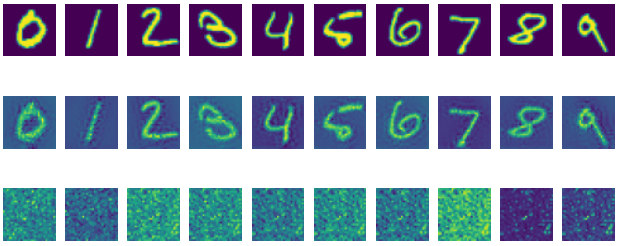
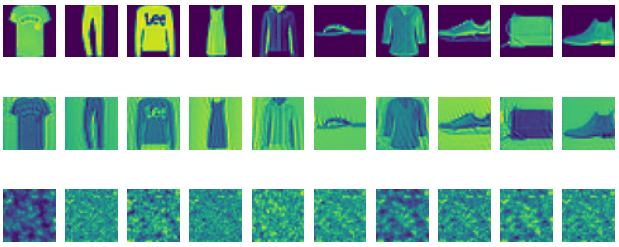


Figure 2: Results of UnSplit attack on F-MNIST. (**Top**): Original images. (**Middle**): CNN-based client model. (**Bottom**): MLP-based client model.



It is important to note that although MLP-based architecture may not be conventional in the field of Computer Vision (where CNN usage is more prevalent), dense layers are the backbone of popular model architectures in many other Deep Learning domains, such as Natural Language Processing, Reinforcement Learning, Tabular Deep Learning, etc. In

[2]Excluding degenerate cases, such as when the server knows that the client's network performs an identity transformation.

these domains, dense layers are commonly found at the very start of the architecture, and thus, when the network is split for VFL training, these layers would be contained in f_c. Furthermore, even within the Computer Vision field, there is a growing popularity of architectures like Vision Transformers (ViT) (Dosovitskiy et al. 2021) and MLP-Mixer (Tolstikhin et al. 2021), which also incorporate dense layers at the early stages of data processing. Therefore, we contend that with careful architectural selection, integrating dense layers on the client side should not lead to a significant deterioration in the model's utility score.

4.1 UnSplit

Before delving into the primary experiments of our study, we must note that unfortunately we were unable to fully reproduce the results of UnSplit using the code from their repository. Specifically, the images reconstructed through the attack were significantly degraded when deeper Cut Layers were used (see column "Without Noise" in Table 4). However, for the case where $cut\ layer = 1$ (i.e., when there is only one layer on the client side), the images were reconstructed quite well. Therefore, we used this setup for our comparisons.

Table 1: UnSplit attack on MNIST.

Model	MSE $\mathcal{X}$	MSE $\mathcal{Z}$	FID	Acc%
MLP-based	0.27	$3 \cdot 10^{-8}$	394	98.42
CNN-based	0.05	$2 \cdot 10^{-2}$	261	98.68

Table 2: UnSplit attack on F-MNIST.

Model	MSE $\mathcal{X}$	MSE $\mathcal{Z}$	FID	Acc%
MLP-based	0.19	$4 \cdot 10^{-5}$	361	88.31
CNN-based	0.37	$4 \cdot 10^{-2}$	169	89.21

Table 3: UnSplit attack on CIFAR-10.

Model	MSE $\mathcal{X}$	MSE $\mathcal{Z}$	FID	Acc%
MLP-based	1.398	$6 \cdot 10^{-6}$	423	89.29
CNN-based	0.056	$4 \cdot 10^{-3}$	455	73.61

As previously mentioned, to test Hypothesis 1, we utilized an MLP model with single or multiple dense layers on the client side. For CIFAR-10, we use MLP-Mixer, which maintains the performance of a CNN-based model while incorporating dense layers into the design. The results of the attack are shown in Figures 1 and 2. Despite our efforts to significantly increase the λ parameter in the TV up to 100 – thereby incorporating a stronger prior about the data into the attacker's model – the attack failed to recover the images, thus supporting the assertion of Lemma 1.

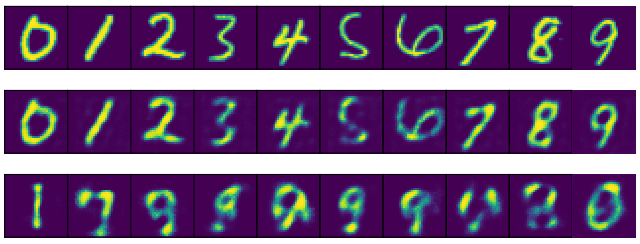
Additionally, Tables 1 and 2 presents the reconstruction loss values between normalized images. Here MSE $\mathcal{X}$ and FID shows the difference between the original and reconstructed images, and MSE $\mathcal{Z}$ refers to the loss between the activations $H = f_c(W_c, X)$ and $\tilde{H} = \tilde{f}_c(\tilde{W}_c, \tilde{X})$. Acc% denotes final accuracy of the trained models, as we can see, the results of MLP-based model are very close to its CNN-based counterpart.

In the image space X, FID appears to be a superior metric compared to MSE for accurately capturing the consequences of the attack. Furthermore, the tables show the MSE between activations before the Cut Layer for both the original and reconstructed images. These results indicate that in the case of the dense layer, the activations almost completely match, with significantly lower MSE than those even for well-reconstructed images. This implies that while the attack can perfectly fit $H = XW$, it fails to accurately recover X.

4.2 FSHA

Similarly to the previous subsections, we replaced the client's model in the FSHA attack (Pasquini, Ateniese, and Bernaschi 2021) with an MLP consisting of one or multiple layers. The attacker's models also varied, ranging from ResNet (He et al. 2015) architectures (following the original paper) to MLPs, ensuring that the attacker's capabilities are not constrained by the limitations of any architectural design. The results, illustrated in Figures 3 and 4, consistently demonstrate that the malicious party fully reconstructs the original data in the case of the ResNet architecture and completely fails in the case of the Dense layer.

Figure 3: Results of FSHA attack on MNIST. (**Top**): Original images. (**Middle**): CNN-based client model. (**Bottom**): MLP-based client model.



In addition to the reconstructed data shown in Figure 6, we computed the Reconstruction error and Encoder-Decoder error for a client using a ResBlock architecture (as in the original paper) and a client employing an MLP architecture. These plots reveal that the Encoder-Decoder pair for both architectures is equally effective at reconstructing data from the public dataset on the attacker's side. However, a challenge arises on the attacker's side with the training of GAN (Goodfellow et al. 2014). It is evident that in the presence of a Dense layer on the client side, the GAN fails to properly align the client's model representation within the

Figure 4: Results of FSHA attack on F-MNIST. (**Top**): Original images. (**Middle**): CNN-based client model. (**Bottom**): MLP-based client model.

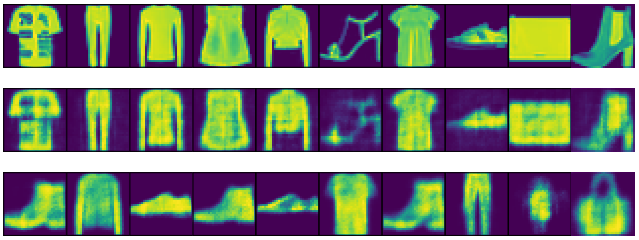


Figure 5: Results of UnSplit attack on CIFAR-10. (**Top**): Original images. (**Middle**): CNN-based client model. (**Bottom**): MLP-Mixer client model.

required subset of the feature space. Instead, it converges to mapping models of all classes into one or several modes within the activation space, corresponding to only a few original classes. This phenomenon is particularly well illustrated for the F-MNIST dataset in Figure 4.

4.3 Evaluation with FID

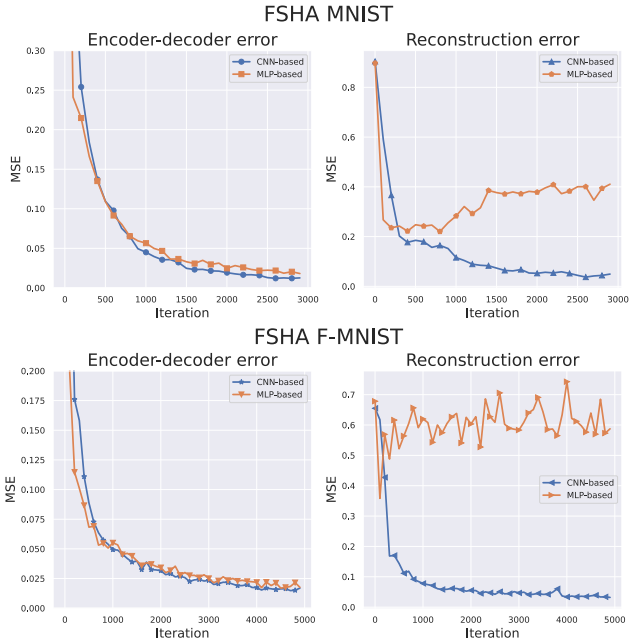
Inspired by prior works on GANs (Goodfellow et al. 2014), we apply FID to the InfoLeak scheme for the next reasons: (1) FID measures the information leakage as the distribution difference between between original and reconstruction images, thus $\text{InfoLeak}(X, X_{\text{rec}}) \propto \text{FID}(X, X_{\text{rec}})$. (2) Usage of FID is a more common approach when dealing with images. (3) The widespread metric in reconstruction evaluation is MSE, that lacks an interpretation for complex images (Sun et al. 2023), at least from the CIFAR-10(Krizhevsky 2009) dataset. However, we notice that the privacy evaluation of feature reconstruction attacks requires refined.

The values of FID and MSE in Tables 1 and 2 suggest that FID is a more accurate reflection of the attack's outcomes than MSE in the image space X. For instance, on the F-MNIST dataset, the MSE is higher for a CNN architecture despite the better quality of the reconstructed images. This discrepancy appears to stem from differences in background pixel values compared to the original images.

5 Discussions

With our work, we contribute to a better understanding of the meaningfulness of feature reconstruction attacks. We show

Figure 6: Encoder-decoder error and Reconstruction error for FSHA attack



that the architectural design of client-side model reflects the attack's performance. Particularly, even the most powerful Black-Box feature reconstruction attacks fail when attempting to compromise client's data when its architecture is MLP. We observe our findings experimentally, and provide a rigorous mathematical explanation of this phenomenon. Our study contributes to recent advances in privacy of VFL (SL) and suggest that novel Black-Box attacks should be revisited to address the challenges which occurs with MLP-based models.

We note that our approach may not be impactful on NLP tasks, since the language models require a discrete input instead of the continuous which we actively exploit during the theoretical justifications and experiments with MLP-based models. However, we note that Unsplit attack also cannot be efficiently performed against the transformer-based architectures due to the huge amount of computational resources for training multi-head attention and FFN layers with the coordinate descent.

References

Balle, B.; and Wang, Y.-X. 2018. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, 394–403. PMLR.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, 1175–1191.

Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; Papadopoulos,

D.; and Yang, Q. 2021. SecureBoost: A Lossless Federated Learning Framework. arXiv:1901.08755.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

Driouich, I.; Xu, C.; Neglia, G.; Giroire, F.; and Thomas, E. 2023. Local Model Reconstruction Attacks in Federated Learning and their Uses. arXiv:2210.16205.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, 265–284. Springer.

Dwork, C.; and Roth, A. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9: 211–407.

Défossez, A.; Bottou, L.; Bach, F.; and Usunier, N. 2022. A Simple Convergence Proof of Adam and Adagrad. arXiv:2003.02395.

Erdogan, E.; Küpçü, A.; and Cicek, A. E. 2022. Splitguard: Detecting and mitigating training-hijacking attacks in split learning. In *Proceedings of the 21st Workshop on Privacy in the Electronic Society*, 125–137.

Erdogan, E.; Teksen, U.; Celiktenyildiz, M. S.; Kupcu, A.; and Cicek, A. E. 2023. SplitOut: Out-of-the-Box Training-Hijacking Detection in Split Learning via Outlier Detection. arXiv:2302.08618.

Erdoğan, E.; Küpçü, A.; and Çiçek, A. E. 2022. UnSplit: Data-Oblivious Model Inversion, Model Stealing, and Label Inference Attacks against Split Learning. In *Proceedings of the 21st Workshop on Privacy in the Electronic Society*, CCS '22. ACM.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

Fu, J.; Ma, X.; Zhu, B. B.; Hu, P.; Zhao, R.; Jia, Y.; Xu, P.; Jin, H.; and Zhang, D. 2023. Focusing on Pinocchio's Nose: A Gradients Scrutinizer to Thwart Split-Learning Hijacking Attacks Using Intrinsic Attributes. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society.

Ganju, K.; Wang, Q.; Yang, W.; Gunter, C.; and Borisov, N. 2018. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations*, 619–633.

Gao, X.; and Zhang, L. 2023. PCAT: Functionality and Data Stealing from Split Learning by Pseudo-Client Attack. In *32nd USENIX Security Symposium (USENIX Security 23)*, 5271–5288. Anaheim, CA: USENIX Association. ISBN 978-1-939133-37-3.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting Gradients – How easy is it to break privacy in federated learning? arXiv:2003.14053.

Ghazi, B.; Golowich, N.; Kumar, R.; Manurangsi, P.; and Zhang, C. 2021. Deep Learning with Label Differential Privacy. arXiv:2102.06062.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. arXiv:1406.2661.

Graves, A. 2014. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850.

Gu, B.; Dang, Z.; Li, X.; and Huang, H. 2020. Federated Doubly Stochastic Kernel Learning for Vertically Partitioned Data. arXiv:2008.06197.

Gu, H.; Luo, J.; Kang, Y.; Fan, L.; and Yang, Q. 2023. FedPass: Privacy-Preserving Vertical Federated Deep Learning with Adaptive Obfuscation. arXiv:2301.12623.

Gupta, O.; and Raskar, R. 2018. Distributed learning of deep neural network over multiple agents. arXiv:1810.06060.

Gupta, S.; Huang, Y.; Zhong, Z.; Gao, T.; Li, K.; and Chen, D. 2022. Recovering Private Text in Federated Learning of Language Models. arXiv:2205.08514.

Hardy, S.; Henecka, W.; Ivey-Law, H.; Nock, R.; Patrini, G.; Smith, G.; and Thorne, B. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv:1711.10677.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

He, Z.; Zhang, T.; and Lee, R. B. 2019. Model inversion attacks against collaborative inference. *Proceedings of the 35th Annual Computer Security Applications Conference*.

He, Z.; Zhang, T.; and Lee, R. B. 2021. Attacking and Protecting Data Privacy in Edge–Cloud Collaborative Inference Systems. *IEEE Internet of Things Journal*, 8: 9706–9716.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Neural Information Processing Systems*.

Hu, Y.; Cai, T.; Shan, J.; Tang, S.; Cai, C.; Song, E.; Li, B.; and Song, D. 2022. Is Vertical Logistic Regression Privacy-Preserving? A Comprehensive Privacy Analysis and Beyond. arXiv:2207.09087.

Huynh-Thu, Q.; and Ghanbari, M. 2008. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44: 800–801.

Jin, X.; Chen, P.-Y.; Hsu, C.-Y.; Yu, C.-M.; and Chen, T. 2022. CAFE: Catastrophic Data Leakage in Vertical Federated Learning. arXiv:2110.15122.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konečný, J.; Korolova, A.;

Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Raykova, M.; Qi, H.; Ramage, D.; Raskar, R.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. arXiv:1912.04977.

Kariyappa, S.; and Qureshi, M. K. 2022. ExPLoit: Extracting Private Labels in Split Learning. arXiv:2112.01299.

Khan, A.; ten Thij, M.; and Wilbik, A. 2023. Vertical Federated Learning: A Structured Literature Review. arXiv:2212.00622.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Konečný, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2017. Federated Learning: Strategies for Improving Communication Efficiency. arXiv:1610.05492.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. In *Learning Multiple Layers of Features from Tiny Images*.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, J.; Rakin, A. S.; Chen, X.; He, Z.; Fan, D.; and Chakrabarti, C. 2022a. ResSFL: A Resistance Transfer Framework for Defending Model Inversion Attack in Split Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10194–10202.

Li, J.; Rakin, A. S.; Chen, X.; Yang, L.; He, Z.; Fan, D.; and Chakrabarti, C. 2023. Model Extraction Attacks on Split Federated Learning. arXiv:2303.08581.

Li, O.; Sun, J.; Yang, X.; Gao, W.; Zhang, H.; Xie, J.; Smith, V.; and Wang, C. 2022b. Label Leakage and Protection in Two-party Split Learning. arXiv:2102.08504.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. arXiv:1812.06127.

Li, Z.; Bao, H.; Zhang, X.; and Richtárik, P. 2021. PAGE: A Simple and Optimal Probabilistic Gradient Estimator for Nonconvex Optimization. arXiv:2008.10898.

Li, Z.; Wang, T.; and Li, N. 2022. Differentially Private Vertical Federated Clustering. *Proc. VLDB Endow.*, 16: 1277–1290.

Liu, J.; Lyu, X.; Cui, Q.; and Tao, X. 2024a. Similarity-Based Label Inference Attack Against Training and Inference of Split Learning. *IEEE Transactions on Information Forensics and Security*, 19: 2881–2895.

Liu, Y.; Kang, Y.; Zou, T.; Pu, Y.; He, Y.; Ye, X.; Ouyang, Y.; Zhang, Y.-Q.; and Yang, Q. 2024b. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 1–20.

Liu, Y.; Zou, T.; Kang, Y.; Liu, W.; He, Y.; Yi, Z.; and Yang, Q. 2022. Batch Label Inference and Replacement Attacks in Black-Boxed Vertical Federated Learning. arXiv:2112.05409.

Luo, X.; Wu, Y.; Xiao, X.; and Ooi, B. C. 2021. Feature Inference Attack on Model Predictions in Vertical Federated Learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 181–192.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629.

Mi, Y.; Liu, H.; Xia, Y.; Sun, Y.; Guan, J.; and Zhou, S. 2023. Flexible Differentially Private Vertical Federated Learning with Adaptive Feature Embeddings. *ArXiv*, abs/2308.02362.

Nguyen, N.-B.; Chandrasegaran, K.; Abdollahzadeh, M.; and Cheung, N.-M. 2023a. Label-Only Model Inversion Attacks via Knowledge Transfer. arXiv:2310.19342.

Nguyen, N.-B.; Chandrasegaran, K.; Abdollahzadeh, M.; and Cheung, N.-M. 2023b. Re-thinking Model Inversion Attacks Against Deep Neural Networks. arXiv:2304.01669.

Nguyen, T. D.; Nguyen, T.; Nguyen, P. L.; Pham, H. H.; Doan, K.; and Wong, K.-S. 2023c. Backdoor Attacks and Defenses in Federated Learning: Survey, Challenges and Future Research Directions. arXiv:2303.02213.

Pasquini, D.; Ateniese, G.; and Bernaschi, M. 2021. Unleashing the Tiger: Inference Attacks on Split Learning. arXiv:2012.02670.

Pedersen, M.; and Hardeberg, J. Y. 2012. *Full-Reference Image Quality Metrics*. 10.1561/0600000037.

Qiu, X.; Leontiadis, I.; Melis, L.; Sablayrolles, A.; and Stock, P. 2024a. Evaluating Privacy Leakage in Split Learning. arXiv:2305.12997.

Qiu, X.; Pan, H.; Zhao, W.; Gao, Y.; Gusmao, P. P. B.; Shen, W. F.; Ma, C.; and Lane, N. D. 2024b. Secure Vertical Federated Learning Under Unreliable Connectivity. arXiv:2305.16794.

Rodríguez-Barroso, N.; Jiménez-López, D.; Luzón, M. V.; Herrera, F.; and Martínez-Cámara, E. 2023. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90: 148–173.

Rudin, L. I.; Osher, S.; and Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D Nonlinear Phenomena*, 60(1-4): 259–268.

Sadiev, A.; Beznosikov, A.; Almansoori, A. J.; Kamzolov, D.; Tappenden, R.; and Takáč, M. 2024. Stochastic Gradient Methods with Preconditioned Updates. *Journal of Optimization Theory and Applications*, 201(2): 471–489.

Sara, U.; Akter, M.; and Uddin, M. S. 2019. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks against Machine Learning Models. arXiv:1610.05820.

Smith, A. D.; Song, S.; and Thakurta, A. 2020. The Flajolet-Martin Sketch Itself Preserves Differential Privacy: Private Counting with Minimal Space. In *Neural Information Processing Systems*.

Sun, J.; Yang, X.; Yao, Y.; and Wang, C. 2022. Label Leakage and Protection from Forward Embedding in Vertical Federated Learning. arXiv:2203.01451.

Sun, J.; Yao, Y.; Gao, W.; Xie, J.; and Wang, C. 2021. Defending against Reconstruction Attack in Vertical Federated Learning. arXiv:2107.09898.

Sun, X.; Gazagnadou, N.; Sharma, V.; Lyu, L.; Li, H.; and Zheng, L. 2023. Privacy Assessment on Reconstructed Images: Are Existing Evaluation Metrics Faithful to Human Perception? arXiv:2309.13038.

Thapa, C.; Chamikara, M. A. P.; Camtepe, S.; and Sun, L. 2022. SplitFed: When Federated Learning Meets Split Learning. arXiv:2004.12088.

Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; Lucic, M.; and Dosovitskiy, A. 2021. MLP-Mixer: An all-MLP Architecture for Vision. arXiv:2105.01601.

Turina, V.; Zhang, Z.; Esposito, F.; and Matta, I. 2021. Federated or Split? A Performance and Privacy Analysis of Hybrid Split and Federated Learning Architectures. In *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, 250–260.

Vepakomma, P.; Singh, A.; Gupta, O.; and Raskar, R. 2020. NoPeek: Information leakage reduction to share activations in distributed deep learning. arXiv:2008.09161.

Wang, R.; Ersoy, O.; Zhu, H.; Jin, Y.; and Liang, K. 2022. FEVERLESS: Fast and Secure Vertical Federated Learning based on XGBoost for Decentralized Labels. *IEEE Transactions on Big Data*, 1–15.

Wang, Y.; Sun, T.; Li, S.; Yuan, X.; Ni, W.; Hossain, E.; and Poor, H. V. 2023. Adversarial Attacks and Defenses in Machine Learning-Powered Networks: A Contemporary Survey. arXiv:2303.06302.

Wang, Z.; and Bovik, A. C. 2002. A universal image quality index. *IEEE Signal Processing Letters*, 9: 81–84.

Wei, K.; Li, J.; Ma, C.; Ding, M.; Wei, S.; Wu, F.; Chen, G.; and Ranbaduge, T. 2022. Vertical Federated Learning: Challenges, Methodologies and Experiments. arXiv:2202.04309.

Wright, S. J. 2015. Coordinate Descent Algorithms. arXiv:1502.04759.

Wu, X.; Fredrikson, M.; Jha, S.; and Naughton, J. F. 2016. A Methodology for Formalizing Model-Inversion Attacks. *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, 355–370.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747.

Yang, L.; Chai, D.; Zhang, J.; Jin, Y.; Wang, L.; Liu, H.; Tian, H.; Xu, Q.; and Chen, K. 2023. A Survey on Vertical Federated Learning: From a Layered Perspective. arXiv:2304.01829.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. arXiv:1902.04885.

Yang, Z.; Chang, E.-C.; and Liang, Z. 2019. Adversarial Neural Network Inversion via Auxiliary Knowledge Alignment. arXiv:1902.08552.

Ye, P.; Jiang, Z.; Wang, W.; Li, B.; and Li, B. 2022. Feature Reconstruction Attacks and Countermeasures of DNN training in Vertical Federated Learning. arXiv:2210.06771.

Yu, F.; Wang, L.; Zeng, B.; Zhao, K.; Pang, Z.; and Wu, T. 2023. How to backdoor split learning. *Neural Netw.*, 168(C): 326–336.

Yu, L.; Han, M.; Li, Y.; Lin, C.; Zhang, Y.; Zhang, M.; Liu, Y.; Weng, H.; Jeon, Y.; Chow, K.-H.; and Patterson, S. 2024. A Survey of Privacy Threats and Defense in Vertical Federated Learning: From Model Life Cycle Perspective. arXiv:2402.03688.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. arXiv:1801.03924.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2020. iDLG: Improved Deep Leakage from Gradients. arXiv:2001.02610.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep Leakage from Gradients. arXiv:1906.08935.

Zou, T.; Liu, Y.; Kang, Y.; Liu, W.; He, Y.; Yi, Z.; Yang, Q.; and Zhang, Y.-Q. 2022. Defending Batch-Level Label Inference and Replacement Attacks in Vertical Federated Learning. *IEEE Transactions on Big Data*, 1–12.

Zou, T.; Liu, Y.; and Zhang, Y.-Q. 2023. Mutual Information Regularization for Vertical Federated Learning. arXiv:2301.01142.