# A Stealthy Wrongdoer: Feature-Oriented Reconstruction Attack against Split Learning

Xiaoyang Xu[1] Mengda Yang[1] Wenzhe Yi [1] Ziang Li [1] Juan Wang[1*] Hongxin Hu [2]
Yong Zhuang [1] Yaxin Liu [1]
[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University
[2] Department of Computer Science and Engineering, University at Buffalo, SUNY
{xiaoyangx, mengday, wenzhey, ziangli, yong.zhuang, yaxin.liu}@whu.edu.cn
jwang@whu.edu.cn, hongxinh@buffalo.edu

## Abstract

*Split Learning (SL) is a distributed learning framework renowned for its privacy-preserving features and minimal computational requirements. Previous research consistently highlights the potential privacy breaches in SL systems by server adversaries reconstructing training data. However, these studies often rely on strong assumptions or compromise system utility to enhance attack performance. This paper introduces a new semi-honest Data Reconstruction Attack on SL, named Feature-Oriented Reconstruction Attack (FORA). In contrast to prior works, FORA relies on limited prior knowledge, specifically that the server utilizes auxiliary samples from the public without knowing any client's private information. This allows FORA to conduct the attack stealthily and achieve robust performance. The key vulnerability exploited by FORA is the revelation of the model representation preference in the smashed data output by victim client. FORA constructs a substitute client through feature-level transfer learning, aiming to closely mimic the victim client's representation preference. Leveraging this substitute client, the server trains the attack model to effectively reconstruct private data. Extensive experiments showcase FORA's superior performance compared to state-of-the-art methods. Furthermore, the paper systematically evaluates the proposed method's applicability across diverse settings and advanced defense strategies.*

## 1. Introduction

Deep Neural Networks (DNN) have gained widespread usage in computer vision due to their excellent learning ability and expressive power. Split Learning (SL) [2, 11, 16, 32, 38, 42, 44] emerged as a distributed collaborative framework that enables clients to cooperate with a server to perform learning task. In SL, the complete DNN model is divided into two parts, which are deployed on the client and server respectively. For a normal training process in SL, the client performs the computational process locally and communicates with the server solely based on intermediate features (referred to as smashed data) and their corresponding gradients. In this case, the server does not have access to any private information (raw data, parameters, architecture) about the client. Therefore, SL is considered effective in protecting the privacy of clients.

However, recent works [6, 10, 19, 31, 36] have shown that there are still privacy risks associated with SL. It is possible for the server to steal private information about the client according to auxiliary knowledge. One particular concern is the Data Reconstruction Attack (DRA) [6, 10, 31], where a server attempts to recover the training data of a client in SL systems. Depending on whether the server affects the normal process of SL, we can categorize adversaries into malicious and semi-honest attackers. Malicious servers such as FSHA [31] can manipulate the SL training process to conduct more effective attack. However, the latest findings [5, 8] show that FSHA's mischief is easily detected by the client, leading to the termination of SL training protocol For semi-honest attackers, *e.g.* PCAT [10] and UnSplit [6], their superior camouflage makes them less likely to be detected. But current semi-honest attackers often rely overly on assumptions that favor their performances. For example, UnSplit requires knowledge of the client's architecture and is only applicable to simple networks or datasets. As for PCAT, it unduly depends on the availability of partial private data to assist in training the pseudo-client. These assumptions contradict the basic principle of SL, which is to ensure that the client's knowledge

[*] Corresponding author.

remains hidden from the server. In summary, we find previous attacks lack consideration of the intrinsic security of SL and the plausibility of their attack hypothesis, which limits the effectiveness and threat of their approach in real-world SL systems scenarios.

In this work, we introduce a novel DRA toward more realistic and more challenging scenarios, where the server cannot access private data or the structures and parameters of the client model. Our scheme stems from new insights into potential privacy breaches in SL. We discover a fundamental phenomenon that the client model has its own *representation preference*, which can be reflected through the output smashed data. More importantly, this unique information can indicate the feature extraction behavior of the client. Based on this new insight, we propose a semi-honest privacy threat, namely Feature-Oriented Reconstruction Attack (FORA). A server adversary could establish a substitute client by narrowing the reference distance with the real client, which allows the substitute model to mimic the behavior of the target model at a finer granularity. To efficiently measure the preference distance of different representations, we introduce domain Discriminator network [9, 14] and Multi-Kernel Maximum Mean Discrepancy (MK-MMD) [15, 29]. These techniques are widely used in domain adaptation [45], enabling us to project various representation preferences into a shared space for comparison. With a well-trained substitute client, the server can successfully recover the private data by constructing an inverse network.

We conduct our evaluation on two benchmark datasets and corresponding networks against different model partitioning strategies. The experimental results indicate that the proposed method significantly outperforms baseline attacks. Taking the reconstructed images of CelebA at layer 2 as an example, UnSplit, PCAT and FORA achieve effects of 8.70, 12.05, and 17.11 on the PSNR [20]. This demonstrates that FORA has significantly outperformed by 1.97x and 1.42x compared to the other two attacks. Although FSHA can achieve attack performance similar to ours, its malicious attack process can be promptly halted through monitoring mechanisms [8], resulting in poor reconstructions. Furthermore, we investigate the potential influences on FORA, including different public knowledge conditions and existing defense strategies, to validate the robustness of FORA.

The main contributions of this paper can be summarized as follows:

- We propose a novel attack, named Feature-Oriented Reconstruction Attack (FORA). As far as we know, FORA is the first work enabling a semi-honest server to perform powerful DRA in more realistic and challenging SL systems. In such scenarios, the server has no prior knowledge of the client model or access to raw data.

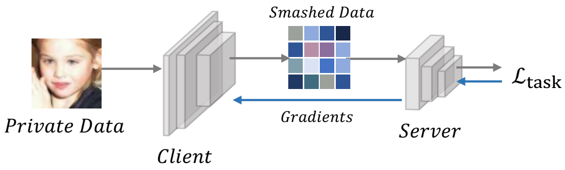


Figure 1. Architecture of two-part split learning.

- We have uncovered an inherent vulnerability in SL, where the server can exploit rich information in the smashed data to steal client representation preference, thereby building a substitute client for better reconstruction.
- We conduct comprehensive experiments with various adversarial knowledge against different benchmark datasets and models. The results demonstrate that FORA can achieve state-of-the-art attack performance compared with baselines and exhibits notable robustness across different settings.

## 2. Background and Related Work

**Split Learning (SL).** SL [2, 16, 32, 38, 42] is an emerging distributed learning paradigm for resource-limited scenarios, which can split the neural network model into both client-side and server-side. As shown in Fig. 1, the client performs forward propagation and transmits the smashed data to the server, which then uses the computed loss for backward propagation and sends the gradients of the smashed data back to the client. Both the client and server will update their weights after receiving the gradients. It is generally believed that SL provides a secure and efficient training protocol by allowing the client to retain a portion of the model and training data locally while offloading most of the computing overhead to the server [2, 16, 32, 42]. However, recent studies [6, 7, 10, 23, 31] have highlighted vulnerabilities in SL, where the server can exploit the latter part of the model to carry out privacy attacks.

**Data Reconstruction Attack (DRA) on SL.** DRA [19, 27, 35, 48] is one of the most powerful privacy attacks that aim to steal the input data by the model's intermediate features. In SL, the server can utilize the smashed data output by the client to reconstruct the training data [6, 10, 31]. One notable attack is known as FSHA [31], where a malicious attacker utilizes the elaborated loss to alter the feature space of the victim client for reconstructing private data. In UnSplit [6], the semi-honest server attempts to reconstruct the training data and client's parameters simultaneously by utilizing the smashed data. Specifically, UnSplit optimizes parameters and inputs sequentially by minimizing the outputs between the clone client and the target client. To the best of our knowledge, PCAT [10] represents the most advanced attack under the semi-honest assumption. PCAT leverages the knowledge embedded in various stages of the server models

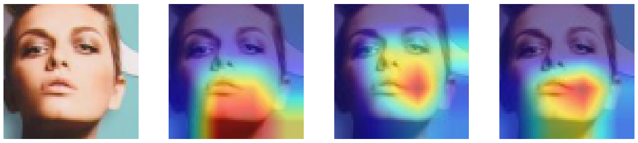


Figure 2. Input image and behavior visualization by Grad-CAM [33]. All the models are trained in CelebA with the task of smiling classification. The figure displays the original images and the representation preferences of three models trained under the same hyperparameter settings but with different random seeds.

to steal private data by constructing a pseudo-client. Unlike previous work, SFA [30] focuses on reconstructing samples during the inference stage rather than the training samples.

Although existing works claim that their attacks pose significant privacy threats to SL, they disregard the plausibility of their threat model. For FSHA, the server reconstructs the raw data while at the cost of destroying the client's utility. While FSHA assumes that the client is entirely free of any awareness of being maliciously disrupted, recent research [5, 8] indicates that such a malicious server can be easily detected by the client, leading to a halt in the SL. UnSplit needs the knowledge of the client's structure and is not suitable for complex networks and datasets due to the infinite searching space of input data and model parameters. As for PCAT, it requires the adversary to have access to a portion of the private dataset. This is an unreasonable assumption that violates the original intention of SL since one of the distinctive characteristics of SL is the ability to train models without sharing the raw data [42]. As a result, how to explore DRA under more realistic assumptions in SL remains an open question.

**Domain Adaptation.** Domain adaptation [9, 12, 15, 29, 40, 41, 46] is a technique that seeks to enhance the generalization of a model by transferring knowledge acquired from a source domain to a distinct yet related target domain. The core idea of domain adaptation is to map data from different domains into the same space for comparison. Here, we apply two popular methods: the domain Discriminator network [9, 41, 46] and the Multi-Kernel Maximum Mean Discrepancy (MK-MMD) function [15, 29, 46] to compare the feature spaces of different models.

# 3. Method

## 3.1. Threat Model

Without loss of generality, given a two-party SL protocol, the SL model $F$ is partitioned to a server model $F_s$ and a client model $F_c$. The server aims to stealthily recover the private training data of the client through the smashed data $Z$ output by $F_c$.

We assume that the server adversary is a semi-honest entity, ensuring that the training process is indistinguishable from ordinary training during attack. Furthermore, we posit that the server adversary must adhere to the foundational principle of the SL — she lacks any means of accessing client-sensitive information. Specifically, the server does not require knowledge of the structure or hyperparameters of $F_c$ and is devoid of access to the client's private training dataset $D_{priv}$. The sole piece of public knowledge available to the server pertains to the auxiliary dataset $D_{aux}$, sourced from the same domain as the private samples. It's important to note that the distribution of $D_{aux}$ typically differs from that of $D_{priv}$. Compared to the threat model of previous works, this assumption is more reasonable and realistic.

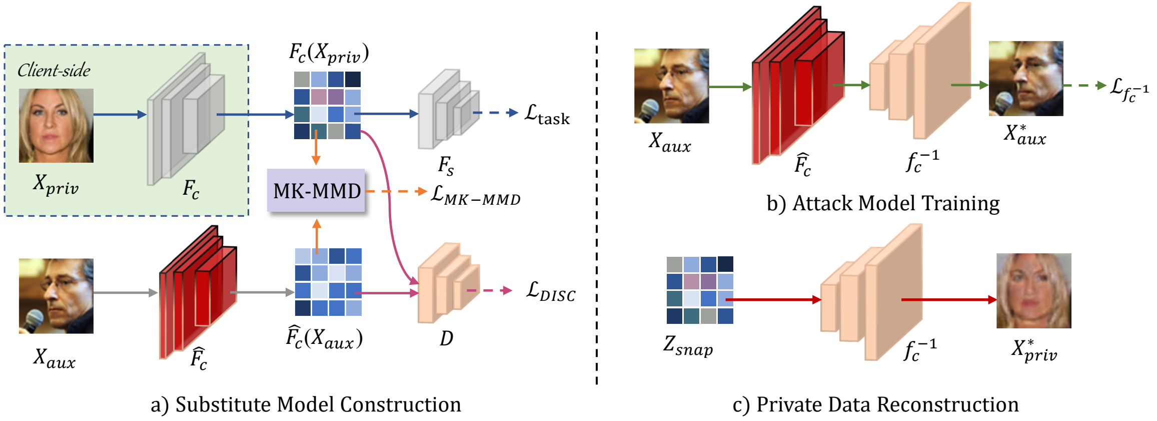
## 3.2. Motivation

Current DRAs rely overly on constructing inverse networks from input-output pairs obtained by querying the target model. However, this approach is impractical for SL because the server only has access to the client's outputs and is not qualified to query. A potential solution is to build a substitute client to mimic the target client, thus enabling the training of the inverse network. However, the variability of the substitute client's behavior affects the generalization of the inverse network to the target client, leading to the failure of the reconstruction, especially without the knowledge of the client model structure and private data distribution.

As illustrated in Fig. 2, we employ Grad-CAM [33] to visualize the attention of intermediate features generated by different clients. From Fig. 2 (a)-(d), it can be noticed that even for models trained under the same setup, there still exists evident differences between their image processing attention. This phenomenon suggests that the smashed data output by the client reflects its distinctive feature extraction behavior, which we define as representation preferences. Our general assumption is that narrowing the gap between the substitute client and the target client in terms of intermediate features can make the representation preferences of the two models more similar, which ensures that the inverse network trained by the substitute client perfectly maps the target smashed data back to the private raw data.

## 3.3. Feature-Oriented Reconstruction Attack

Inspired by the differences in model representation preferences, we propose a novel data reconstruction attack against SL, called Feature-Oriented Reconstruction Attack (FORA). In order to mount FORA, the adversary needs to contrive a way to obtain the representation preferences of the $F_c$. To address this problem, we utilize domain adaptation techniques [9, 15, 29] to project different preference representations into the same space. Specifically, the adversary conducts feature-level transfer learning by exploiting the $Z_c$ collected in each training iteration and then obtains a substitute model that mimics well the feature extraction be-



a) Substitute Model Construction

b) Attack Model Training

c) Private Data Reconstruction

Figure 3. Attack pipeline of Feature-Oriented Reconstruction Attack (FORA) against SL. (a) shows the substitute model training phase. The attacker constructs a substitute model $\hat{F}_c$ using $\mathcal{L}_{DISC}$ and $\mathcal{L}_{MK-MMD}$ to mimic the behavior of the client model $F_c$. (b) means training an inverse network $f_c^{-1}$ using public data $X_{aux}$. (c) represents the final attack phase using the attack model to reconstruct training data from snapshot $Z_{snap}$ of target smashed data.

havior of the $F_c$. Through this approach, the adversary can smoothly construct an attack model (inverse mapping network) to recover the private samples. The detailed pipeline of FORA is shown in Fig. 3. It consists of three phases: substitute model construction, attack model training, and private data reconstruction.

**Substitute Model Construction.** Before SL training commences, the server initializes a substitute client, denoted by $\hat{F}_c$. The $\hat{F}_c$ will be trained locally at the server in parallel with the victim's $F_c$, and such process will take place throughout the entire SL collaboration. In each training iteration, the client will send smashed data of the current batch to the server for completing the subsequent computations. Concurrently, the server will use the collected smashed data to perform training on the $\hat{F}_c$. For this purpose, the server introduces the Discriminator module and the MK-MMD module to extract the representation preferences. We define its training objective as:

$$\min_{\hat{F}_c} \mathcal{L}_{DISC} + \mathcal{L}_{MK-MMD}, \tag{1}$$

where $\mathcal{L}_{DISC}$ is the Discriminator module constraining $Z_{aux} = \hat{F}_c(X_{aux})$ and $Z_{priv} = F_c(X_{priv})$ to be indistinguishable, while $\mathcal{L}_{MK-MMD}$ is the MK-MMD module making $Z_{aux}$ as close as possible to $Z_{priv}$ in shared space.

The Discriminator [3, 9, 13] $D$ is also a network that needs to be trained synchronously and is tasked with efficiently distinguishing the generated features between $F_c$ and $\hat{F}_c$, maximizing probabilities of the former and minimizing probabilities of the latter [31]. Therefore, the parameters of $D$ will be updated to minimize the following loss function:

$$\mathcal{L}_D = \log\left(1 - \mathcal{D}(F_c(X_{priv})\right) + \log \mathcal{D}(\hat{F}_c(X_{aux})). \tag{2}$$

After each local training step of $D$, the server utilizes $D$ to instruct substitute client's representation preference to be consistent with that of the victim client. Specifically, an adversarial loss is constructed as the following:

$$\mathcal{L}_{DISC} = \log\left(1 - D(\hat{F}_c(X_{aux}))\right). \tag{3}$$

The MK-MMD module [15, 29] is designed to align two sets of generated features into a shared space using kernel functions and compute their difference, where a smaller difference signifies closer representation preferences. Then, for the substitute client, the objective extends beyond maximizing the probabilities output by the $D$, it also seeks to minimize the MK-MMD loss function, namely:

$$\mathcal{L}_{MK-MMD} = \left\| \phi\left(\hat{F}_c(X_{aux})\right) - \phi\left(F_c(X_{priv})\right) \right\|_{\mathcal{H}}, \tag{4}$$

$$\begin{cases} \phi = \sum_{j=1}^{m} \beta_j k_j, \\ \sum_{j=1}^{m} \beta_j = 1, \beta_j \geq 0, \forall j, \end{cases} \tag{5}$$

where $k$ is a single kernel function, $\phi$ denotes a set of kernel functions that project different smashed data into Reproducing Kernel Hilbert Space $\mathcal{H}$, $\beta$ is the weight coefficient corresponding to the single kernel function.

**Attack Model Training.** At the end of the training of SL, the server can obtain a substitute client with a feature extraction behavior extremely similar to that of the victim client. Moreover, its feature space is known to the adversary, who can recover the original input from the smashed data by applying an inverse network (denoted as $f_c^{-1}$). Following previous DRAs [19, 35], we adopt the $f_c^{-1}$ consisting of a set of Transposed Convolution layers and Tanh activations as our attack model. The server can leverage the auxiliary dataset to train the attack model by minimizing the mean square error between $f_c^{-1}(\hat{F}_c(X_{aux}))$ and $X_{aux}$ as follows:

$$\mathcal{L}_{f_c^{-1}} = \|f_c^{-1}(\hat{F}_c(X_{aux})) - X_{aux}\|_2^2. \tag{6}$$

**Private Data Reconstruction.** The server keeps a snapshot $Z_{snap} = F_c(X_{priv})$ of all smashed data output by the target client under the final training iteration for reconstruction. Since the substitute client is able to mimic the target client's representation preferences well, the server can subtly use $f_c^{-1}$ to perform the attack by mapping the target smashed data directly into the private raw data space, namely:

$$X_{priv}^* = f_c^{-1}(Z_{snap}). \tag{7}$$

Here, $X_{priv}^*$ are the reconstructed private training samples.

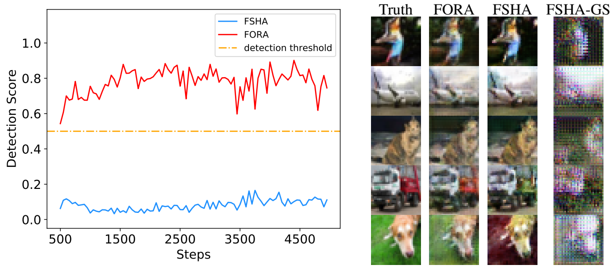
# 4. Experiments

## 4.1. Experimental Setup

**Datasets.** In our experiments, we rely on CIFAR-10 [26] and CelebA [28] to validate the attacks, due to their dominance in the research on SL [6, 10, 31]. They will be used as private data for the client's target training tasks. According to Sec. 3.1, we assume that the server adversary has access to a set of auxiliary samples that are distinct from the client's private data. Therefore, we choose CINIC-10 [4] and FFHQ [24] as the adversary's auxiliary dataset, respectively. We exclude images in CINIC-10 that overlapped with CIFAR-10, and randomly select 5,000 samples and 10,000 samples from the preprocessed CINIC-10 and FFHQ as the final auxiliary data. Appendix A.1 provides the detailed information for different datasets.

**Models.** We consider two popular types of neural network architectures, including MobileNet [21] and ResNet-18 [17], as target models for the classification tasks of CIFRA-10 and CelebA, respectively. We set various split points for different target models to show our attack performance. Since the server is entirely unaware of the client's model structure from Sec. 3.1, we use VGG blocks [34] (consisting of a sequence of Convolutional, BatchNorm, ReLU, and MaxPool layers) to construct substitute models. In addition, the adversary's substitute models adaptively depend on the size of the intermediate features output by the client. All the architecture information and splitting schemes used in this paper are reported in Appendix A.2.

**Metrics.** In addition to analyzing the qualitative results of attack performances visually, we chose three quantitative metrics to evaluate the quality of the reconstructed images: Structural Similarity (SSIM) [47], Peak Signal-to-Noise Ratio (PSNR) [20], and Learned Perceptual Image Patch Similarity (LPIPS) [49]. We also use Cosine Similarity and Mean Square Error to measure the similarity between the substitute client and the target client in feature space.



(a) Detection Score

(b) Reconstruction Results

Figure 4. Attack performance comparison of FSHA [31] and FORA on CIFAR-10 with layer 2. (a) shows the detection score of two attacks detected by GS. (b) represents the reconstruction results of two attacks, and FSHA-GS is the reconstructed images when detected by GS.

**Attack Baselines.** We mainly compare our approach with three representative existing methods, which are FSHA [31], UnSplit [6], and PCAT [10]. For the malicious attack FSHA, we use sophisticated detection mechanism to jointly evaluate the attack's effectiveness. For the semi-honest attack UnSplit, we make it consistent with our experimental settings to ensure fairness. PCAT requires an understanding of the learning task while relying on a subset of the private training data to build the pseudo-client, and in order to comply with this assumption, we set the proportion of the CIFAR-10 private dataset to be 5% (the maximal threshold suggested by the original paper), and for more complex CelebA dataset, we extend the proportion to be 10%.

## 4.2. Comparison with Malicious Attack

Since FSHA severely undermines the utility of the target client, recent work has proposed the Gradients Scrutinizer (GS) [8] to defend against such hijacking attacks by detecting the gradients returned from the server to the client. The GS will perform a similarity computation on the gradients, and if the calculated value is lower than a set threshold, it will be considered as a potential attack, resulting in the training of SL being immediately suspended. More details about GS can be found in Appendix C.1. We can observe from Fig. 4 that the reconstruction results of FORA are almost the same as those of FSHA in the unprotected SL system. Although FSHA performs well in capturing fine graphical details, it also leads to noticeable color shifts in some reconstruction results. Moreover, since FSHA drastically tampers with the updated gradient returned to the client model, it is easily detected by GS, leading to the failure of reconstruction.

Table 1. Data reconstruction results of UnSplit, PCAT, and FORA on CIFAR-10 and CelebA in different splitting settings.

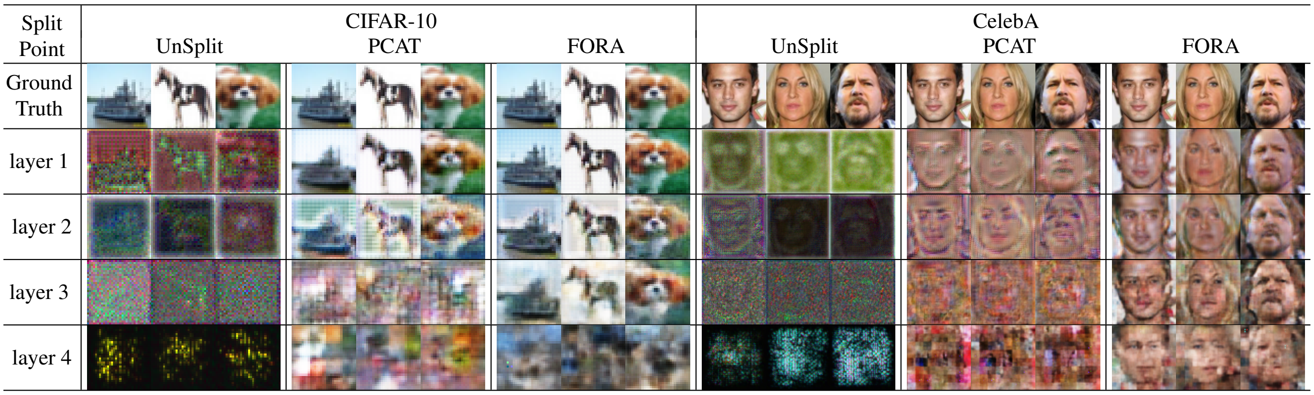


Table 2. SSIM, PSNR, and LPIPS of the reconstructed images on CIFAR-10 among three attacks.

| Split | SSIM↑ | | | PSNR↑ | | | LPIPS↓ | | |
|---|---|---|---|---|---|---|---|---|---|
| Point | UnSplit | PCAT | FORA | UnSplit | PCAT | FORA | UnSplit | PCAT | FORA |
| layer 1 | 0.171 | 0.853 | **0.926** | 11.03 | 22.10 | **25.87** | 0.677 | 0.219 | **0.120** |
| layer 2 | 0.101 | 0.642 | **0.830** | 10.48 | 17.29 | **22.19** | 0.689 | 0.432 | **0.252** |
| layer 3 | 0.104 | 0.291 | **0.622** | 11.14 | 13.18 | **18.93** | 0.741 | 0.615 | **0.381** |
| layer 4 | 0.108 | **0.121** | 0.030 | 8.62 | **11.08** | 10.45 | 0.722 | 0.676 | **0.628** |

Table 3. SSIM, PSNR and LPIPS of the reconstructed images on CelebA among three attacks.

| Split | SSIM↑ | | | PSNR↑ | | | LPIPS↓ | | |
|---|---|---|---|---|---|---|---|---|---|
| Point | UnSplit | PCAT | FORA | UnSplit | PCAT | FORA | UnSplit | PCAT | FORA |
| layer 1 | 0.137 | 0.333 | **0.485** | 9.26 | 13.45 | **17.72** | 0.804 | 0.634 | **0.320** |
| layer 2 | 0.170 | 0.316 | **0.476** | 8.70 | 12.05 | **17.11** | 0.747 | 0.653 | **0.381** |
| layer 3 | 0.156 | 0.164 | **0.191** | 10.66 | 11.63 | **14.19** | 0.793 | 0.731 | **0.509** |
| layer 4 | 0.084 | 0.092 | **0.192** | 7.94 | 10.60 | **13.00** | 0.804 | 0.738 | **0.621** |

Table 4. Feature similarity measured by Mean Square Error and Cosine Similarity on CIFAR-10 and CelebA at layer 2.

| Method | CIFAR-10 | | | CelebA | | |
|---|---|---|---|---|---|---|
| | UnSplit | PCAT | FORA | UnSplit | PCAT | FORA |
| Mean Square Error↓ | 1.041 | 0.528 | **0.274** | 50.773 | 1.353 | **0.753** |
| Cosine Similarity↑ | 0.200 | 0.592 | **0.810** | 0.333 | 0.480 | **0.778** |

## 4.3. Comparison with Semi-Honest Attacks

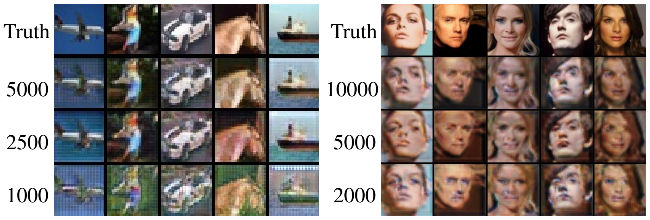
**Reconstruction Performance.** We show in detail the reconstruction results for UnSplit, PCAT, and our proposed FORA on all split points for both datasets. As depicted in Tab. 1, compared to other attacks, the images reconstructed by FORA exhibit a significant improvement visually. Due to the vast search space and inefficient optimization approach, UnSplit almost fails to recover training data in both datasets, even at layer 1. Although PCAT can reconstruct training samples in the shallow settings of the CIFAR-10 dataset, such as layer 1 and layer 2, the reconstruction quality is still lower than that of FORA. For the more complex CelebA dataset, PCAT struggles to produce quality reconstructions. Tab. 2 and Tab. 3 provides the quantitative results of the attacks. Except for the anomaly at the layer 4 split point of CIFAR-10, where FORA slightly underperforms PCAT in terms of SSIM and PSNR metric, FORA is superior to both methods in all other settings, especially in terms of the LPIPS metric, which is considered to be more aligned with human perception. Notably, even though PCAT has access to a subset of the private data, while FORA only obtains samples with different distributions, FORA substantially surpasses PCAT for reconstruction. This further emphasizes the robust privacy threat our approach poses to SL. More reconstructed images are presented in Appendix B.1.

**Feature Similarity.** As shown in Tab. 4, we measure the feature distance between the proxy clients built by UnSplit, PCAT, and FORA and the target client at layer 2. The results show that the substitute clients trained by our method exhibit more similar representation preferences to the target client. The basic optimization approach of UnSplit makes it difficult to regularize the feature space of the proxy client. As for PCAT, it simply makes the smashed data generated by the pseudo model more favorable to the server model but fails to mimic the behavior of the client model. In contrast, FORA can impose stronger constraints in the feature space, which directly contributes to successful reconstruction.

## 4.4. Effect of Auxiliary Dataset

Next, we analyze the effect of several important factors regarding the auxiliary dataset on attack performance. We first explore the impact of the fitting level of substitute models by varying the size of the auxiliary data. Then, we discuss the impact of the presence of a more significant distribution shift, i.e., the absence of some categories, between the auxiliary and target samples. Finally, we relax the major assumption about the adversary, namely that the server has access to the similarly distributed auxiliary dataset. We set the split point at layer 2 for ablation, and the full experimental results are provided in Appendix B.2.

**Auxiliary Set Size.** As shown in Fig. 5, when we reduce the size of the auxiliary dataset to half of the previous

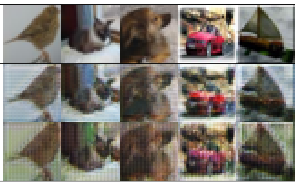


(a) CIFAR-10 (b) CelebA

Figure 5. Effects of varying auxiliary data size on FORA performed on CIFAR-10 and CelebA at layer 2.

one, the attack performance of FORA remains almost unchanged. When we further reduce the number of auxiliary samples to 20%, the quality of the reconstructed images decreases slightly but still preserves the full outline and most of the details. In that case, the percentage of the public auxiliary dataset is very small compared to the huge private training set (50,000 for CIFAR-10 and 162770 for CelebA), only 2% and 1.2%, respectively. This implies that even with a rather limited auxiliary dataset, FORA is still able to effectively reconstruct the client's training samples.

Table 5. Effect of absence of categories on FORA performed on CIFAR-10 at layer 2.

| Absent Categories | SSIM↑ | PSNR↑ | LPIPS↓ |  |
|---|---|---|---|---|
| Living | 0.768 | 20.44 | 0.300 | |
| Non-living | 0.732 | 18.43 | 0.395 | |

**Absence of Categories.** It is likely that the adversary's public auxiliary data misses some semantic classes of the private data distribution. To model this situation, we create two special auxiliary datasets for CIFAR-10, one containing "Living" items (birds, cats, etc.), and the other containing "Non-living" items (airplanes, cars, etc.), both with 5,000 randomly sampled samples from CINIC-10. As presented in Tab. 5, even if a class is absent from the auxiliary dataset, FORA can still reconstruct samples of that class. In fact, FORA focuses on stealing the mapping relationship between client inputs and smashed data and therefore does not require class alignment. We observe that the absence of the "Non-living" category leads to a moderate degradation in the reconstruction results. We believe that the reason behind this phenomenon is that the greater variation of classes within the "Non-living" category helps to increase the generalization level of the substitute client, which in turn facilitates improved attack performance.

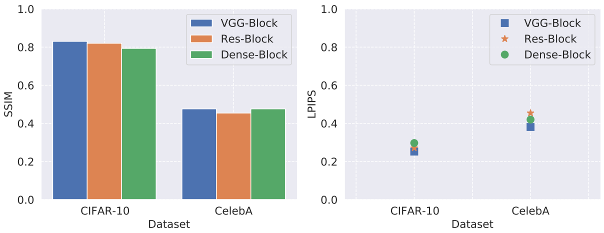
**Distribution Shift.** Here we further analyze the impact of the auxiliary dataset distribution on FORA. In contrast to our default experimental setup, we selected 5000 and 10000 images from the original testing sets of CIFAR-10 and CelebA, respectively, as the auxiliary datasets with the same distribution. As shown in Tab. 6, a more similar distribution can facilitate substitute clients stealing the representation preference, resulting in better reconstruction performance. We observe that the attack results on the facial dataset are more vulnerable to the data distribution shift compared to the object dataset. One possible reason is that tasks related to facial datasets are more sensitive to variations in sampling methods and alignment conditions across different datasets. For object datasets, due to substantial distribution variation between different categories of themselves, *e.g.* ranging from animals to vehicles, which contributes to their robustness in handling distribution shifts.

Table 6. Effects of auxiliary dataset distribution shift on FORA performed on CIFAR-10 and CelebA at layer 2. "Different" represents auxiliary data sampled from CINIC-10, and FFHQ respectively, and "Same" means auxiliary dataset come from their original test set.

| Dataset | CIFAR-10 | | CelebA | |
|---|---|---|---|---|
| Size | Different | Same | Different | Same |
| SSIM↑ | 0.830 | 0.832 | 0.476 | 0.777 |
| PSNR↑ | 22.19 | 22.78 | 17.11 | 21.55 |
| LPIPS↓ | 0.252 | 0.207 | 0.381 | 0.264 |

## 4.5. Effect of Substitute Client Structure

After validating the impact of the auxiliary dataset, here we are interested in the impact of substitute client architectures on FORA. We chose three different model structures as attack variants: the VGG block [34], the ResNet block [18], and the DenseNet block [22]. As can be seen in Fig. 6, the SSIM and LPIPS quantization results for the reconstructed images remain similar. This indicates that the extracted representation preferences on the basis of MK-MDD and Discriminator are close to that of the target client, despite the fact that the substitute clients use different architectures. Additional results are shown in Appendix B.3.



(a) SSIM↑ (b) LPIPS↓

Figure 6. Effect for FORA with varying substitute model architectures on both datasets at layer 2.

### 4.6. Counter Defense Techniques

There have been a number of defenses aimed at perturbing the smashed data claiming that they can reduce the risk of privacy leakage in SL to a certain extent. We select three well-known defense techniques, i.e., distance correlation minimization [37, 43, 44], differential privacy [1], and noise obfuscation [39], to evaluate the effectiveness of FORA. Tab. 7 shows the limited impact of these defenses on FORA. See Appendix C.1 for more details on defense techniques. See Appendix C.2 for more defense results and discussions about possible adaptive defenses.

**Distance Correlation Minimization (DCOR).** DCOR can uncorrelate irrelevant and sensitive features from the smashed data associated with the target client, which results in a lack of detailed expression of the input data in the representation preferences learned by the substitute client, especially in colors. However, FORA retains the ability to reconstruct the structural details of the private image.

**Differential Privacy (DP).** DP protects training data privacy by adding carefully crafted Laplace noise to the gradients. However, the effectiveness of DP against FORA is very limited under all privacy budgets. When the test accuracy of the model is reduced by nearly 10% (the functionality is severely damaged), the SSIM of the reconstructed samples still reaches about 75% of the original. This trade-off between classification accuracy and defense strength makes DP not feasible for practical applications of SL.

**Noise Obfuscation (NO).** NO is a direct defense to destroy the mapping relationship between smashed and input data. We observe that on the one hand, the noise of a small scale enhances the generalization level of the SL model to maintain or even improve the classification accuracy, on the other hand raising the noise scale helps to introduce deviations to the features extracted from the target client, making it more difficult to learn the representations and reconstruct the data for FORA.

## 5. Discussion and Conclusion

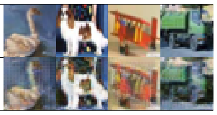
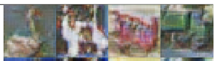
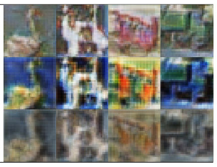
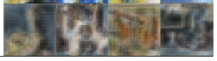
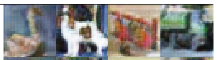
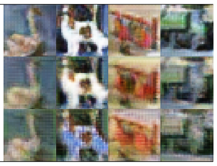
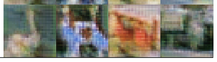
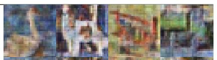
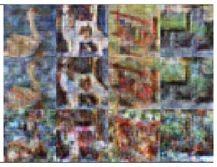
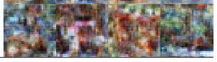
In this section, we first discuss the potential improvement and scalability of FORA, then we summarize this work. We also show limitation and future work in Appendix D.

**Improvement using Generative Adversarial Networks.** Li *et al.* [27] propose a novel StyleGAN-based reconstruction attack against split inference, and their research focus is orthogonal to our contribution. Therefore, the reconstruction task in FORA can be further optimized using pre-trained StyleGAN [25]. As shown in Fig. 7, the well-trained substitute client in FORA combined with StyleGAN optimization can provide additional improvements in reconstruction performance.

**Attack on Label-Protected SL.** Another popular setup for SL requires the client to keep the labels locally [42], but this case does not have any influence on the implementation and performance of FORA. Since FORA is only related to the smashed data output from the target client, it does not depend on the server model as well as the training task.

**Conclusion.** In this work, we propose a novel data reconstruction attack against SL, named Feature-Oriented Reconstruction Attack (FORA). Unlike all previous attack schemes, FORA enables a semi-honest server to secretly reconstruct the client's private training data with very little prior knowledge. Thanks to our new perspective of extracting representation preferences from smashed data, the server can contemporaneously train a substitute client that approximates the target client's behavior to conduct the attack. Our extensive experiments in various settings demonstrate the state-of-the-art performance of FORA. Due to its stealth and effectiveness, it poses a real privacy threat to SL. We hope our work can inspire future efforts to explore it in more practical SL, and we are eager to draw attention to more robust defense techniques.

Table 7. Effect of utility and FORA performance against three defense techniques on CIFAR-10 at layer 2.

| Defense Hyperparam | Test Acc (%) | SSIM↑ | PSNR↑ | LPIPS↓ | |
|---|---|---|---|---|---|
| | | | | |  |
| 0<br>(w/o defense) | 71.25 | 0.830 | 22.19 | 0.252 |  |
| **DCOR** ($\alpha$) | | | | | |
| 0.2 | 70.91 | 0.692 | 17.91 | 0.360 |  |
| 0.5 | 70.06 | 0.628 | 15.99 | 0.441 |  |
| 0.8 | 69.72 | 0.563 | 15.40 | 0.471 |  |
| **DP** ($\epsilon$) | | | | | |
| $+\infty$ | 69.68 | 0.823 | 22.36 | 0.225 |  |
| 100 | 63.05 | 0.711 | 20.36 | 0.394 |  |
| 10 | 61.93 | 0.621 | 18.03 | 0.487 |  |
| **NO** ($\sigma$) | | | | | |
| 1.0 | 74.39 | 0.640 | 17.29 | 0.367 |  |
| 2.0 | 73.14 | 0.583 | 16.29 | 0.444 |  |
| 5.0 | 70.62 | 0.394 | 14.35 | 0.550 |  |

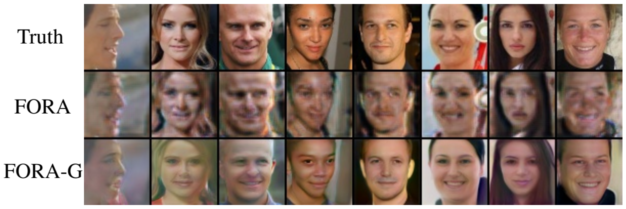


Figure 7. Reconstructed CelebA images of FORA and FORA-G, FOAR-G represents FORA combined with StyleGAN.

## References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 8

[2] Sharif Abuadbba, Kyuyeon Kim, Minki Kim, Chandra Thapa, Seyit A Camtepe, Yansong Gao, Hyoungshick Kim, and Surya Nepal. Can we use split learning on 1d cnn models for privacy preserving training? In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 305–318, 2020. 1, 2

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 4

[4] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018. 5

[5] Ege Erdogan, Alptekin Küpçü, and A Ercument Cicek. Splitguard: Detecting and mitigating training-hijacking attacks in split learning. In *Proceedings of the 21st Workshop on Privacy in the Electronic Society*, pages 125–137, 2022. 1, 3

[6] Ege Erdoğan, Alptekin Küpçü, and A Ercüment Çiçek. 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*, pages 115–124, 2022. 1, 2, 5

[7] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1397–1414, 2022. 2

[8] Jiayun Fu, Xiaojing Ma, Bin B. Zhu, Pingyi Hu, Ruixin Zhao, Yaru Jia, Peng Xu, Hai Jin, , and Dongmei Zhang. 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, 2023. 1, 2, 3, 5

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 2, 3, 4

[10] Xinben Gao and Lan Zhang. PCAT: Functionality and data stealing from split learning by Pseudo-Client attack. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5271–5288, Anaheim, CA, 2023. USENIX Association. 1, 2, 5

[11] Yansong Gao, Minki Kim, Sharif Abuadbba, Yeonjae Kim, Chandra Thapa, Kyuyeon Kim, Seyit A Camtepe, Hyoungshick Kim, and Surya Nepal. End-to-end evaluation of federated learning and split learning for internet of things. *arXiv preprint arXiv:2003.13376*, 2020. 1

[12] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. In *PRICAI 2014: Trends in Artificial Intelligence: 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia, December 1-5, 2014. Proceedings 13*, pages 898–904. Springer, 2014. 3

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 4

[14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 2672–2680, 2014. 2

[15] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. *Advances in neural information processing systems*, 25, 2012. 2, 3, 4

[16] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018. 1, 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[19] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019. 1, 2, 4

[20] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 2, 5

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 7

[23] Sanjay Kariyappa and Moinuddin K Qureshi. Exploit: Extracting private labels in split learning. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 165–175. IEEE, 2023. 2

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 5

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 8

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[27] Ziang Li, Mengda Yang, Yaxin Liu, Juan Wang, Hongxin Hu, Wenzhe Yi, and Xiaoyang Xu. GAN you see me? enhanced data reconstruction attacks against split inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 8

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 5

[29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2, 3, 4

[30] Sida Luo, Fangchao Yu, Lina Wang, Bo Zeng, Zhi Pang, and Kai Zhao. Feature sniffer: A stealthy inference attacks framework on split learning. In *International Conference on Artificial Neural Networks*, pages 66–77. Springer, 2023. 3

[31] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2113–2129, 2021. 1, 2, 4, 5

[32] Maarten G Poirot, Praneeth Vepakomma, Ken Chang, Jayashree Kalpathy-Cramer, Rajiv Gupta, and Ramesh Raskar. Split learning for collaborative deep learning in healthcare. *arXiv preprint arXiv:1912.12115*, 2019. 1, 2

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 7

[35] Abhishek Singh, Ayush Chopra, Ethan Garza, Emily Zhang, Praneeth Vepakomma, Vivek Sharma, and Ramesh Raskar. Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12125–12135, 2021. 2, 4

[36] Congzheng Song and Vitaly Shmatikov. Overlearning reveals sensitive attributes. *arXiv preprint arXiv:1905.11742*, 2019. 1

[37] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. 2007. 8

[38] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8485–8493, 2022. 1, 2

[39] Tom Titcombe, Adam J Hall, Pavlos Papadopoulos, and Daniele Romanini. Practical defences against model inversion attacks for split neural networks. *arXiv preprint arXiv:2104.05743*, 2021. 8

[40] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 3

[41] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 3

[42] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018. 1, 2, 3, 8

[43] Praneeth Vepakomma, Otkrist Gupta, Abhimanyu Dubey, and Ramesh Raskar. Reducing leakage in distributed deep learning for sensitive health data. *arXiv preprint arXiv:1812.00564*, 2, 2019. 8

[44] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 933–942. IEEE, 2020. 1, 8

[45] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomput.*, 312(C):135–153, 2018. 2

[46] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[48] Mengda Yang, Ziang Li, Juan Wang, Hongxin Hu, Ao Ren, Xiaoyang Xu, and Wenzhe Yi. Measuring data reconstruction defenses in collaborative inference systems. *Advances in Neural Information Processing Systems*, 35:12855–12867, 2022. 2

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 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, 2018. 5