Protecting Split Learning by Potential Energy Loss

Fei Zheng[1], **Chaochao Chen**[1], **Lingjuan Lyu**[2], **Xinyi Fu**[3],
Xing Fu[3], **Weiqiang Wang**[3], **Xiaolin Zheng**[1*], **Jianwei Yin**[1]
[1]College of Computer Science and Technology, Zhejiang University
[2]Sony AI
[3]Ant Group
{zfscgy2, zjuccc}@zju.edu.cn, lingjuan.lv@sony.com, fxy122992@antgroup.com,
{fux008, wang.weiqiang}@gmail.com, xlzheng@zju.edu.cn, zjuyjw@cs.zju.edu.cn

Abstract

As a practical privacy-preserving learning method, split learning has drawn much attention in academia and industry. However, its security is constantly being questioned since the intermediate results are shared during training and inference. In this paper, we focus on the privacy leakage from the forward embeddings of split learning. Specifically, since the forward embeddings contain too much information about the label, the attacker can either use a few labeled samples to fine-tune the top model or perform unsupervised attacks such as clustering to infer the true labels from the forward embeddings. To prevent such kind of privacy leakage, we propose the potential energy loss to make the forward embeddings more 'complicated', by pushing embeddings of the same class towards the decision boundary. Therefore, it is hard for the attacker to learn from the forward embeddings. Experiment results show that our method significantly lowers the performance of both fine-tuning attacks and clustering attacks.

1 Introduction

Split learning [Vepakomma *et al.*, 2018; Gupta and Raskar, 2018] is a practical method for privacy-preserving machine learning on distributed data. By splitting the model into multiple parts (sub-models), split learning allows different parties to keep their data locally and only to share the intermediate output of their sub-models. Compared to cryptographic methods like [Mohassel and Zhang, 2017; Rathee *et al.*, 2020; Zhicong *et al.*, 2022], split learning is much more efficient in computation and communication. To date, split learning has been applied in multiple fields, e.g., graph learning [Chen *et al.*, 2022b], medical research [Ha *et al.*, 2021], and the internet of things [Koda *et al.*, 2020].

To perform split learning, the model should be split into multiple parts. Without loss of generality, we suppose the model is split into two parts, i.e., the bottom model M_b and the top model M_t, held by the feature owner (Alice) and the label owner (Bob), respectively. As shown in Figure 1 (top part), during the forward pass, Alice feeds the input feature X to the bottom model to get the forward embedding $Z = M_b(X)$, then sends Z to Bob. Bob feeds Z to the top model and gets the prediction $\hat{Y} = M_t(Z)$. As for the backward pass, Bob computes the gradients on the loss $\partial L/\partial M_t$ and $\partial L/\partial Z$. He uses the former one to update M_t and sends the latter one to Alice. Alice then computes the gradient with respect to M_b and updates its parameters. From the above description, we can see that split learning is very straightforward with low computation and communication overhead compared with cryptographic methods.

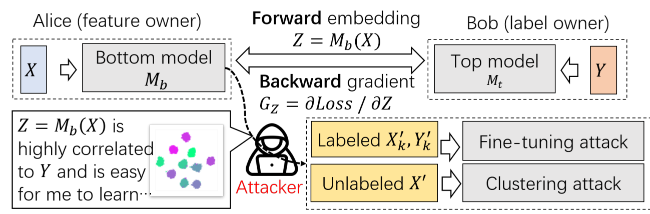

Figure 1: Attacks on the forward embeddings of split learning.

However, the price for efficiency is privacy. Many previous studies have investigated the privacy leakage of input features in split learning caused by the *exchange of intermediate results* [Abuadbba *et al.*, 2020; Pasquini *et al.*, 2021]. Instead, we focus on the privacy leakage caused by the *trained split model itself* in classification tasks, which has been demonstrated by [Fu *et al.*, 2022a; Sun *et al.*, 2022]. Consider the two-party split learning scenario described in the previous paragraph. As shown in Figure 1 (bottom part), if M_b is trained well, it gains the ability to separate samples of different classes and cluster the samples of the same class, in other words, Z becomes 'meaningful' and highly correlated with the label. Hence, Alice (or whoever obtains the bottom model) can 1) fine-tune M_t from random initialization with a few labeled samples, or 2) simply perform clustering with enough unlabeled input samples. In both cases, Alice can learn the complete classification model based on the forward embeddings. Considering that the complete model and thelabel could be private assets, such *model completion attack* caused by the forward embedding poses a significant privacy threat to split learning.

The protection of input features in split learning is already studied. For example, Vepakomma et al. [2020] decorrelate H_b and X by adding distance correlation [Székely *et al.*, 2007]

*Corresponding author

loss. This method is empirically successful because Z does not need to contain the majority of X's information—it only needs to contain the part most relevant to the label Y. On the other side, protecting label information is more challenging. Since $M_t(Z) = \hat{Y}$ is the model prediction, Z contains all information about $\hat{Y}$. While the training target is to make the prediction $\hat{Y}$ and the real label Y as close as possible, it seems impossible to prevent the attacker from deriving Y from Z.

To solve this problem, in this paper, we view it from a different perspective. At first glance, the attacker's target seems exactly the same as Bob's training objective, i.e., to learn a mapping from Z to Y. But there is a crucial difference between them. Alice and Bob perform the learning procedure on the entire training dataset, but on the contrary, the attacker can only access a small number of labeled samples by assumption. The problem now turns into modifying Z's distribution so that the attacker learns poorly with a few labeled samples while the benign actors learn well with sufficient labeled samples. We observe that such distribution can be obtained if the embeddings of the same class are distributed near the boundary of the decision region. However, during vanilla split learning, different-class embeddings tend to separate from each other while same-class embeddings are densely clustered, which is opposite to our purpose.

Our solution is inspired by a well-known physics phenomenon named electrostatic equilibrium, i.e., all the net charges distribute on the surface of the conductor. Interestingly, this distribution happens to be the desired distribution for Z, if we view the forward embeddings as charges. One reason for this phenomenon is that there is a repulsive force between every pair of *like charges* (charges that have the same sign, e.g., positive charges). Inspired by this, we propose the *potential energy loss* on the forward embeddings. The potential energy loss adds a repulsive force between each pair of same-class embeddings. During training, same-class embeddings are pushed toward the boundary of the decision region, resulting in a large learning error for the attacker. Therefore, the attacker cannot fine-tune the bottom model with a few labels or cluster unlabeled embeddings.

In summary, we make the following contributions:

- We formalize the privacy leakage of the forward embeddings in terms of the learning error of the attacker, and demonstrate that making embeddings lie near the decision boundary enhances privacy.
- We propose the potential energy loss on the forward embeddings to push them to their decision boundary, in order to reduce the privacy leakage from the forward embeddings.
- We conduct extensive experiments on multiple datasets, showing that our method significantly reduces the attacker's learning accuracy of both fine-tuning attacks and clustering attacks, and performs better than the existing distance correlation approach.

2 Related Work

Privacy Concerns of Split learning. Many studies have demonstrated the privacy concerns of split learning. Most of them focus on the privacy of input features. Abuadbba et al. [2020] show that applying split learning to CNN models can be dangerous since the intermediate output is highly correlated to the input. Luo et al. [2021] propose methods for feature inference attacks of split learning under certain conditions. As for the privacy of the label data, the attack can be based on either the forward embeddings or the backward gradients. For **forward embeddings**: Sun et al. [2022] find that bottom model output (forward embeddings) also leaks label data. Fu et al. [2022a] point out that the attacker can easily fine-tune the top model with a few labeled samples. For **backward gradients**: Li et al. [2022] investigated the label leakage brought by the backward gradients, for example, the norm of different classes' gradients can be different. Studies [Erdoğan *et al.*, 2022; Kariyappa and Qureshi, 2023] use a surrogate model and label to match the backward gradients and reveal the training labels. While the above studies assume the attacker is *passive*, i.e., he will not tamper with the training procedure but only exploits the information he received, Pasquini et al. [2021] propose an *active* attack by making the bottom model invertible by modifying the training objective.

Privacy Protection for Split Learning. Aside from relatively expensive cryptographic-based methods such as privacy-preserving machine learning [Mohassel and Zhang, 2017; Rathee *et al.*, 2020; Zhicong *et al.*, 2022] or partial cryptographic split learning methods [Fu *et al.*, 2022b; Zhou *et al.*, 2022], non-cryptographic methods mostly protect privacy by perturbing the forward embeddings or backward gradients. **Perturbing forward embeddings**: Vepakomma et al. [2020] add a distance correlation [Székely *et al.*, 2007] loss to decorrelate the input features, and similarly, Sun et al. [2022] use distance correlation to protect the label. Duan et al. [2023] perturb the embedding by minimizing the mutual information with both label and features to protect both, however, it requires a customized model structure. Chen et al. [2022a] protect the embedding in the recommendation model via differential privacy. **Perturbing backward gradients**: Li et al. [2022] protect the label during training by perturbing the backward gradients, although the forward embeddings could still leak the label information.

Data-Dependent Generalization Error. Most studies on data-dependent generalization error are based on the Rademacher and Gaussian complexity [Koltchinskii and Panchenko, 2002; Kontorovich and Weiss, 2014; Lei *et al.*, 2019], or the mutual information between the data and the algorithm output [Negrea *et al.*, 2019; Pensia *et al.*, 2018; Russo and Zou, 2020]. Different from them, Jin et al. [2020] derived generalization bounds directly from the data distribution, by proposing the so-called cover complexity, which is computed from the distances between same-class data points and different-class data points. It is somewhat related to our work since our method makes the data distribution more 'complicated' by pushing the data points to the decision boundary of their class.

3 Problem: Model Completion Attack

In this section, we demonstrate and formalize the privacy problem arising from the bottom model in split learning.

3.1 Leakage from Forward Embeddings

The hidden embeddings of neural networks are widely studied [Rauber *et al.*, 2017; Pezzotti *et al.*, 2018; Cantareira *et al.*, 2020]. Through visualization and other techniques, those studies show that the neural network gradually learns to make hidden embeddings of different classes separate, and those of the same class clustered together. Although this 'separation ability' seems to be essential for neural networks and may be the reason why they perform well on various tasks, it also brings security hazards for split learning.

In split learning, the model is split into one (or multiple) bottom model(s) and top model(s), which are held by different parties. Ideally speaking, any single party can not perform inference tasks, since he only gets a part of the complete model. Only multiple parties work together, can they make use of the complete model and perform inference tasks. In other words, the model should be trained and used in a 'shared' manner.

However, given the fact that the hidden embeddings of the model are meaningful, the attacker who has the bottom model can either fine-tune the top model with a small number of labeled samples [Fu *et al.*, 2022a], or just perform clustering on the forward embeddings to infer the labels [Sun *et al.*, 2022]. Thus, the privacy of split learning is violated.

3.2 Threat Model

We consider two threat models, i.e., fine-tuning attack with a few labeled samples and unsupervised clustering attack with massive unlabeled samples.

Fine-tuning Attack. We assume that the attacker has access to the trained bottom model M_b, along with a few labeled samples X'_k, Y'_k which include k samples for each class. The attacker also knows the architecture of the top model M_t, and performs the model completion attack via training M_t from a random initialization, given X'_k and Y'_k, with pre-trained M_b fixed.

Clustering Attack. We assume that the attacker has access to the trained bottom model M_b, along with massive unlabeled samples X'. To infer the labels of X', the attacker performs clustering algorithms on the forward embedding $Z' = M_b(X')$.

Notably, we assume **the attacks are conducted in the inference phase instead of the training phase**, as our method aims to reduce the privacy leakage from forward embeddings. To prevent leakage from backward gradients during training, one can use non-sensitive data for training or adopt existing approaches such as cryptography-based secure computation described in Section 2. It is worth noting that, some unsupervised/semi-supervised learning methods can train the complete model with good performance over unlabeled/partially labeled data [Berthelot *et al.*, 2019; Xu *et al.*, 2021]. However, they are not relevant to split learning since they do not require any knowledge about the trained bottom model or forward embedding.

3.3 Problem Formulation

In order to reduce the aforementioned privacy leakage while maintaining the model performance at the same time, our purpose is to train a split model $M = (M_b, M_t)$ such that the output of M_b is hard for the attacker to learn, while the complete M still maintains a high performance.

Definition 1 (Bottom model advantage)**.** *The bottom model advantage is the extra advantage obtained by the attacker when he has access to the trained bottom model. Consider an attack algorithm $\mathcal{A}$ whose input is the data D and (optionally) the bottom model M_b, the bottom model advantage is defined as follows:*

$$Adv(M_b; \mathcal{A}) = \mathbb{E}_D \left\{ R[\mathcal{A}(D; null)] - R[\mathcal{A}(D; M_b)] \right\}, \quad (1)$$

where $R[\cdot]$ is an error metric for the attack outcome. For example, in the fine-tuning attack, $D = (X'_k, Y'_k)$ is the leaked labeled samples, and $R[\cdot]$ is the error of the fine-tuned model on the test data; in the unsupervised attack, $D = X$ is the unlabeled data, and $R[\cdot]$ is the error of the clustering model on the test data. We use $\mathcal{A}(D; null)$ to represent the case that the attacker trains the whole model solely based on the leaked data, without any information about the bottom model or forward embedding (i.e., training from scratch).

Definition 2 (Perfect protection)**.** *For an attack algorithm $\mathcal{A}$, if the bottom model M_b satisfies $Adv(M_b; \mathcal{A}) = 0$, then we say that M_b achieves perfect protection against attack $\mathcal{A}$, since M_b provides no extra advantage for the attack.*

Thus, our purpose becomes to train a split model (M_b, M_t) such that:

- The performance of the complete model on the original task is as high as possible.
- The bottom model advantage under fine-tuning attacks and clustering attacks is as small as possible.

4 Method: Potential Energy Loss

In this section, we view the privacy leakage of the bottom model as a learning problem for the attacker. We first study the generalization error when fine-tuning the model with a small number of labeled samples. By a simplified example, we show that pushing the embeddings of same-class samples toward the decision boundary increases the generalization error. At the same time, clustering is difficult since the same-class embeddings are no longer close to each other. Inspired by the electrostatic equilibrium and Coulomb's law, we propose the potential energy loss on the forward embedding, to realize such distribution. The high-level view of our idea is presented in Figure 2.

4.1 Learning Error from Data Distribution

Recall that our goal is to train a split model (M_b, M_t), such that the bottom model provides little advantage to the attacker. To do this, we consider the attack process to be a learning process on the forward embeddings produced by the bottom model. We show that when the same-class embeddings are distributed near the boundary of the decision region, the performance of fine-tuning attack is decreased since a small number of samples cannot represent the overall distribution, and a small error on the estimation of decision boundary will cause a large classification error. Moreover, it naturally prevents clustering attacks since same-class embeddings are no longer close to each other.

Genralization Error

We use a simplified example to get some insights into the relationship between the data distribution and generalization error. Assume that all data points are distributed on the d-sphere $\{\mathbf{x} : \sum_{i=1}^{d} x_i^2 = 1\}$. Let the hypothesis set be an arbitrary hemisphere

$$\mathcal{H} = \{h : h(\mathbf{x}) = \text{Sign}[\mathbf{w} \cdot \mathbf{x}], \|\mathbf{w}\|_2 = 1\}. \tag{2}$$

Without loss of generality, we assume the target hypothesis is $f(\mathbf{x}) = \text{Sign}(x_1)$. We make the following assumptions:

- The probability density of samples only depends on the first dimension x_1, i.e., it is isotropic in any other dimensions.
- Given a set of positive samples $S = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, the learning algorithm simply outputs the normalized mean of these samples as the parameter of learned hypothesis, i.e., $f^{(S)}(\mathbf{x}) = \text{Sign}\left[\sum_{i=1}^{n} \mathbf{x}_i \cdot \mathbf{x} / \|\sum_{i=1}^{n} \mathbf{x}_i\|_2\right]$.

Now we want to estimate the generalization error when the learned parameter $\mathbf{w} = \sum_{i=1}^{n} \mathbf{x}_i / \|\sum_{i=1}^{n} \mathbf{x}_i\|_2$ slightly differs from the true parameter $\mathbf{e}_1$. Since the distribution is isotropic except in the direction of $\mathbf{e}_1$, we may assume that $\mathbf{w}$ lies on the plane expanded by the first two axis, i.e., $\mathbf{w} = \mathbf{e}_1 \cos\epsilon + \mathbf{e}_2 \sin\epsilon$, where ϵ is a small angle between $\mathbf{w}$ and $\mathbf{e}_1$. The generalization error is

$$\begin{aligned}
\frac{1}{2} \cdot R[\mathbf{w}] &= \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{S}} \text{Sign}[x_1] \cdot I[x_1 \cos\epsilon + x_2 \sin\epsilon \leq 0] \\
&= \int_{\substack{x_1 > 0 \\ x_1 \cos\epsilon + x_2 \sin\epsilon \leq 0 \\ x_1^2 + ... + x_d^2 = 1}} p(x_1, x_2, ..., x_d) dS \\
&\leq \int_{\substack{x_1^2 + ... + x_d^2 = 1 \\ 0 < x_1 \leq \tan\epsilon}} p(x_1, x_2, ..., x_d) dS \\
&\approx \int_{x_1=0}^{\epsilon} p_1(x_1) dx_1 \approx \epsilon p_1(0),
\end{aligned} \tag{3}$$

where p is the probability density of sample feature, and p_1 is the marginal density function of x_1. From (3) we can see that with ϵ fixed, the generalization bound is approximately proportional to the probability mass of the data points falling near the boundary of the target region.

Sampling Error

In the above analysis, the estimation error ϵ is fixed. We now explore the relationship between the data distribution and the distribution of ϵ. Notice that for any random variable X, if $X_1, ..., X_m$ are m independent samples, we have $\mathbb{E}\left[\left(\frac{1}{m}\sum_{i=1}^{m} X_i - \mathbb{E}[X]\right)^2\right] = \frac{1}{m}\mathbb{E}\left[(X - \mathbb{E}X)^2\right]$. In other words, if the random variable is likely to fall far from the mean of its distribution, the sample mean tends to have a larger error. Although ϵ is not exactly the error of the sample mean in our case since it is an angle, it is also reasonable to assume $\mathbb{E}[\epsilon^2] \propto \mathbb{E}\left[(X - \mathbb{E}X)^2\right]$. To make (the magnitude of) ϵ larger, the data points should be away from their mean as much as possible. Interestingly, in our case, this is also equivalent to pushing the data points toward the decision boundary.

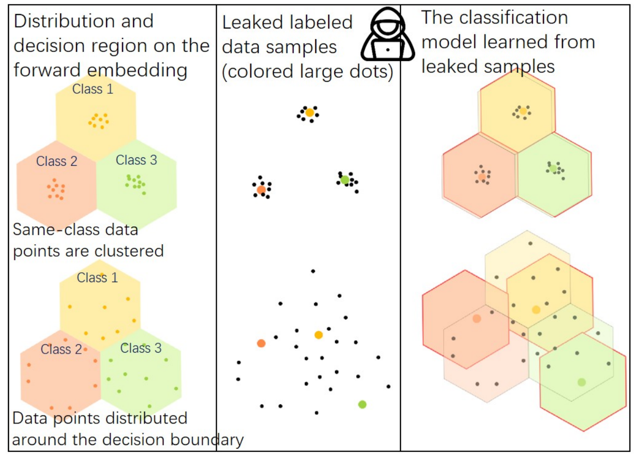

Figure 2: A high-level illustration of our idea.

Clustering Error

While the above discussion is on the generalization error under a small number of labeled data, we can easily see that such distribution makes clustering difficult. This is because the basic idea of clustering is that the data points in the same cluster shall be close to each other, and those in the different clusters shall be far from each other. While the same-class embeddings are being pushed toward the decision boundary, intra-class distances are increased, while inter-class distances are decreased. This is exactly opposite to the basic requirement of clustering.

In summary, pushing data points to the boundary of the decision region will increase the learning error of both fine-tuning and clustering for the following reasons:

- The sampling error tends to be larger.
- A small error in the decision region will result in a large generalization error.
- Intra-class distances are larger and inter-class distances are smaller.

4.2 Potential Energy Loss

When electrostatic equilibrium is established, any net charge resides on the surface of the conductor [Griffiths, 2005, ch. 2]. This is partly caused by Coulomb's law, which tells us that *like charges* (electric charges of the same sign) repel each other and *opposite charges* attract each other. Inspired by this, we can view the embeddings of the same class as like charges and have repulsive forces against each other. As a result, those data points will tend to be away from each other and be pushed to the boundary of the decision region.

Coulomb's law is stated as follows:

$$\mathbf{F} = kq_1q_2(\mathbf{r}_1 - \mathbf{r}_2)/\|r_1 - r_2\|_2^3, \tag{4}$$

where k is a constant, q_1, q_2 is the signed magnitude of the charges, $\mathbf{r}_1, \mathbf{r}_2$ are their positions, $\|\cdot\|_2$ is the Euclidean norm, and $\mathbf{F}$ is the repulsive force on the first charge caused by the second charge. Since we assume all embeddings belonging to the same class have the same sign and magnitude while ignoring the constant term, (4) becomes $\mathbf{F} = \frac{\mathbf{r}_1 - \mathbf{r}_2}{\|\mathbf{r}_1 - \mathbf{r}_2\|_2^3}$. Notice that the repulsive force is the gradient of the electric potential energy, we can further write $\mathbf{F}$ as $\mathbf{F} = \nabla_{\mathbf{r}_1} \frac{1}{\|\mathbf{r}_1 - \mathbf{r}_2\|_2}$, which is naturally suited to the gradient descent method. Based on this, we define the *potential energy loss* (PELoss) as

$$L_{\text{pe}} = \sum_{c \in \mathcal{C}} \sum_{\mathbf{z} \in Z_c} \sum_{\mathbf{z}' \in Z_c, \mathbf{z}' \neq \mathbf{z}} \frac{1}{\|\mathbf{z} - \mathbf{z}'\|_2}, \tag{5}$$

where $\mathcal{C}$ is the label set, and Z_c is the forward embeddings of c-labeled samples.

By adding L_{pe} to the loss function, during the training of the split model, the bottom model outputs of the same class are pushed away from each other, and move towards the boundary of the decision region of its own class. While in the 3-D case, minimizing the potential energy leads to zero charge density inside the region by Thomson's theorem, it is not necessarily true in the high-dimensional case. However, we are able to prove a weaker theorem, i.e., minimizing the potential energy leads to a non-zero probability mass in the border of the region.

Theorem 1 (Border distribution)**.** *Consider a d-dimensional bounded region $\Omega \subset \mathbb{R}^d$, we denote the density of probability distribution in Ω which minimizes the potential energy functional as*

$$f^* = \operatorname*{argmin}_f \text{PE}(f) = \operatorname*{argmin}_f \int_{x \in \Omega} \int_{y \in \Omega} \frac{f(x) f(y)}{\|x - y\|_2} dU dV. \tag{6}$$

where f is the probability density function of some distribution such that $f \geq 0$ and $\int_\Omega f(x) dV = 1$, $\Delta_\epsilon \Omega = \{x : x + \epsilon r \notin \Omega, x \in \Omega, \|r\|_2 = 1\}$ is the set of points whose distance to the border of Ω is less than ϵ. Then f^* satisfies that $\int_{x \in \Delta_\epsilon \Omega} f^*(x) dV > 0$ for any $\epsilon > 0$.

Proof. See Appendix A. □

Adding Layer Normalization

One hidden condition for our method is that the decision boundary is also the actual set boundary of same-class embeddings. This requires the embedding space to be a borderless manifold. Otherwise, adding repulsive force may just push the embeddings toward the set boundary instead of the decision boundary. For example, if the embedding space is the trivial Euclidean space $\mathbb{R}^d$, the repulsive force will make embeddings far from the origin, while the inter-class distances are still large. To overcome this, we simply enforce layer normalization [Ba *et al.*, 2016] (without element-wise affine transformation) on $\mathbf{z}$, which restricts $\mathbf{z}$ to the d-sphere of radius $\sqrt{d}$, i.e., $\|\mathbf{z}\|_2^2 = d$. Accordingly, the distance metric is changed to angular distance, i.e., $\arccos\langle \mathbf{z}, \mathbf{z}' \rangle$. The potential energy loss defined in (5) should be changed to

$$L_{\text{pe}} = \sum_{c \in \mathcal{C}} \sum_{\substack{\mathbf{z}, \mathbf{z}' \in Z_c \\ \mathbf{z}' \neq \mathbf{z}}} \frac{1}{\arccos\langle \mathbf{z}, \mathbf{z}' \rangle}. \tag{7}$$

The combined loss for split training is $L' = L + \alpha L_{\text{pe}}$, where L is the original loss function (i.e., cross-entropy loss), α is a coefficient to control the intensity of repulsive force.

4.3 Relationship with Distance Correlation

Distance correlation is used to protect both the input feature [Vepakomma *et al.*, 2020] and the label [Sun *et al.*, 2022]. Here we consider the label-protection case that distance correlation loss is applied to the forward embedding and the label. The distance correlation loss on one batch is

$$L_{\text{dcor}} = \sum_{i,j=1}^{n} d_{i,j} d'_{i,j} \Big/ \sqrt{\sum_{i,j=1}^{n} d_{i,j}^2 \sum_{i,j=1}^{n} d'^2_{i,j}}, \tag{8}$$

where $d_{i,j}$ is the doubly-centered distance between i-th sample's embedding and j-th sample's embedding, and $d'_{i,j}$ is the doubly-centered distance between i-th label and j-th label. In the classification task, same-class samples have the same label. If the i-th sample and j-th sample belong to the same class, we have $d'_{i,j} = 0$. Thus, (8) (ignoring the denominator) is converted to

$$\sum_{\substack{c, c' \in \mathcal{C} \\ c \neq c'}} \sum_{\substack{\mathbf{z} \in Z_c \\ \mathbf{z}' \in Z_{c'}}} k \left(\|\mathbf{z} - \mathbf{z}'\|_2 - \overline{\|\mathbf{z} - \cdot\|_2} - \overline{\|\cdot - \mathbf{z}'\|_2} + \overline{\|\cdot - \cdot\|_2} \right), \tag{9}$$

where $\mathcal{C}$ is the set of labels, k is some constant, $\overline{\|\mathbf{z} - \cdot\|_2}$ is the average distance from $\mathbf{z}$ to other embeddings within the batch (similar for $\overline{\|\cdot - \mathbf{z}'\|_2}$), and $\overline{\|\cdot - \cdot\|_2}$ is the average distance between all embedding pairs within the batch. We can see that minimizing the distance correlation is similar to minimizing the inter-class distances. As our method is to maximize intra-class distances, minimizing distance correlation has a similar effect. However, in an intuitive understanding, the fact that embeddings of different classes lie in different directions makes distance correlation naturally noisy. Experiments in Section 5.4 also illustrate that minimizing distance correlation fails to push away some same-class samples.

5 Empirical Study

We conduct experiments on four different datasets, i.e., MNIST [LeCun *et al.*, 1998], Fashion-MNIST [Xiao *et al.*, 2017], CIFAR-10 [Krizhevsky *et al.*, 2009], and DBpedia [Auer *et al.*, 2007]. We compare the attack performance of vanilla split training, training with potential energy loss (PELoss, our method), training with distance correlation loss (DcorLoss) proposed by [Vepakomma *et al.*, 2020; Sun *et al.*, 2022], and a simple baseline of label differential privacy method based on randomly flipping a certain portion of labels (LabelDP). For DcorLoss, layer normalization is also added like in our method for training stability. The attacks include both fine-tuning attacks and clustering attacks. For clustering attacks, we use the classical k-Means algorithm [MacQueen and others, 1967], with the number of classes known

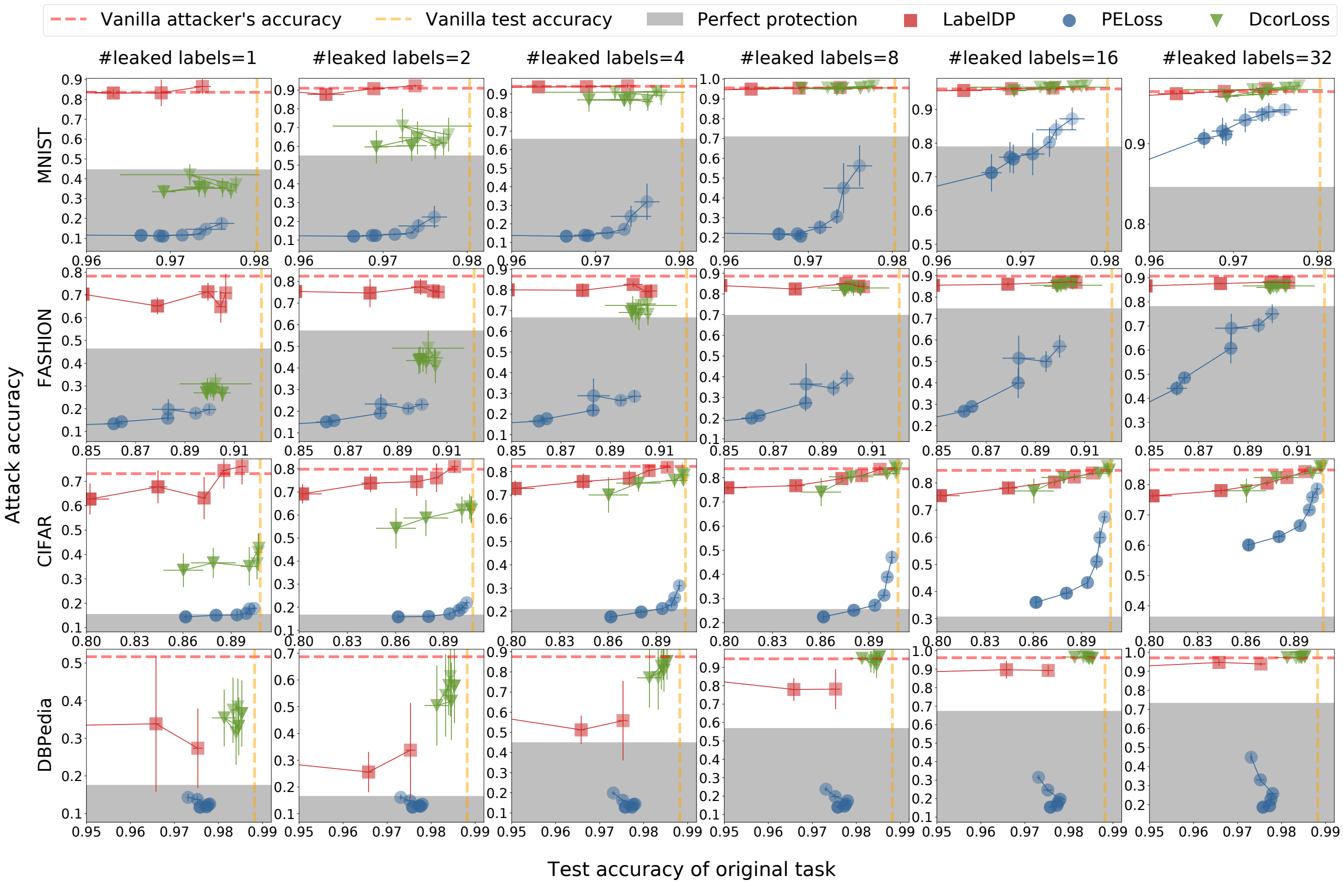

Figure 3: Test accuracy vs. attack accuracy of the fine-tuning attack. The data point lies at the lower-right position means the experiment achieves higher test accuracy on the original task with lower attacker's accuracy, which is our desired result. The data point lies in the gray area means that the attacker's accuracy is lower than training from scratch without the bottom model's information, i.e., perfect protection.

to the attacker. We also measure the distances between sample pairs to illustrate the mechanism of PELoss.

Due to space limits, detailed descriptions of the model architectures and training strategies, and more results are presented in Appendix B and C. Other results include studies on different split positions, different attack layers, different forward embedding dimensions, and the t-SNE [Van der Maaten and Hinton, 2008] visualization of the forward embeddings.

5.1 Experiment Settings

We implement the experiment codes using the PyTorch and Scikit-Learn [Pedregosa *et al.*, 2011] libraries, and run them on servers with NVIDIA RTX3090 GPUs. Each experiment is repeated 3 times for the original tasks and 5 times for the attack tasks, with different random seeds. We use a 3-layer fully-connected network for MNIST, a simple convolutional network for Fashion-MNIST, ResNet-20 [He *et al.*, 2016] for CIFAR-10 dataset, and TextCNN [Kim, 2014] for DBPedia. The split position for each model is the last dense layer by default since the last forward embedding is the closest to the label and is the most difficult to protect. For the selection of hyperparameters, we vary the loss coefficient (α) of PELoss from $0.25 \sim 32$ and $1 \sim 32$ for DcorLoss. For LabelDP, the ratio of randomly flipped labels varies from $0.01 \sim 0.16$. In all experiments, the value doubles each time. Detailed experiment settings are provided in Appendix B.

5.2 Fine-tuning Attack

We report the test accuracy of the original task and the accuracy of fine-tuning attacks using different protection methods in Figure 3. Deeper colors represent stronger protection, e.g., larger loss coefficient or label flipping probability, while lighter colors mean more leaning toward preserving the test accuracy, e.g., smaller loss coefficient or label flipping probability. We also plot the test accuracy (orange dashed line) and the attack accuracy (red dashed line) in the vanilla split learning case, and the perfect protection area (where the attack accuracy is lower than the accuracy of training from scratch using leaked labels). For all methods, we observe that decreasing the attack accuracy usually also lowers test accuracy on the original task, and more leaked labels lead to higher attack accuracy. Although all methods protect privacy to some extent at the cost of damaging the model performance, it is obvious that our proposed PELoss is superior to DcorLoss and LabelDP. PELoss has the following advantages compared with other methods:

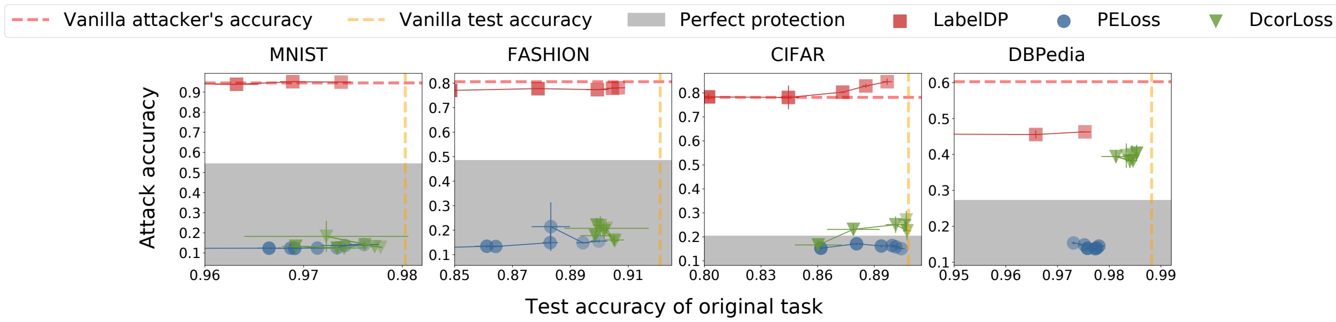

Figure 4: Test accuracy vs. attack accuracy of the clustering attack. The notations are the same as Figure 3.

- The curves of PELoss are constantly at the lower-right side of other methods. In other words, on the same test accuracy level, the PELoss has a significantly lower attacker's accuracy than DcorLoss and LabelDP.
- The curves are smoother and the error bars are also smaller using PELoss, indicating it is more responsive to the change of α and the performance is more stable. Thus, it is easier to balance privacy and model performance using PELoss.
- PELoss has many data points in the perfect protection zone, while other methods rarely achieve perfect protection.

In summary, PELoss provides significantly stronger privacy protection against model completion attacks, while preserving the model performance better than other methods. We also notice that in some datasets, varying the coefficients of DcorLoss seems to have no effect, however, when the coefficient is large enough (e.g., 32), the training could diverge.

5.3 Clustering Attack

We report the test accuracy of the original task and the accuracy of clustering attacks using different protection methods in Figure 4. In all the experiments, the attacker can access the entire unlabeled test dataset which contains more than 10,000 unlabeled samples, and the number of classes is known to the attacker. The notations are the same as fine-tuning attacks, while here the gray area (perfect protection) means that the accuracy of clustering on forward embeddings is lower than directly clustering on the raw input samples. The accuracy here is defined as the maximum accuracy among all possible cluster-label assignments.

We can see that PELoss achieves perfect protection in all cases (i.e., the clustering result on the forward embeddings is worse than on the raw input data), and is significantly better than DcorLoss on CIFAR and DBPedia. On the other side, LabelDP performs badly against clustering attacks.

5.4 Distances between Sample Pairs

To better illustrate the distribution of forward embeddings, we plot the distribution of angular distances between same-class and different-class sample pairs in Figure 5. For vanilla training, the angular distances are small between same-class samples, and are large between different-class samples. For

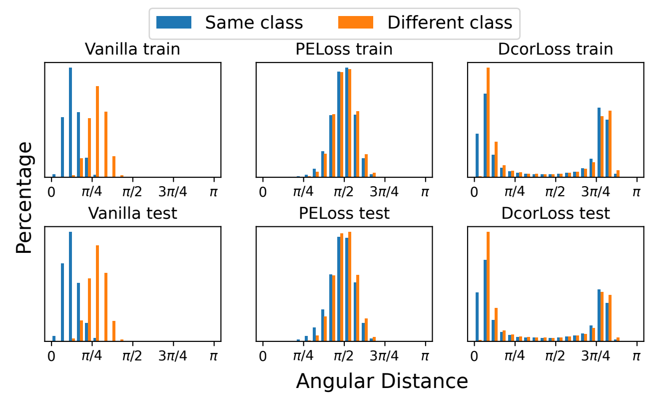

Figure 5: Angular distance distribution between sample pairs on MNIST. $\alpha = 1$ for PELoss and 0.5 for DcorLoss, as they have similar performance on the original task.

PELoss, the angular distances between sample pairs are close to $\pi/2$, no matter whether they belong to the same class or not, in both training data and test data. In contrast, for DcorLoss, the angular distances appear to have a double-peak distribution. We can see that there are significantly more same-class sample pairs that have an angular distance around 0, which weakens the protection.

6 Conclusion

In this paper, we investigate the privacy leakage of split learning arising from the learned bottom model. We view the model completion attack as a learning procedure for the attacker and turn the privacy-preserving problem into a learning problem. We find that pushing embeddings to their decision boundary increases the learning error for the attacker, and propose the potential energy loss on forward embedding to protect the privacy. Extensive experiments show that our method significantly restricts the ability of the attacker's fine-tuning and clustering attacks while reserving the model performance, superior to baselines in terms of both utility and privacy. The limitation of this work mainly includes that only the inference process is protected, and the lack of a theoretical leakage bound.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 72192823), the "Ten Thousand Talents Program" of Zhejiang Province for Leading Experts (No. 2021R52001), and Ant Group.

References

[Abuadbba *et al.*, 2020] Sharif Abuadbba, Kyuyeon Kim, Minki Kim, Chandra Thapa, Seyit Ahmet Çamtepe, Yansong Gao, Hyoungshick Kim, and Surya Nepal. Can we use split learning on 1d CNN models for privacy preserving training? In *ASIA CCS '20: The 15th ACM Asia Conference on Computer and Communications Security*, pages 305–318. ACM, 2020.

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2007.

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pages 5050–5060, 2019.

[Cantareira *et al.*, 2020] Gabriel Dias Cantareira, Elham Etemad, and Fernando V. Paulovich. Exploring neural network hidden layer activity using vector fields. *Inf.*, 11(9):426, 2020.

[Chen *et al.*, 2022a] Chaochao Chen, Huiwen Wu, Jiajie Su, Lingjuan Lyu, Xiaolin Zheng, and Li Wang. Differential private knowledge transfer for privacy-preserving cross-domain recommendation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1455–1465. ACM, 2022.

[Chen *et al.*, 2022b] Chaochao Chen, Jun Zhou, Longfei Zheng, Huiwen Wu, Lingjuan Lyu, Jia Wu, Bingzhe Wu, Ziqi Liu, Li Wang, and Xiaolin Zheng. Vertically federated graph neural network for privacy-preserving node classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 1959–1965, 2022.

[Duan *et al.*, 2023] Lin Duan, Jingwei Sun, Yiran Chen, and Maria Gorlatova. Privascissors: Enhance the privacy of collaborative inference through the lens of mutual information. *CoRR*, abs/2306.07973, 2023.

[Erdoğan *et al.*, 2022] 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.

[Fu *et al.*, 2022a] 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)*, 2022.

[Fu *et al.*, 2022b] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. Blindfl: Vertical federated machine learning without peeking into your data. In *SIGMOD '22: International Conference on Management of Data*, pages 1316–1330. ACM, 2022.

[Griffiths, 2005] David J Griffiths. Introduction to electrodynamics, 2005.

[Gupta and Raskar, 2018] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *J. Netw. Comput. Appl.*, 116:1–8, 2018.

[Ha *et al.*, 2021] Yoo Jeong Ha, Minjae Yoo, Gusang Lee, Soyi Jung, Sae Won Choi, Joongheon Kim, and Seehwan Yoo. Spatio-temporal split learning for privacy-preserving medical platforms: Case studies with covid-19 ct, x-ray, and cholesterol data. *IEEE Access*, 9:121046–121059, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778. IEEE Computer Society, 2016.

[Jin *et al.*, 2020] Pengzhan Jin, Lu Lu, Yifa Tang, and George Em Karniadakis. Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. *Neural Networks*, 130:85–99, 2020.

[Kariyappa and Qureshi, 2023] 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, 2023.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1746–1751. ACL, 2014.

[Koda *et al.*, 2020] Yusuke Koda, Jihong Park, Mehdi Bennis, Koji Yamamoto, Takayuki Nishio, Masahiro Morikura, and Kota Nakashima. Communication-efficient multimodal split learning for mmwave received power prediction. *IEEE Communications Letters*, 24(6):1284–1288, 2020.

[Koltchinskii and Panchenko, 2002] Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

[Kontorovich and Weiss, 2014] Aryeh Kontorovich and Roi Weiss. Maximum margin multiclass nearest neighbors. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 892–900, 2014.

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

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[Lei *et al.*, 2019] Yunwen Lei, Ürün Dogan, Ding-Xuan Zhou, and Marius Kloft. Data-dependent generalization bounds for multi-class classification. *IEEE Trans. Inf. Theory*, 65(5):2995–3021, 2019.

[Li *et al.*, 2022] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

[Luo *et al.*, 2021] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature inference attack on model predictions in vertical federated learning. In *37th IEEE International Conference on Data Engineering, ICDE 2021*, pages 181–192. IEEE, 2021.

[MacQueen and others, 1967] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

[Mohassel and Zhang, 2017] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy, SP 2017*, pages 19–38. IEEE Computer Society, 2017.

[Negrea *et al.*, 2019] Jeffrey Negrea, Mahdi Haghifam, Gintare Karolina Dziugaite, Ashish Khisti, and Daniel M. Roy. Information-theoretic generalization bounds for SGLD via data-dependent estimates. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pages 11013–11023, 2019.

[Pasquini *et al.*, 2021] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2113–2129. ACM, 2021.

[Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[Pensia *et al.*, 2018] Ankit Pensia, Varun S. Jog, and Po-Ling Loh. Generalization error bounds for noisy, iterative algorithms. In *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*, pages 546–550. IEEE, 2018.

[Pezzotti *et al.*, 2018] Nicola Pezzotti, Thomas Höllt, Jan C. van Gemert, Boudewijn P. F. Lelieveldt, Elmar Eisemann, and Anna Vilanova. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE Trans. Vis. Comput. Graph.*, 24(1):98–108, 2018.

[Rathee *et al.*, 2020] Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow2: Practical 2-party secure inference. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Securit*, pages 325–342. ACM, 2020.

[Rauber *et al.*, 2017] Paulo E. Rauber, Samuel G. Fadel, Alexandre X. Falcão, and Alexandru C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Trans. Vis. Comput. Graph.*, 23(1):101–110, 2017.

[Russo and Zou, 2020] Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *IEEE Trans. Inf. Theory*, 66(1):302–323, 2020.

[Sun *et al.*, 2022] Jiankai Sun, Xin Yang, Yuanshun Yao, and Chong Wang. Label leakage and protection from forward embedding in vertical federated learning. *CoRR*, abs/2203.01451, 2022.

[Székely *et al.*, 2007] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769 – 2794, 2007.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[Vepakomma *et al.*, 2018] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data, 2018.

[Vepakomma *et al.*, 2020] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning. In *20th International Conference on Data Mining Workshops, ICDM Workshops 2020*, pages 933–942. IEEE, 2020.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[Xu *et al.*, 2021] Yi Xu, Jiandong Ding, Lu Zhang, and Shuigeng Zhou. DP-SSL: towards robust semi-supervised learning with A few labeled samples. In *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pages 15895–15907, 2021.

[Zhicong *et al.*, 2022] Huang Zhicong, Lu Wen-jie, Hong Cheng, and Ding Jiansheng. Cheetah: Lean and fast secure Two-Party deep neural network inference. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, August 2022.

[Zhou *et al.*, 2022] Jun Zhou, Longfei Zheng, Chaochao Chen, Yan Wang, Xiaolin Zheng, Bingzhe Wu, Cen Chen, Li Wang, and Jianwei Yin. Toward scalable and privacy-preserving deep neural network via algorithmic-cryptographic co-design. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–21, 2022.