Privacy Against Agnostic Inference Attacks in Vertical Federated Learning

MORTEZA VARASTEH, CSEE, University of Essex, Colchester, United Kingdom of Great Britain and Northern Ireland

A novel form of inference attack in vertical federated learning (VFL) is proposed, where two parties collaborate in training a machine learning (ML) model. Logistic regression is considered for the VFL model. One party, referred to as the active party, possesses the ground truth labels of the samples in the training phase, while the other, referred to as the passive party, only shares a separate set of features corresponding to these samples. It is shown that the active party can carry out inference attacks on both training and prediction phase samples by acquiring an ML model independently trained on the training samples available to them. This type of inference attack does not require the active party to be aware of the score of a specific sample, hence it is referred to as an agnostic inference attack. It is shown that utilizing the observed confidence scores during the prediction phase, before the time of the attack, can improve the performance of the active party's autonomous ML model, and thus improve the quality of the agnostic inference attack. As a countermeasure, privacy-preserving schemes (PPSs) are proposed. While the proposed schemes preserve the utility of the VFL model, they systematically distort the VFL parameters corresponding to the passive party's features. The level of the distortion imposed on the passive party's parameters is adjustable, giving rise to a tradeoff between privacy of the passive party and interpretability of the VFL outcomes by the active party. The distortion level of the passive party's parameters could be chosen carefully according to the privacy and interpretability concerns of the passive and active parties, respectively, with the hope of keeping both parties (partially) satisfied. Finally, experimental results demonstrate the effectiveness of the proposed attack and the PPSs.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Privacy-preserving protocols**; *Privacy protections;*

Additional Key Words and Phrases: Machine learning, privacy, federated learning

ACM Reference Format:
Morteza Varasteh. 2026. Privacy Against Agnostic Inference Attacks in Vertical Federated Learning. *ACM Trans. Priv. Sec.* 29, 3, Article 27 (May 2026), 34 pages. https://doi.org/10.1145/3808698

1 Introduction

The emergence of distributed **machine learning** (**ML**) techniques has been a game-changer in the way valuable information is obtained from raw data in various fields, such as computer vision, image recognition, financial services, and natural language processing. With the growing need for the utilization of distributed data to construct more accurate and advanced ML models, there has been a surge in demand in multiple sectors. To address the limitations of centralized ML models,

Author's Contact Information: Morteza Varasteh, CSEE, University of Essex, Colchester, United Kingdom of Great Britain and Northern Ireland; e-mail: mv19977@essex.ac.uk.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).
ACM 2471-2566/2026/05-ART27
https://doi.org/10.1145/3808698

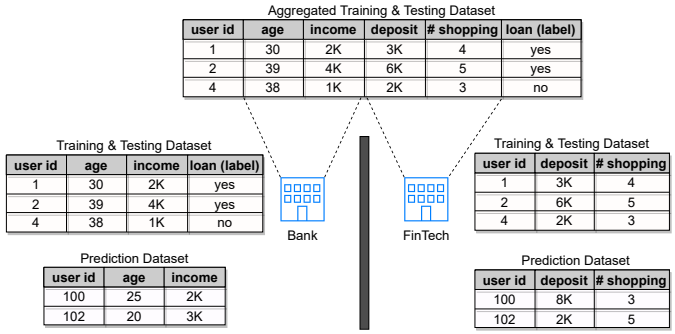


Fig. 1. Digital banking as an example of vertical federated learning [11].

including issues with data storage, excessive computations, security and privacy breaches, various distributed ML techniques have been proposed [1]. One such technique is **federated learning** (**FL**), which was introduced in [2]. In FL, two or more data-centers, known as parties, work together to train a shared ML model, thereby addressing the concerns of traditional centralized learning. This approach has gained widespread attention for its applications in real-life scenarios, such as in health systems [3, 4], keyboard prediction [5, 6], and e-commerce [7, 8].

Three types of FL are commonly studied in the literature, including **Horizontal FL** (**HFL**), **Vertical FL** (**VFL**), and Transfer FL. The distinction between these approaches lies in the manner in which data are partitioned and shared among parties. In HFL, each party holds a unique set of samples, but all parties share the same features [9]. On the other hand, VFL involves parties sharing the same samples, but each party trains and predicts collaboratively using a separate set of features [10]. As shown in Figure 1, an illustration of VFL is presented, where a bank and a FinTech company collaborate in training a classifier that will be used by the bank to approve or reject credit card applications. The bank, as the active party, holds the ground truth labels for the samples in the training set. Meanwhile, the FinTech company, as the passive party, contributes to the VFL process by sharing a distinct set of features pertaining to the samples.

After the VFL model is trained, both the bank and the FinTech company use it collaboratively to make predictions for new samples. To do this, each party inputs their corresponding features for each sample into the model. Typically, a third party known as the **coordinator authority** (**CA**) is involved in this collaboration during both the training and prediction phases. The CA's role during the prediction phase is to obtain the model's outcome for new samples and share it with the active party when requested. The model output can be in the form of a confidence score, which is a probability vector representing the likelihood of each class, or a one-hot vector representing the output class label. Due to privacy concerns, the confidence score of a sample may be distorted before it is delivered to the active party, typically by providing a noisy or rounded version of the score. In the literature, generally two scenarios are considered for this type of collaboration, namely white-box and black box settings, where the active party is aware and unaware of the passive party's parameters, respectively.

Focusing on the prediction phase, in [11], it has been demonstrated that when the exact form of the confidence scores is shared with the active party, there is a risk of the adversary (i.e., the active party) reconstructing the features of the passive party, which are considered private information. This can result in a privacy leak for the users, and therefore known as inference attack. The reconstruction is accurate when the number of unknown features, or the passive party features, is less than the number of classes. In [12] various approaches regarding improved

inference attacks have been proposed that are based on solving an under-determined system of linear equations. In particular, it is shown that using methods that approximate the best worst case solution among feasible solutions can significantly improve the attack accuracy. Furthermore, as a defense mechanism, it has been shown in [12] that by transforming the passive features [1] in a linear fashion and keeping the transformation mapping hidden from the active party, it is possible to preserve the model's accuracy while reducing the privacy leakage (i.e., higher mean squared error in reconstructing the passive features).

While previous works have studied inference attacks in the prediction phase using samples whose confidence scores (or a different version of them) are available at the active party, in this article, under the white-box setting, we focus on the privacy leakage of samples whose confidence scores are not yet available at the active party. This can include samples in the prediction phase for which no score has been requested by the active party, as well as samples used in the training phase to train the VFL model. Since the adversary is unaware of the confidence score of the attack target, we refer to this type of attack as an *agnostic inference attack*. Focusing on this type of privacy leakage, the main contributions of the article are as follows.

— Building on the linear-algebraic feature reconstruction framework originally introduced in [12], the active party uses its available resources, i.e., the training data with only the active features and labels, to reconstruct an independent classifier similar to the VFL model. This classifier is named the **adversary model** (**AM**). This is possible because the active party already has access to the ground truth labels of samples in the training set. Unlike prior inference attacks that assume access to exact or perturbed confidence scores of the attack targets, the AM enables inference in an *agnostic* setting where such scores are unavailable. The AM can be used for two purposes. First, as a means to estimate any sample's confidence score and conduct inference attacks on the passive features using the reconstruction methods introduced in [12]. The main aspect of this type of attack is that the adversary is agnostic to the confidence score or any other form of it corresponding to the attack target, which puts not only prediction samples but also *all training samples* at risk of privacy leakage. Second, by comparing the sample-by-sample scores obtained from the AM and the VFL model during the prediction phase, the active party can use the amount of score mismatch as a measure of the improvement in classification due to the VFL and decide whether to continue or stop requesting further collaborative predictions. This is particularly useful when the active party incurs a cost for each VFL model prediction made through the CA.

— Although the AM is primarily trained on the training samples, its ability to conduct an attack can be enhanced by taking into account the prediction samples and their associated confidence scores that have already been communicated to the active party. We consider scenarios where the confidence scores are delivered in their exact form to the active party.[2] It is observed that incorporating these scores leads to significant improvements in the effectiveness of the attack. This refinement mechanism, referred to as the **refined adversary model** (**RAM**), aligns the AM's outputs more closely with those of the jointly trained VFL model using only a small number of observed prediction scores. The RAM does not introduce a new

[1] For brevity, throughout the article we use passive (active) features and passive (active) parameters instead of passive (active) party's features and passive (active) party's parameters, respectively.

[2] We assume that confidence scores are delivered to the active party in their exact form (soft version). This represents a worst-case scenario from a privacy perspective, as soft scores generally reveal more information than hard labels or coarsened outputs. At the same time, this assumption is practically motivated as in many real-world deployments, the active party requires calibrated confidence scores for downstream tasks such as ranking, risk assessment, threshold selection, or soft clustering (e.g., prioritizing decisions under budget or resource constraints).

reconstruction technique, but rather improves the quality of score approximation within the same underlying linear system used in prior inference attacks.
This realization highlights the limitations of **privacy-preserving schemes** (**PPSs**) that operate solely at the score level, and motivates a shift in focus toward the model parameters themselves rather than the confidence scores or passive features, departing from strictly black-box or white-box views in existing VFL settings.

— We propose PPSs as a defense mechanism against agnostic inference attacks. The proposed schemes are designed to maintain the utility of the VFL model, i.e., delivering confidence scores in their exact form, while at the same time preventing adversaries from misusing the VFL collaboration. Rather than altering the confidence scores, the PPSs operate by systematically distorting the parameters corresponding to the passive party's features, while keeping the underlying prediction model intact.
The level of distortion applied to the passive parameters is adjustable and determines the degree of privacy leakage of the passive party, as measured by the MSE of reconstructed features. A higher level of distortion results in lower privacy compromise. This introduces an explicit privacy–interpretability tradeoff, where interpretability at the active party is quantified via the deviation between the released and original passive parameters.
To address this tradeoff, we formulate PPS design as constrained optimization problems, including problems over Stiefel manifolds, that enable controlled parameter distortion while preserving model utility. These schemes generalize prior feature-transformation-based defenses by operating directly on model parameters and by explicitly accounting for agnostic inference attacks.

The article is structured as follows. In Section 2, an explanation of the system model under study is provided. In Section 3, the characteristics of the datasets utilized in this article, as well as the model used for evaluating the results, are reviewed. Section 4 is dedicated to the introduction of agnostic inference attack and is divided into three sub-sections. In Section 4.1, the necessary background information required to perform an inference attack is discussed. Subsequently, in Section 4.2, the procedure for obtaining and refining the AM is detailed, followed by a performance analysis of the agnostic inference attack in Section 4.3. The proposed PPSs against agnostic inference attack are described in detail in Section 5, which is further divided into two sub-sections. In Section 5.1, the geometry of optimization algorithms with orthogonality constraints, specifically optimization over Stiefel manifolds, is reviewed to provide a better understanding of one of the proposed PPSs. Additionally, an overview of the concept of interpretability in the context of the present work is also provided. In Section 5.2, the technical analysis of the proposed PPSs is presented. In Section 6, the experimental results are reported and discussed, along with a comprehensive evaluation of the performance of agnostic inference attack on real-world datasets under various scenarios and the performance of the proposed PPSs. The article concludes in Section 7.

Notations. Matrices and vectors [3] are denoted by bold capital (e.g., $\mathbf{A}, \mathbf{Q}$) and bold lower case letters (e.g., $\mathbf{b}, \mathbf{z}$), respectively. Sets are denoted by capital letters in calligraphic font (e.g., $\mathcal{X}, \mathcal{G}$) with the exception of the set of real numbers, which is denoted by $\mathbb{R}$. The cardinality of a finite set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. The transpose of $\mathbf{A}_{m\times k}$ is denoted by $\mathbf{A}^T$, and when $m = k$, its trace and determinant are denoted by $\mathrm{Tr}(\mathbf{A})$ and $\det(\mathbf{A})$, respectively. For an integer $n \geq 1$, the terms $\mathbf{I}_n$, $\mathbf{1}_n$, and $\mathbf{0}_n$ denote the n-by-n identity matrix, the n-dimensional all-one, and all-zero column vectors, respectively, and whenever it is clear from the context, their subscripts are dropped. For integers $m \leq n$, we have the discrete interval $[m : n] \triangleq \{m, m+1, \ldots, n\}$, and the set $[1 : n]$ is written in

[3] All the vectors considered in this article are column vectors.

short as $[n]$. For $\mathbf{x} \in \mathbb{R}^n$ and $p \in [1, \infty]$, the L^p-norm is defined as $\|\mathbf{x}\|_p \triangleq (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$, $p \in [1, \infty)$, and $\|\mathbf{x}\|_\infty \triangleq \max_{i\in[n]} |x_i|$. Throughout the article, $\|\cdot\|$ (i.e., without subscript) refers to the L^2-norm and spectral norm for vectors and matrices, respectively. For a matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$, we have $\text{Vec}(\mathbf{W}) = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_n^T]^T$. The symbol $\otimes$ stands for the Kronecker product.

2 System Model

2.1 Machine Learning (ML)

An ML model is a function $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{U}$ parameterized by the vector $\boldsymbol{\theta}$, where $\mathcal{X}$ and $\mathcal{U}$ denote the input and output spaces, respectively. In this article, we consider the supervised classification setting, where a labeled training dataset is used to train the model.

Assume that a training dataset $\mathcal{D}_{\text{train}} \triangleq \{(\mathbf{x}_i, u_i) | i \in [n]\}$ is given, where each $\mathbf{x}_i$ is a d_t-dimensional example/sample and u_i denotes its corresponding label. Learning refers to the process of obtaining the parameter vector $\boldsymbol{\theta}$ in the minimization of a loss function, i.e.,

$$\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} l(f_{\boldsymbol{\theta}}(\mathbf{x}_i), u_i) + \omega(\boldsymbol{\theta}), \tag{1}$$

where $l(\cdot, \cdot)$ measures the prediction loss of $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$, while the true label is $u_i, i \in [n]$. To avoid overfitting, a regularization term $\omega(\boldsymbol{\theta})$ can be added to the optimization.

Once the model is trained, i.e., $\boldsymbol{\theta}$ is obtained, it can be used for the prediction of any new sample. In practice, the prediction is a (probability) vector-valued, i.e., it is a vector of confidence scores as $\mathbf{c} = (c_1, c_2, \dots, c_k)^T$ with $\sum_j c_j = 1, c_j \geq 0, j \in [k]$, where c_j denotes the probability that the sample belongs to class j, and k denotes the number of classes. Classification can be done by choosing the class that has the highest probability.

2.2 Vertical Federated Learning

VFL is a type of distributed ML model training approach in which two or more parties are involved in the training process, such that they hold the same set of samples with disjoint sets of features. The main goal in VFL is to train a model in a privacy-preserving manner, i.e., to collaboratively train a model without each party having access to other parties' features. Typically, the training involves a trusted third party known as the CA, and it is commonly assumed that only one party has access to the label information in the training and testing datasets. This party is named *active* and the remaining parties are called *passive*. Throughout this article, we assume that only two parties are involved; one is active and the other is passive. The active party is assumed to be *honest but curious*, i.e., it obeys the protocols exactly, but may try to infer passive features based on the information received. As a result, the active party is exchangeably referred to as the *adversary* in this article.

In the existing VFL frameworks, CA's main task is to coordinate the learning process once it has been initiated by the active party. During the training, CA receives the intermediate model updates from each party, and after a set of computations, backpropagates each party's gradient updates, separately and securely. To meet the privacy requirements of parties' datasets, cryptographic techniques such as **secure multi-party computation** (**SMC**) [13] or **homomorphic encryption** (**HE**) [14] are used.

Once the global model is trained, upon the request of the active party for a new record prediction, each party computes the results of their model using their own features. CA aggregates these results from all the parties, obtains the prediction vector (confidence scores), and delivers that to the active party for further action. Throughout this article, we assume that the active party is not allowed to make any query about the scores of the samples in the training set. This is justified due to the fact that

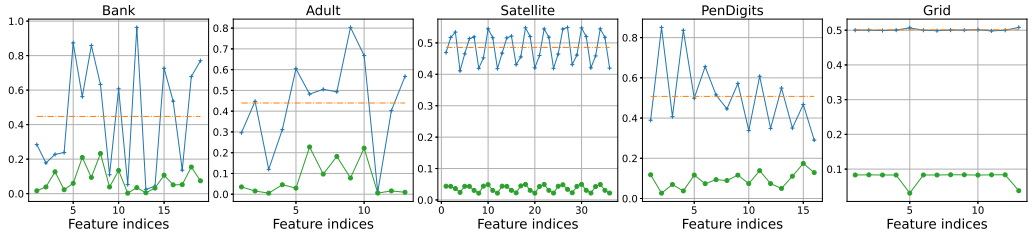


Fig. 2. Empirical mean, variance, and mean of the means of features.

Table 1. Details of the Datasets

Dataset	#Features	#Classes	#Records
Bank	19	2	41188
Adult	13	2	48842
Satellite	36	6	6430
PenDigits	16	10	10992
Grid	13	2	10000

the active party already has the ground truth labels of the training set. Note that there could be some edge cases where confidence scores of all the samples (both in the training and prediction phases) are required, such as when the active party aims at forming a soft clustering among all the samples via Gaussian mixture model to determine a high-confidence region within each class.[4] Where this is the case, i.e., exact or distorted versions of attack targets' scores are available at the active party, inference attack methods studied in [12] could be applied to reconstruct the passive features.

As in [11, 12], we assume that the active party has no information whatsoever about the underlying distribution of passive features. However, it is assumed that the knowledge about the names, types and ranges of passive features is available to the active party to decide whether to participate in a VFL or not.

3 Datasets and Model

Datasets. In this article, we perform a comprehensive evaluation of our proposed methods on five real-world and publicly available datasets, namely Bank, Adult salaries, Pen-based handwriting digits, Electrical grid stability and Satellite. These datasets have been widely used in the literature and have been obtained from the ML Repository website [15]. They have been selected to cover a wide range of classification tasks, including both binary and multi-class classification. The details of each of these datasets are provided in Table 1. Furthermore, to gain further insights into the characteristics of these datasets, we provide a visual representation of some of their empirical statistics in Figure 2.

In the Bank dataset, there are a total of 20 features. However, as stated in the dataset's description file on the ML Repository website [15], the 11th feature has a significant impact on the output target and should be disregarded in order to obtain a more accurate predictive model. As a result, the training process in our study is based on 19 features, as detailed in Table 1. Additionally, this dataset has 10 categorical features which need to be handled properly by the predictive model. Common ways to do this include one-hot encoding of the categorical features, mapping ordinal values to

[4]For instance, in the Bank-FinTech example, the bank may wish to grant credit approval for clients with low entropy confidence scores due to budget limitations.

each category, mapping categorical values to their statistics, and so on. In this article, we have chosen to map each categorical feature to its average within each class. This approach has been found to be effective for models such as **Logistic Regression** (**LR**) and **Neural Networks** (**NN**).

Model. In this article, we focus on LR, which can be modelled as

$$\mathbf{c} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{2}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the parameters collectively denoted as $\boldsymbol{\theta}$, and $\sigma(\cdot)$ is the softmax function. Each party holds their parameters corresponding to their local features. The VFL model is trained in a centralized manner, which is a reasonable assumption according to [11]. It is assumed that no intermediate information is revealed during the training phase, and only the final model is disclosed.

As in [11], the feature values in each dataset are normalized into $[0, 1]$. It is important to note that normalizing the entire dataset, including both the training and test data, may lead to an overly optimistic model accuracy, a phenomenon known as *data snooping* [16]. This effect occurs when working with very noisy datasets and should be avoided in such cases. However, in our experiments, we have not considered this issue as the datasets under consideration do not seem excessively noisy.

Each dataset has been divided into 80% training data and 20% prediction data. The training set which is the one used for VFL model training is again split into 80% training and 20% test data. Data splitting is done using train_test_split in the scikit-learn package. In training LR, we apply early stopping, and the training is done without considering any regularization. ADAM optimization is used for training, and the codes, which are in PyTorch, are available online in our GitHub repository [17].

Baselines. Depending on the number of unknown features compared to the number of classes in a dataset, two different scenarios are considered. To approach these cases we use two different inference attack (estimation) methods, namely the least squares and half* methods. The latter is proposed in [12] which is also briefly explained in Section 4.1.

4 Agnostic Inference Attack

4.1 Preliminaries

Let $(\mathbf{Y}^T, \mathbf{X}^T)^T$ denote random d_t-dimensional input samples, where the $(d_t - d)$-dimensional $\mathbf{Y}$, and the d-dimensional $\mathbf{X}$, correspond to the feature values held by the active and passive parties, respectively. We denote the samples in the training and prediction phase by the subscripts t and p, respectively, e.g., $\mathbf{X}_p$ represents the features of a sample in the prediction phase held by the passive party.[5]

The VFL model under consideration is LR, where the confidence score is given by $\mathbf{c} = \sigma(\mathbf{z})$ with $\mathbf{z} = \mathbf{W}_{\text{act}}\mathbf{Y} + \mathbf{W}_{\text{pas}}\mathbf{X} + \mathbf{b}$. Denoting the number of classes in the classification task by k, $\mathbf{W}_{\text{act}}$ (with dimension $k \times (d_t - d)$) and $\mathbf{W}_{\text{pas}}$ (with dimension $k \times d$) are the model parameters of the active and passive parties, respectively, and $\mathbf{b}$ is the k-dimensional bias vector. From the definition of $\sigma(\cdot)$, we have

$$\ln \frac{c_{m+1}}{c_m} = z_{m+1} - z_m, \; m \in [k-1], \tag{3}$$

where c_m, z_m denote the mth component of $\mathbf{c}, \mathbf{z}$, respectively. Define $\mathbf{J}$ as

$$\mathbf{J} \triangleq \begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & -1 & 1 \end{bmatrix}_{(k-1)\times k}, \tag{4}$$

[5]When no distinction is required, we use the generic forms $\mathbf{Y}, \mathbf{X}$.

whose rows are cyclic permutations of the first row with offset equal to the row index−1. By multiplying both sides of $\mathbf{z} = \mathbf{W}_{\text{act}}\mathbf{Y} + \mathbf{W}_{\text{pas}}\mathbf{X} + \mathbf{b}$ with $\mathbf{J}$, and using Equation (3), we get

$$\mathbf{J}\mathbf{W}_{\text{pas}}\mathbf{X} = \mathbf{J}\mathbf{z} - \mathbf{J}\mathbf{W}_{\text{act}}\mathbf{Y} - \mathbf{J}\mathbf{b} \tag{5}$$

$$= \mathbf{c}' - \mathbf{J}\mathbf{W}_{\text{act}}\mathbf{Y} - \mathbf{J}\mathbf{b}, \tag{6}$$

where $\mathbf{c}'$ is a $(k-1)$-dimensional vector whose mth component is $\ln \frac{c_{m+1}}{c_m}$. Denoting the RHS of Equation (6)) by $\mathbf{b}'$, Equation (6) writes in short as $\mathbf{A}\mathbf{x} = \mathbf{b}'$, where $\mathbf{A} \triangleq \mathbf{J}\mathbf{W}_{\text{pas}}$.

The white-box setting refers to the scenario where the adversary is aware of $(\mathbf{W}_{\text{act}}, \mathbf{W}_{\text{pas}}, \mathbf{b})$ and the black-box setting refers to the context in which the adversary is only aware of $\mathbf{W}_{\text{act}}$. In this article, we are interested in a scenario where the active party wishes to reconstruct passive features of samples whose confidence scores $\mathbf{c}$ have not been observed yet (which encompasses all the samples in the training phase as well). One measure by which the attack performance can be evaluated is the *mean squared error* per feature, i.e.,

$$\text{MSE} = \frac{1}{d}\mathbb{E}\left[\|\mathbf{X} - \hat{\mathbf{X}}\|^2\right], \tag{7}$$

where $\hat{\mathbf{X}}$ is the adversary's estimate. Let N denote the number of samples whose passive features have been estimated. Assuming that these N samples are produced in an i.i.d. manner, ***Law of Large Numbers*** (**LLN**) allows to approximate MSE by its empirical value $\frac{1}{Nd}\sum_{i=1}^{N}\|\mathbf{X}_i - \hat{\mathbf{X}}_i\|^2$, since the latter converges almost surely to Equation (7) as N grows.

Remark 1 (On the use of MSE as a privacy metric). The MSE in Equation (7) is used as a measure of privacy leakage through feature reconstruction. Although MSE is not a formal privacy definition, it admits a clear operational interpretation in the present setting: smaller MSE corresponds to more accurate reconstruction of the passive party's features and hence higher inference risk. This choice can be theoretically motivated through information-theoretic connections. In particular, the mutual **information–minimum mean square error** (**I–MMSE**) relationship [18] establishes that estimation distortion and information leakage are fundamentally linked. Moreover, it is shown in [19] that differential privacy can be interpreted as a constraint on mutual information between private data and released outputs. Taken together, these results suggest that controlling reconstruction MSE implicitly limits information leakage, providing a meaningful proxy for privacy under the considered adversarial model.[6]

4.2 Adversary Model (AM)

Recall that we study a scenario where prediction scores (exact forms or other) of attack targets are not available at the active party. When evaluating the potential inference attacks within this context, it is important to consider the passive party's vulnerabilities due to not just the white-box setting, but also the availability of data to the active party. This aspect of the scenario (the active party's data) can play a significant role in the success or failure of an attack. The active party's access to data and the quality of that data can greatly impact their ability to carry out an effective inference attack. Despite the limitations imposed by the lack of access to prediction scores, the active party can still utilize the data and resources available to them to conduct an inference attack on passive features during both the training and prediction phases. To achieve this, the active party can build an independent AM using the training data at their disposal. This allows the active party

[6]We emphasize that MSE does not provide worst-case or distribution-independent guarantees and does not capture all aspects of privacy. Nevertheless, it is appropriate here because it aligns with prior work on inference attacks in VFL [11, 12], admits analytical characterization, and enables explicit optimization of the privacy–interpretability tradeoff.

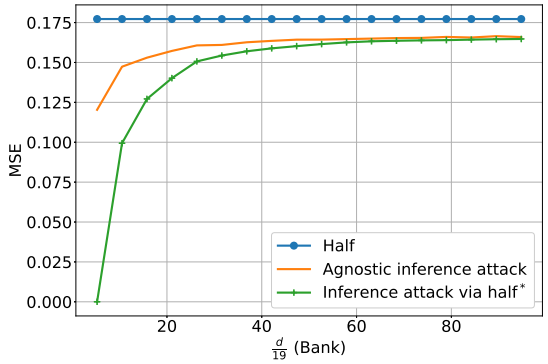


Fig. 3. Comparison of MSE per feature obtained from half estimation, agnostic inference attack and half* method in [12].

to estimate the confidence scores of their target, empowering them to carry out a successful attack. To build an AM, the active party trains a classifier, denoted by $f_{\boldsymbol{\theta}_a} : \mathcal{Y} \to \mathcal{U}$, using only the active features, independently of the passive party and CA. This is feasible as the active party already has the ground truth labels for their training set. The parameters of the AM are obtained through a training process applied over the active party's dataset, via the following minimization

$$\min_{\boldsymbol{\theta}_a} \frac{1}{n_t} \sum_{i=1}^{n_t} H(f_{\boldsymbol{\theta}_a}(\mathbf{y}_{t,i}), u_i) + \omega(\boldsymbol{\theta}_a), \tag{8}$$

where $H(\cdot,\cdot)$ is the cross entropy loss function, n_t is the number of training samples, and $\mathbf{y}_{t,i}$ is the active features of the ith sample in the training set. $\boldsymbol{\theta}_a$ denotes the parameters of AM.

Training an independent classifier by the active party may appear to be a useful method to evaluate the improvement in performance due to collaboration with the passive party when building the VFL model. However, this approach can also be used for malicious purposes. For example, the active party can use the function $f_{\boldsymbol{\theta}_a}$ to estimate confidence scores, denoted by $\hat{\mathbf{c}}$, without even making a prediction query from CA. These estimates can then be used to conduct inference attacks on passive features of any training or unseen sample. This can be achieved by forming a new set of equations using the steps outlined in Equations (3) to (6), with $\mathbf{c}$ being replaced with $\hat{\mathbf{c}}$. As discussed in further detail in Section 4.3, with this new set of equations, the adversary can use known methods for inference attacks [12] to estimate the passive features. Needless to say that this puts all samples in the training set and samples whose scores have yet to be observed at risk of privacy breaches.

Postponing the technical analysis to Section 4.3, in Figure 3 the performance of agnostic inference attack (solid line) on the bank dataset is compared with two other baselines, namely the half (green solid+ line) and half* (blue solid• line) estimations. In the former, the passive features are estimated as half (recall that features are normalized into $[0, 1]$). In the latter, the half* estimation method, proposed in [12], is used. As it is observed from the results in Figure 3, even in the absence of targets' confidence scores, the adversary manages to conduct an inference attack comparably well, especially for larger values of d where the difference with the normal inference attack is negligible.

4.2.1 Refined AM. To further enhance the accuracy of the AM in estimating the targets' confidence scores, the active party can leverage the prediction scores received from CA before carrying out an inference attack on a specific target. These scores are generated by applying the VFL model

that has been trained on the combination of both active and passive features. As a result, they carry a higher level of confidence compared to the scores generated from $f_{\boldsymbol{\theta}_a}$. To take advantage of this increased reliability, the active party could adopt a training approach where the objective loss function incorporates the scores received from the CA. This can be done in a way that during the AM training, a higher cost is incurred in the event of a mismatch between the refined AM's scores and the scores received from CA.

The refined AM, denoted as $f_{\boldsymbol{\theta}_a^r}$, is trained on the active party's available data by solving the following optimization problem

$$\min_{\boldsymbol{\theta}_a^r}\Bigg(\frac{1}{n_t}\sum_{i=1}^{n_t} H(f_{\boldsymbol{\theta}_a^r}(\mathbf{y}_{t,i}), u_{t,i}) + \frac{\alpha}{n_p}\sum_{j=1}^{n_p} S\Big(f_{\boldsymbol{\theta}_a^r}(\mathbf{y}_{p,j}), \mathbf{c}_j\Big) + \omega(\boldsymbol{\theta}_a^r)\Bigg), \tag{9}$$

where $\boldsymbol{\theta}_a^r$ denotes the parameters of the refined AM. n_p denotes the total number of prediction samples (whose scores have already been delivered to the active party). The hyperparameter α is used to stress the level of reliability of the prediction scores from CA as compared to the ones obtained by feeding the corresponding active features into AM. Additionally, the scoring function, $S(\mathbf{c}^{'}, \mathbf{c})$, is defined as $S(\mathbf{c}^{'}, \mathbf{c}) = \sum_{i=1}^{k} \log^2(c_i^{'}/c_i)$.

Remark 2. The choice of the function $S(\cdot,\cdot)$ in Equation (9) is motivated by the desire to model the similarity (or soft mismatch) between the confidence scores delivered by CA and the ones produced by AM. The function $S(\mathbf{c}^{'}, \mathbf{c})$ is a quasiconvex [7] function that attains its minimum value of zero when $\mathbf{c}^{'} = \mathbf{c}$. By setting $\alpha > 0$, one can adjust the significance of samples during the prediction phase. This is essentially equivalent to enforcing the AM to overfit on the prediction confidence scores received from CA. The value of the hyperparameter α in Equation (9) must be determined by the active party without knowledge of the actual values of the passive features. In our numerical results we consider $\alpha = 1$.

4.3 Performance Analysis

Upon agreement on a trained VFL model, during the prediction phase, the active party receives corresponding confidence scores (or a modified version of them, such as a noisy or rounded version, or the corresponding class label). In the white-box setting, the active party can construct a system of linear equations as described in Equation (6), where the unknown variables are the passive features. If the number of unknowns, d, is equal to or less than the number of equations, $k-1$, and the confidence scores are provided in their exact form, the active party can recover the passive features by solving the linear system, $\mathbf{AX} = \mathbf{b'}$. However, if the confidence scores have been altered to satisfy privacy requirements, or if the number of unknowns, d, is greater than the number of equations, $k-1$, causing $\mathbf{A}$ to not have full column rank, there may exist either a different solution or an infinite number of solutions for the linear system, respectively.

A closer examination of Equation (6) poses the following question: In a white-box setting, to what extent does the lack of confidence scores from the attack targets contribute to the average reconstruction error in Equation (7)? Or equivalently, can an inference attack still be performed effectively in the absence of the actual confidence score? In order to answer these questions, we evaluate the performance of an attack when an estimate of a confidence score obtained from AM is used instead of the actual value. To that end, we consider two scenarios: (i) $d < k$, and (ii) $d \geq k$.

4.3.1 $d < k$. In this case, we have an overdetermined system of linear equations, i.e., we have more equations ($k-1$) than unknowns (d). An overdetermined system can have solutions in some

[7] A function is quasi-convex if its sublevel sets are convex.

cases, e.g., when some equations are linear combinations of others. In the context of this article, an example of an overdetermined system with a unique solution is when the active party has all the ingredients needed to solve Equation (6). In this case, no matter which d equations are chosen out of $k-1$, it yields the same solution. However, depending on how many of the equations are independent of each other, we can have other cases with no solution or infinite solutions. The case of interest where the active party does not have the attack target's confidence score, is an example of an overdetermined system with no solution. In this case, approximate solutions can be obtained via the method of ordinary least squares, i.e., $\min_{\mathbf{X}} \|\mathbf{AX} - \hat{\mathbf{b}}'\|$, the solution of which can be written with the normal equations as [8]

$$\hat{\mathbf{X}}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{b}}'. \tag{10}$$

With Equation (10), an approximate solution is found when no exact solution exists ($\hat{\mathbf{b}}' \neq \mathbf{b}'$), and it gives an exact solution when one does exist (i.e., when $\hat{\mathbf{b}}' = \mathbf{b}'$).

Assuming that the adversary has access to an estimate of a confidence score, denoted by $\hat{\mathbf{c}}'$, different from the one in Equation (6), an approximate solution denoted by $\hat{\mathbf{X}}_{\text{LS},a}$ is obtained as below [9]

$$\hat{\mathbf{X}}_{\text{LS},a} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\hat{\mathbf{c}}' - \mathbf{J}\mathbf{W}_{\text{act}}\mathbf{Y} - \mathbf{J}\mathbf{b}) \tag{11}$$

$$= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{b}' + \hat{\mathbf{c}}' - \mathbf{c}'). \tag{12}$$

The MSE of the estimation in Equation (12) reads as (see Appendix A for the derivation of (13))

$$\text{MSE}(\hat{\mathbf{X}}_{\text{LS},a}) = \frac{1}{d}\text{Tr}\left(\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-2}\mathbf{A}^T\mathbf{K}_{\mathbf{c}',\hat{\mathbf{c}}'}\right), \tag{13}$$

where $\mathbf{K}_{\mathbf{c}',\hat{\mathbf{c}}'} \triangleq \mathbb{E}[(\hat{\mathbf{c}}' - \mathbf{c}')(\hat{\mathbf{c}}' - \mathbf{c}')^T]$. From Equation (13), it is easy to note that when the active party has the exact form of the scores, the MSE in this case is zero, i.e., perfect reconstruction. Additionally, note that Equation (13) is non-negative due to the non-negativity of MSE.

4.3.2 $d \geq k$. In [12], several inference attack methods were evaluated, including RCC1, RCC2, CLS, and half*. These methods were found to significantly improve the adversary's estimation quality, resulting in lower MSE compared to the methods discussed in [11, 20]. While the results and discussions for agnostic inference attack in this article are general and apply to any of the feature estimation methods, for the purposes of this article, we specifically focus on using the half* method. This is because, on one hand, the half* method is analytically feasible, making it easier to study. On the other hand, as indicated in [12, Figure 4], the half* reconstruction method was found to have the best performance, or at least performance comparable to the best method. Thus, before proceeding with our analysis, we briefly outline the half* estimation method.

Any solution of an underdetermined system of linear equations ($d \geq k$) can be written as $\mathbf{A}^+\mathbf{b}' + (\mathbf{I}_d - \mathbf{A}^+\mathbf{A})\mathbf{w}$ for some $\mathbf{w} \in \mathbb{R}^d$, where $\mathbf{A}^+$ denotes the pseudoinverse of $\mathbf{A}$ satisfying the Moore–Penrose conditions[21].[10] As elaborated further in [12], when the only available information about $\mathbf{X}$ is that it belongs to $[0,1]^d$, then $\hat{\mathbf{X}}_{\text{half}} = \frac{1}{2}\mathbf{1}_d$ is optimal in the best-worst sense. Indeed, the naive estimate of $\frac{1}{2}\mathbf{1}_d$ is the Chebyshev center of $[0,1]^d$. The adversary can perform better when the side information $(\mathbf{b}', \mathbf{A})$ is available. The half* scheme is built on top of the naive $\frac{1}{2}\mathbf{1}_d$

[8] Throughout the article, $\hat{\mathbf{c}}'$ is adversary's estimation of $\mathbf{c}'$, and accordingly, $\hat{\mathbf{b}}'$ is an approximation of $\mathbf{b}'$ induced by $\hat{\mathbf{c}}'$.
[9] Throughout the article, subscript a in estimations such as $\hat{\mathbf{X}}_{\text{LS},a}$ is used to emphasize that the attack is agnostic.
[10] When $\mathbf{A}$ has linearly independent rows, we have $\mathbf{A}^+ = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$.

estimation as follows. The estimator finds a solution in the solution space that is closest to $\frac{1}{2}\mathbf{1}_d$. In [12, Proposition 1] it is shown that

$$\hat{\mathbf{X}}_{\text{half}^*} = \mathbf{A}^+\mathbf{b}' + \frac{1}{2}(\mathbf{I}_d - \mathbf{A}^+\mathbf{A})\mathbf{1}_d. \tag{14}$$

As discussed in Section 4.2, although the confidence score is not available to the adversary, they manage to obtain an estimate of it via AM. Under these circumstances, to obtain an estimate of $\mathbf{c}'$, the adversary follows the lines similar to Equations (3)–(6). We have

$$\begin{aligned}\mathbf{J}\mathbf{W}_a\mathbf{Y} &= \mathbf{J}\mathbf{z}_a - \mathbf{J}\mathbf{b}_a \\ &= \hat{\mathbf{c}}' - \mathbf{J}\mathbf{b}_a,\end{aligned} \tag{15}$$

where the subscript a in $\mathbf{W}_a, \mathbf{b}_a, \mathbf{z}_a$ is to clarify the attribution of these parameters to AM. Noting that $\hat{\mathbf{c}}'$ is an estimate of $\mathbf{c}'$, the adversary conducts agnostic inference attack using half* method. Denoting this estimation by $\hat{\mathbf{X}}_{\text{half}^*,a}$, we have

$$\hat{\mathbf{X}}_{\text{half}^*,a} = \mathbf{A}^+(\hat{\mathbf{c}}' - \mathbf{J}\mathbf{W}_{\text{act}}\mathbf{Y} - \mathbf{J}\mathbf{b}) + \frac{1}{2}(\mathbf{I} - \mathbf{A}^+\mathbf{A})\mathbf{1}_d \tag{16}$$

$$= \hat{\mathbf{X}}_{\text{half}^*} + \mathbf{A}^+(\hat{\mathbf{c}}' - \mathbf{c}'). \tag{17}$$

The MSE resulted from the estimation in Equation (17) reads as (see Appendix B for the derivation of (18))

$$\text{MSE}(\hat{\mathbf{X}}_{\text{half}^*,a}) = \frac{1}{d}\text{Tr}\left(\left(\mathbf{I} - \mathbf{A}^+\mathbf{A}\right)\mathbf{K}_{\frac{1}{2}\mathbf{1}}\right) + \frac{1}{d}\text{Tr}\left(\mathbf{A}^{+T}\mathbf{A}^+\mathbf{K}_{\mathbf{c}',\hat{\mathbf{c}}'}\right). \tag{18}$$

From Equation (18) it is clear that the adversary's ability to carry out agnostic inference attack is dependent on the accuracy of the estimation of a target confidence score, as reflected through the term $\mathbf{K}_{\mathbf{c}',\hat{\mathbf{c}}'}$. The adversary's performance in this scenario cannot surpass the performance obtained when the exact form of confidence scores is known (see Appendix B). Note that, neither the active party nor the passive party have access to the exact form of the confidence scores or the parameters of AM, respectively, making it impossible for them to determine the exact value of Equation (18). Despite this, we utilize Equation (18) in Section 6 to obtain our numerical results when evaluating the adversary's performance.

It is worth mentioning here that the replacement of $\mathbf{c}'$ with $\hat{\mathbf{c}}'$ in Equation (6) may result in some components of the final solution falling outside of the feasible region. To handle such cases, there are multiple strategies that can be implemented, including but not limited to: (i) Mapping the individual components of the solution to the closest value of 0 or 1 if they are outside of the feasible region, (ii) Mapping those individual components to the value of 0.5, or (iii) Mapping all the passive features to 0.5 once an estimated feature of a sample falls outside the feasible region. The best strategy depends on the underlying data statistics and requires deeper knowledge of the data which is indeed not available at the active party. Therefore, in order to ensure fairness in our numerical results, this article adopts the first strategy in all datasets used in our experiments.

Remark 3 (Why LR and Extension to NN). We focus on LR as the primary model of interest for three reasons. First, LR remains widely deployed in VFL applications where interpretability and auditability are essential, such as finance, healthcare, and risk assessment [22–25]. Second, LR admits a tractable analytical characterization, which enables explicit derivations of feature reconstruction error and principled optimization-based PPSs. Third, softmax LR constitutes the final prediction layer in many NN architectures, making the notion of confidence scores and their approximation via AMs directly relevant beyond pure LR. To elaborate further on this, in the

following of this Remark, we consider a thought experiment on shallow NNs. Consider a shallow NN with two fully connected layers and $\tanh(\cdot)$ activations,

$$c = \text{softmax}(W_2 \tanh(W_1 \tilde{x} + b_1) + b_2),$$

where $\tilde{x} = [y^\top, x^\top]^\top$ concatenates active and passive features. In this setting, the active party can still train an AM or RAM using only local features and labels (and possibly observed scores) to approximate the confidence scores c. However, unlike LR, the mapping from passive features x to the logits becomes nonlinear. If the activation function is invertible (e.g., tanh), and the passive contribution enters the network in a structurally identifiable manner, feature reconstruction could in principle be formulated as a nonlinear inverse problem. In contrast, for non-invertible activations such as ReLU, the many-to-one mapping introduces intrinsic ambiguity, which can be interpreted as an additional source of distortion that may further degrade reconstruction accuracy.[11] A systematic treatment of these aspects is left for future work.

Remark 4 (Collusion and Auxiliary Information). Collusion between the active party and the CA or other parties can be modeled as an enlarged semi-honest adversarial coalition [26–28], where parties follow the protocol but share their views. In this case, the active party may gain access to additional side information, such as partial passive features or additional confidence scores. Such auxiliary information can be incorporated into the RAM by augmenting the training objective with additional alignment or regularization terms, similar in spirit to auxiliary-data-based learning approaches [20, see Algorithm 2]. Naturally, increased side information can improve the quality of confidence score approximation and, consequently, the effectiveness of inference attacks. We emphasize that adversarial models involving actively malicious behavior or protocol deviations by the CA fall outside the honest-but-curious threat model assumed in this article and are left for future investigation.

Remark 5 (Motivation and implications of agnostic attacks). In many real deployments—such as credit scoring (see Figure 1), medical-risk prediction, or e-commerce recommendation—the passive features may include sensitive financial, behavioural, or clinical attributes that the passive party is unwilling to reveal directly. Using agnostic attacks, even when confidence scores corresponding to the target samples are not revealed, the active party can still reconstruct passive features using only its local data and model outputs. If the reconstructed features correlate strongly with sensitive attributes (e.g., income proxies in credit scoring or comorbidity indicators in medical diagnosis), this may lead to concrete privacy harms such as profiling, discrimination, or unintended leakage of protected characteristics. The effectiveness of agnostic attacks depends primarily on the statistical structure of the joint feature space. The risk is highest when (i) the active party holds a large portion of the predictive signal (labels and moderately informative features), and (ii) strong correlation exists between active and passive features. In such regimes, the AM and RAM can approximate score behaviour sufficiently well to support accurate linear reconstruction of passive features, even without access to the true confidence scores. This will be elaborated further in Section 6.2.

5 Privacy Against Agnostic Inference Attack

In the literature a number of algorithms have been proposed for privacy in the context of VFL in the white-box setting, such as those proposed in [11, 12, 20]. The majority of these algorithms have attempted to alter the confidence scores delivered to the active party in some way. However, as demonstrated in Figure 3 and discussed in Section 4, simply modifying the confidence scores is

[11] For ReLU activations, the loss of injectivity implies that multiple passive feature values can yield identical activations, even when confidence scores are known.

not enough to defend against agnostic inference attacks, where the adversary can build a separate classifier and improve their attack as they collect more scores from the CA. As such, it becomes clear that this approach is insufficient and leaves the passive party vulnerable to such attacks.

The central importance of the passive party's parameters in any potential adversarial attack highlights the need for effective defense techniques that specifically target these parameters. A straightforward approach would be to implement a black box setting, in which the active party is not privy to the passive party's parameters, $\mathbf{W}_{\text{pas}}$. However, this approach sacrifices interpretability of the VFL model, which is a critical aspect in many applications where the client or user demands a clear explanation for the decisions made by the machine. This creates a tension between the active party, who desires a white-box setting for interpretability purposes, and the passive party, who values privacy and prefers a black box setting. Therefore, there is a need for a more nuanced and balanced approach to designing PPSs in the context of VFL, one that strikes a delicate balance between privacy and interpretability, and minimizes the cost of collaboration for all involved parties. The following sections present an effort towards realizing this elusive goal of balancing privacy and interpretability in VFL.

The question we aim to address is: Can a PPS be designed that creates a balance between preserving the privacy of the passive party and enabling the interpretability capabilities of the active party? To achieve this, we take into account the potential for agnostic inference attacks by an adversary using either Equation (12) or (16) in different scenarios. Our proposed PPSs aim to systematically manipulate the parameters $\mathbf{W}_{\text{pas}}$ in order to partially address the concerns of both parties and provide a mutually acceptable outcome.

In the following, in Section 5.1, we first review a particular case of optimization algorithms. Additionally, we provide an informal discussion on interpretability of parameters in LR. Then, in Section 5.2, in a case-by-case fashion, we cover the technical analysis of the proposed PPSs.

Remark 6 (Black-box Passive Parameters). The inference attacks analyzed in this article rely on explicit access to the passive party's parameters in order to construct linear systems for feature reconstruction. In a black-box setting, where the active party does not know the passive parameters, this algebraic reconstruction mechanism is no longer directly applicable. As a result, both reconstruction accuracy and interpretability at the active party are substantially reduced, while privacy for the passive party is strengthened. Although black-box access does not preclude other forms of attacks (e.g., model extraction or surrogate learning via adaptive queries), such attacks rely on fundamentally different assumptions and techniques. Accordingly, this work focuses on the white-box regime, where the privacy–interpretability tradeoff is most pronounced and analytically tractable.

5.1 Preliminaries

5.1.1 Optimization Problems with Orthogonality Constraint. Here, we provide a brief overview of an algorithm employed in this article to determine the well-performing design parameters of one of the proposed PPSs. This algorithm is a specific instance of a larger class of optimization problems that incorporate an orthogonality constraint, specifically optimization problems over Stiefel manifolds. While this particular case is reviewed in the sequel, for a more in-depth and comprehensive understanding of the geometry of the problem and the solver algorithms, we refer the readers to [29] and [30, 31], respectively.

Consider the following optimization problem with an orthogonality constraint:

$$\min_{\substack{\mathbf{R}:\\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_d}} \quad f(\mathbf{R}), \tag{19}$$

where $f : \mathbb{R}^{d\times d} \to \mathbb{R}$ satisfies the so-called *blanket assumption*, i.e., the function $f(\mathbf{R})$ is differentiable and $\frac{\partial f}{\partial \mathbf{R}}$ is Lipschitz continuous.[12] The orthogonality constraint in (19) can be expressed as $\mathcal{S}_d = \{\mathbf{R} \in \mathbb{R}^{d\times d} | \mathbf{R}^T\mathbf{R} = \mathbf{I}_d\}$, and is denoted as Stiefel manifold in real matrix space.

Remark 7. The problem outlined in Equation (19) plays a crucial role in various applications, such as "discretized Kohn-Sham energy minimization" and "unsupervised feature selection." There are numerous methods available to address this problem, the majority of which utilize a trust-region strategy with quadratic approximation. In our numerical experiments, we utilize the PySTOP module based on the methods described in [30, 31].

Almost all the algorithms tackling problems of the type Equation (19) require gradient of the objective function $f(\mathbf{R})$ wrt the set $\mathcal{S}_d$. Denoting this gradient by $\nabla_s f(\mathbf{R})$, it is shown in [29, Equation (2.53)] that it is of the form

$$\nabla_s f(\mathbf{R}) = \frac{\partial f}{\partial \mathbf{R}} - \mathbf{R}\left(\frac{\partial f}{\partial \mathbf{R}}\right)^T \mathbf{R}. \tag{20}$$

One can view Equation (20) as correcting the nominal gradient $\frac{\partial f}{\partial \mathbf{R}}$ by subtracting off its projection, i.e., $(\frac{\partial f}{\partial \mathbf{R}})^T\mathbf{R}$ onto the current solution $\mathbf{R}$. This ensures that the corrected gradient $\nabla_s f(\mathbf{R})$ is tangent to the manifold.

In the following, we provide an analytically tractable example of this optimization Equation (19) in the context of PPSs in VFL. As it will be elaborated later in Section 5.2, providing analytical solutions for more interesting problems are still open if not infeasible. Therefore, in such cases we resort to iterative solver algorithms.

Example 1. PPS for the least squares method: We aim at designing a PPS against an adversary who uses a naive least squares method. To that end, the passive parameters $\mathbf{W}_{\text{pas}}$ are transformed via a secret orthonormal matrix $\mathbf{R}$ ($\mathbf{R}^T\mathbf{R} = \mathbf{I}$), and it is the transformed version $\mathbf{W}_{\text{pas}}\mathbf{R}$ which is revealed to the active party. The adversary, having only the transformed parameters, forms the following system of equations $\mathbf{A}_n\mathbf{X} = \mathbf{b'}$ where $\mathbf{A}_n = \mathbf{J}\mathbf{W}_{\text{pas}}\mathbf{R} = \mathbf{A}\mathbf{R}$.[13] Using least squares estimation method, the adversary obtains the estimated features as $\hat{\mathbf{X}}^{\text{PPS}}_{\text{LS},a} = \mathbf{A}_n^+\mathbf{b'}$.[14] The MSE resulted from least squares inference attack when PPS is in place reads as (see Appendix C for the derivation of (21))

$$\text{MSE}(\hat{\mathbf{X}}^{\text{PPS}}_{\text{LS}}) = \text{Tr}(\mathbf{I} + \mathbf{A}^+\mathbf{A}) - 2\text{Tr}(\mathbf{R}\mathbf{K}_\mathbf{0}\mathbf{A}^+\mathbf{A}). \tag{21}$$

To maximize $\text{MSE}(\hat{\mathbf{X}}^{\text{PPS}}_{\text{LS}})$, and hence deteriorate the adversary's attack performance, we equivalently solve the following optimization problem:

$$\min_{\substack{\mathbf{R}\\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_d}} \; f^{\text{PPS}}_{\text{LS},a}(\mathbf{R}) \triangleq \text{Tr}(\mathbf{R}\mathbf{K}_0\mathbf{A}^+\mathbf{A}). \tag{22}$$

[12] Assume two metric spaces $(\boldsymbol{\mathcal{R}}, d_{\mathbf{R}})$ and $(\mathcal{X}, dx)$, where $d_{\mathbf{R}}$ and dx denote the metrics on $\boldsymbol{\mathcal{R}}$ and $\mathcal{X}$, respectively. A function $f : \boldsymbol{\mathcal{R}} \to \mathcal{X}$ is called Lipschitz continuous if there exists a real constant $K \geq 0$, such that for all $\mathbf{R}_1$ and $\mathbf{R}_2$ in $\boldsymbol{\mathcal{R}}$, we have

$$d_x(f(\mathbf{R}_1), f(\mathbf{R}_2)) \leq K \cdot d_{\mathbf{R}}(\mathbf{R}_1, \mathbf{R}_2)$$

.

[13] We assume that the adversary has been given or has estimated the exact form of the attack targets' confidence score.

[14] Superscript PPS is used to note that a privacy-preserving technique is in place at the time of an attack.

Noting that the objective function in (22) is differentiable, and its derivative $\frac{\partial f_{\text{LS},a}^{\text{PPS}}}{\partial \mathbf{R}} = \mathbf{A}^{+}\mathbf{A}\mathbf{K_0}$ is Lipschitz continuous,[15] we have

$$\nabla_s f_{\text{LS},a}^{\text{PPS}} = \mathbf{A}^{+}\mathbf{A}\mathbf{K_0} - \mathbf{R}\mathbf{K_0}\mathbf{A}^{+}\mathbf{A}\mathbf{R}. \tag{23}$$

Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be an SVD of $\mathbf{A}^{+}\mathbf{A}\mathbf{K_0}$. It is easily verified that there are two candidates making the Stiefel gradient in Equation (23) equal to zero, i.e., $\mathbf{R} = \pm\mathbf{U}\mathbf{V}^T$. By substituting the candidate solutions in Equation (22) we have

$$\text{Tr}(\mathbf{R}\mathbf{K_0}\mathbf{A}^{+}\mathbf{A}) = \pm\text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T) = \pm\text{Tr}(\mathbf{U}\mathbf{\Sigma}\mathbf{U}^T) = \pm\sum_{i=1}^{d}\sigma_i, \tag{24}$$

where σ_i is the ith singular value of $\mathbf{A}^{+}\mathbf{A}\mathbf{K_0}$. Noting that the singular values of a matrix are non-negative, the minimiser of Equation (22) is $\mathbf{R} = -\mathbf{U}\mathbf{V}^T$.

Remark 8. A similar problem has been considered in [12], where instead of transforming the parameters, the passive party transforms the features. In that study, the proof approach relies on Von Neumann's trace inequality, which can serve as an alternative to the optimization over Stiefel manifolds discussed in this article. However, as will be made evident in the following sections, the optimization over Stiefel manifolds has a distinct advantage in that it allows for the treatment of a wide range of more complex problems through the use of iterative algorithms.

Remark 9. It is worth noting that the example discussed in this section has a couple of limitations. Firstly, the assumption that the adversary will use the least squares method is unrealistic as there are other inference attack methods that are more advanced and efficient. Secondly, transforming the parameter (or feature space, as discussed in [12]) results in a significant and uncontrolled distortion of the parameters. This can make it difficult for the active party to interpret the results of the VFL model, and may even obstruct their ability to justify the outcomes in certain circumstances. For example, in the Bank-FinTech scenario, the bank may be required to provide a justification for rejecting a customer's credit card application. Thus, it is crucial that these limitations are taken into consideration when designing any PPS so as to ensure the active party's concerns are addressed.

5.1.2 Interpretability of the VFL Model at the Active Party. The increasing use of ML-based automation systems in sectors like government, banking, healthcare, and education presents both benefits and challenges. A significant challenge in VFL involves the interpretability of decisions made by models trained with PPSs, especially when these decisions impact legal, insurance, or health matters. A lack of outcome justifiability may erode trust between users and the system.

Interpretability is crucial in ML and pertains to understanding the relationship between a model's parameters and its outcomes. Simpler models such as linear and logistic regression, decision trees, and Lasso/Ridge regression are often favored over complex deep learning models for their interpretability when their performance is comparable. For instance, in LR, the odds ratio between two classes for a sample can be calculated using $e^{(\mathbf{w}_i-\mathbf{w}_j)\mathbf{x}_1}$, where $\mathbf{w}_i$ is the ith row of the parameter matrix $[\mathbf{W}_{\text{act}}, \mathbf{W}_{\text{pas}}]$. Consider a Bank-FinTech scenario where two customers, $\mathbf{x}_1$ and $\mathbf{x}_2$, have similar active features but differ in passive features; if one is approved and the other rejected for a credit application, the bank must justify the decision by the difference in passive features and the resultant change in the odds ratio, expressed as $e^{(w_{i,d_f}-w_{j,d_f})(x_{1,d_f}-x_{2,d_f})}$, where d_f denotes the index of a specific passive feature.

[15] $\|\mathbf{A}^{+}\mathbf{A}\mathbf{K_0} - \mathbf{A}^{+}\mathbf{A}\mathbf{K_0}\| = 0 \leq K \cdot \|\mathbf{R}_1 - \mathbf{R}_2\|$ for any $K \geq 0$ and any matrix norm.

5.2 Privacy-Preserving Schemes (PPS)

When evaluating the effectiveness of agnostic inference attacks, we employ a "genie-aided" approach where the adversary is assumed to know the exact form of the confidence scores. This assumption is necessary because the performance of agnostic inference attacks as described in Equations (13) and (18) heavily relies on the matrix $\mathbf{K}_{\mathbf{c},\hat{\mathbf{c}}'}$. Given that the passive party lacks access to $\hat{\mathbf{c}}'$, precise calculations cannot be made without this assumption. By considering a genie-aided scenario, we recognize the adversary's potential capabilities to accurately estimate $\hat{\mathbf{c}}'$, whether through their own resources or via collusion with data owners as suggested by [11]. Therefore, our objective is to mitigate the adversary's optimal attack performance under these conditions. It is advantageous for the passive party to develop PPSs that leverage the availability of $\mathbf{c}'$ at the active party to enhance privacy protection.

In designing the PPSs, our goal is to maintain the interpretability of passive parameters. To measure the distance between the accessible parameter set $\mathbf{W}_n$ and the original passive parameters $\mathbf{W}_{\text{pas}}$, we define the function

$$g(\mathbf{R}) = \frac{1}{dk}\text{Tr}\left((\mathbf{W}_{\text{pas}} - \mathbf{W}_n)^T(\mathbf{W}_{\text{pas}} - \mathbf{W}_n)\right), \tag{25}$$

where $\mathbf{W}_n$ represents the parameters given to the active party instead of $\mathbf{W}_{\text{pas}}$.[16] To achieve our objective, we consider the following optimization problem:

$$\min_{\substack{\mathbf{R}:\\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_d}} \; f(\mathbf{R}) + \lambda\left(g(\mathbf{R}) - \varepsilon\right)^2, \tag{26}$$

where $f(\mathbf{R})$ is a function such that its minimization corresponds to maximizing the overall MSE. Here, λ acts as a penalizing factor, imposing a certain level of distortion on the parameters while optimizing.

We study the PPSs in the following four distinct, disjoint cases:

(*i*) For $d \geq k > 2$, we propose a heuristic approach based on transforming the passive parameters via orthonormal matrices.
(*ii*) For $1 < d < k$, a similar heuristic approach is applied.
(*iii*) For $d = 1$, we attempt to tackle the general problem without specific assumptions on the mapping from passive parameters to those accessible to the active party.
(*iv*) For $k = 2$ and $d > 1$, the approach again addresses the general problem without particular mapping assumptions.

5.2.1 *Case* $\boldsymbol{d \geq k > 2}$.

In this case, the passive party's parameters $\mathbf{W}_{\text{pas}}$ are transformed as

$$\mathbf{W}_n = \mathbf{W}_{\text{pas}}\mathbf{R},$$

where $\mathbf{R}$ is assumed to be an orthonormal matrix, namely $\mathbf{R}^T\mathbf{R} = \mathbf{I}$.[17] Therefore, the function in Equation (25) reads as

$$\begin{aligned} g(\mathbf{R}) &= \frac{1}{dk}\text{Tr}(2\mathbf{W} - (\mathbf{R} + \mathbf{R}^T)\mathbf{W}) \\ &= \frac{2}{dk}\text{Tr}((\mathbf{I} - \mathbf{R})\mathbf{W}), \end{aligned} \tag{27}$$

[16] The function g will take different arguments in the sequel depending on the mapping considered from $\mathbf{W}_{\text{pas}}$ to $\mathbf{W}_n$.

[17] The assumption of orthonormality for the transforming matrix $\mathbf{R}$ ensures that our algorithm design remains analytically tractable. Without this assumption, solving the given problem becomes cumbersome. This complexity arises because, generally, for two matrices $\mathbf{A}_{k-1\times d}$ and $\mathbf{R}_{d\times d}$, the equality $(\mathbf{AR})^+ = \mathbf{R}^{-1}\mathbf{A}^+$ does not hold. There are some sufficient conditions under which this equality is valid, one of which is the orthonormality of $\mathbf{R}$ [32]. We leave the investigation of the general case of the transformer matrix $\mathbf{R}$ as an open problem for future research.

where Equation (27) is due to $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, $\mathrm{Tr}(\mathbf{R}^T\mathbf{W}) = \mathrm{Tr}(\mathbf{RW})$ and $\mathbf{W} \triangleq \mathbf{W}_{\text{pas}}^T\mathbf{W}_{\text{pas}}$.

Once the adversary receives the passive party's transformed parameters $\mathbf{W}_{\text{pas}}\mathbf{R}$, they *form* a system of linear equations $\mathbf{A}_n\mathbf{X} = \mathbf{b'}$,[18] where $\mathbf{A}_n = \mathbf{J}\mathbf{W}_{\text{pas}}\mathbf{R} = \mathbf{AR}$. Note that $\mathbf{b'}$ is still the same as in Equation (6). By applying half* method on this system, we have

$$\begin{aligned}\hat{\mathbf{X}}^{\text{PPS}}_{\text{half}^*,a} &= \mathbf{A}_n^+\mathbf{b'} + \frac{1}{2}(\mathbf{I} - \mathbf{A}_n^+\mathbf{A}_n)\mathbf{1}_d \\ &= \mathbf{R}^T\mathbf{A}^+\mathbf{b'} + \frac{1}{2}(\mathbf{I} - \mathbf{R}^T\mathbf{A}^+\mathbf{AR})\mathbf{1}_d. \end{aligned} \tag{28}$$

Therefore, the adversary's MSE corresponding to half* estimations reads as (see Appendix D for the derivation of (29))

$$\mathrm{MSE}(\hat{\mathbf{X}}^{\text{PPS}}_{\text{half}^*,a}) = \frac{1}{d}\mathrm{Tr}(\mathbf{K}_{\frac{1}{2}}\mathbf{1}) - \frac{1}{d}\mathrm{Tr}\left(\mathbf{A}^+\mathbf{A}\left(2\mathbf{K_0}\mathbf{R}^T - \mathbf{K_0} - \mathbf{RMR}^T\right)\right). \tag{29}$$

Our goal is to maximise the term in Equation (29) over orthonormal matrices $\mathbf{R} \in \mathcal{S}_d$, such that a certain level of interpretability is additionally preserved at the active party, i.e., $g(\mathbf{R}) = \varepsilon$. To that end, we consider the following optimization problem:

$$\min_{\mathbf{R}:\mathbf{R}\in\mathcal{S}_d} \; f^{\text{PPS}}_{\text{half}^*,a}(\mathbf{R}) + \lambda(g(\mathbf{R}) - \varepsilon)^2, \tag{30}$$

where $f^{\text{PPS}}_{\text{half}^*,a}(\mathbf{R})$ denotes the second term in Equation (29). Note that ε is a design parameter introduced into the model, potentially as part of a joint agreement between the active and passive parties. However, $\lambda \geq 0$ is an optimization tuning parameter, which should be considered large enough to enforce the equality $g(\mathbf{R}) = \varepsilon$.

Remark 10. The way the adversary obtains a system of linear equations in Equation (6) is not unique; however, all such systems can be expressed in a form similar to Equation (6) via $\mathbf{J}_{\text{new}} = \mathbf{TJ}$, with $\mathbf{T}_{(k-1)\times(k-1)}$ being an invertible matrix, secretly used by the adversary. From this standpoint, it is crucial that, regardless of the choice of $\mathbf{T}$, the MSE in Equation (29), upon which the optimization in Equation (30) is based, remains consistent with the MSE experienced by the adversary. Therefore, any designed PPS should be independent of any transformation matrix $\mathbf{T}$ used by the adversary, as otherwise, intended privacy (certain MSE) and interpretability ($g(\mathbf{R}) = \varepsilon$) concerns are not guaranteed. From the objective function in Equation (30), the independence of $g(\mathbf{R})$ is obvious. For the function $f^{\text{PPS}}_{\text{half}^*,a}(\mathbf{R})$, note that its dependency on $\mathbf{J}$ is via $\mathbf{A}^+\mathbf{A}$. Hence, if the adversary forms a new matrix $\mathbf{A}_{\text{new}} = \mathbf{J}_{\text{new}}\mathbf{W}_{\text{pas}} = \mathbf{TJW}_{\text{pas}}$, we have

$$\begin{aligned}\mathbf{A}^+_{\text{new}}\mathbf{A}_{\text{new}} &= (\mathbf{J}_{\text{new}}\mathbf{W}_{\text{pas}})^+(\mathbf{J}_{\text{new}}\mathbf{W}_{\text{pas}}) = (\mathbf{TJW}_{\text{pas}})^+(\mathbf{TJW}_{\text{pas}}) \\ &= (\mathbf{JW}_{\text{pas}})^+\mathbf{T}^{-1}\mathbf{T}(\mathbf{JW}_{\text{pas}}) = \mathbf{A}^+\mathbf{A}.\end{aligned}$$

Therefore, the solution of the objective function is independent of any orthogonal transformation of $\mathbf{J}$.

If the blanket assumption is satisfied on the objective of the minimization problem in (30), it fits within the category of the optimization problems reviewed in Section 5.1. Continuity is obvious. The following proposition ascertains the Lipschitz continuity.

Proposition 1. *The objective function in Equation* (30) *is Lipschitz continuous.*

[18]Here, we emphasize on the word "form" as this is passive party's deception of active party. Given $\mathbf{X}$, the true equation is $\mathbf{AX} = \mathbf{b'}$.

Proof. The derivatives of $f^{\text{PPS}}_{\text{half}^*,a}(\mathbf{R})$ and $(g(\mathbf{R})-\varepsilon)^2$ are obtained as below [19]

$$\frac{\partial f^{\text{PPS}}_{\text{half}^*,a}(\mathbf{R})}{\partial \mathbf{R}} = \frac{1}{d}\left(-2\mathbf{A}^+\mathbf{A}\mathbf{K_0} + \mathbf{A}^+\mathbf{A}\mathbf{R}(\mathbf{M}+\mathbf{M}^T)\right) \tag{31}$$

$$\frac{\partial}{\partial \mathbf{R}}(g-\varepsilon)^2 = -\frac{4}{dk}\left(\frac{2}{dk}\text{Tr}((\mathbf{I}-\mathbf{R})\mathbf{W}) - \varepsilon\right)\mathbf{W}, \tag{32}$$

where Equation (32) is due to $\mathbf{W} = \mathbf{W}^T$. Lipschitz continuity of Equation (31) is confirmed noting that

$$\begin{aligned}\|\mathbf{A}^+\mathbf{A}\mathbf{R}_1(\mathbf{M}+\mathbf{M}^T) - \mathbf{A}^+\mathbf{A}\mathbf{R}_2(\mathbf{M}+\mathbf{M}^T)\| &= \|\mathbf{A}^+\mathbf{A}(\mathbf{R}_1-\mathbf{R}_2)(\mathbf{M}+\mathbf{M}^T)\| \\ &\le k_{\text{half}^*}\|\mathbf{R}_1-\mathbf{R}_2\|\end{aligned} \tag{33}$$

where in Equation (33) we used the inequality $\|\mathbf{AB}\| \le \|\mathbf{A}\|\|\mathbf{B}\|$ and we have $k_{\text{half}^*} \triangleq \|\mathbf{A}^+\mathbf{A}\|\|(\mathbf{M}+\mathbf{M}^T)\|$ which is a constant by design. Note that Equation (33) is valid for any matrix norm. Lipschitz continuity of Equation (32) is also shown noting that

$$\begin{aligned}\|\text{Tr}((\mathbf{R}_1-\mathbf{R}_2)\mathbf{W} + (\mathbf{R}_1-\mathbf{R}_2)^T\mathbf{W})\mathbf{W}\| &= 2\|\text{Tr}((\mathbf{R}_1-\mathbf{R}_2)\mathbf{W})\mathbf{W}\| \\ &\le k_g\|\mathbf{R}_1-\mathbf{R}_2\|,\end{aligned} \tag{34}$$

where $k_g \triangleq 2\sigma_1\|\mathbf{W}\|$ with σ_1 being the largest singular value of the matrix $\mathbf{W}$. Additionally, in Equation (34) we used trace-norm inequality.[20] Proof of the proposition follows by noting that the sum of two Lipschitz continuous functions is Lipschitz continuous. □

5.2.2 Case $\mathbf{1 < d < k}$. In this case, the adversary uses Equation (10) to solve the formed system of equations, i.e., $\hat{\mathbf{X}}^{\text{PPS}}_{\text{LS},a} = (\mathbf{A}_n^T\mathbf{A}_n)^{-1}\mathbf{A}_n^T\mathbf{b}'$. Recall that this is a case of an overdetermined system, where a unique solution is obtained when the adversary has access to all the parameters in Equation (5). Otherwise, an approximate solution is obtained. Following similar steps as in the previous case, and noting that the adversary uses $\mathbf{W}_n = \mathbf{W}_{\text{pas}}\mathbf{R}$ for their attack, the resulting MSE reads as (see Appendix E for the derivation of (35))

$$\text{MSE}(\hat{\mathbf{X}}^{\text{PPS}}_{\text{LS},a}) = \frac{2}{d}\text{Tr}\left(\mathbf{K_0} - \mathbf{R}\mathbf{K_0}\right). \tag{35}$$

In Equation (35), we use the identity $\text{Tr}(\mathbf{R}^T\mathbf{K}_0) = \text{Tr}(\mathbf{R}\mathbf{K}_0)$. By calculating the Stiefel gradient of Equation (35) and following the steps outlined in Section 5.1.1, it is straightforward to verify that in the absence of interpretability constraints, the optimal transformation matrix is $\mathbf{R} = -\mathbf{I}$. This implies that the passive party simply changes the sign of their parameters. To also address the interpretability concerns of the active party, we form the following optimization problem:

$$\min_{\mathbf{R}:\mathbf{R}\in\mathcal{S}_d} \ \frac{2}{d}\text{Tr}(\mathbf{R}\mathbf{K}_0) + \lambda\left(g(\mathbf{R})-\varepsilon\right)^2. \tag{36}$$

It is easily verified that the problem in Equation (36) satisfies the blanket assumption.

Remark 11. Note that when the active party has no interpretability concerns, it would be clear to them that the transformation matrix is $\mathbf{R} = -\mathbf{I}$. In such cases, they could easily recover the passive party's features by simply reversing the sign of the parameters. One potential solution to mitigate this issue is to maintain uncertainty for the active party about whether $\mathbf{R} = \mathbf{I}$ or $\mathbf{R} = -\mathbf{I}$ was used in the parameter mapping. Under this scenario, even if the adversary takes the average of the estimations from both cases, the convexity of the MSE would prevent them from achieving an

[19] Recall that for matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we have $\frac{\partial}{\partial \mathbf{A}}\text{Tr}(\mathbf{AB}) = \mathbf{B}^T$ and $\frac{\partial}{\partial \mathbf{A}}\text{Tr}(\mathbf{ABA}^T\mathbf{C}) = \mathbf{CAB} + \mathbf{C}^T\mathbf{AB}^T$

[20] For two square matrices $\mathbf{A}, \mathbf{B}$, we have $\text{Tr}(\mathbf{A}^T\mathbf{B}) \le \sigma_1\|\mathbf{B}\|$, which can be easily verified via Von Neumann's Trace Inequalities.

estimation error lower than the average MSE of each case. Another approach to enhance privacy is to choose a random value for ε and transform the parameters accordingly without disclosing the value of ε to the active party.

5.2.3 Case $\boldsymbol{d = 1}$. In this case, the passive party parameters are a k-dimensional vector, i.e., $\mathbf{w}_{\text{pas}}$. As a result, the matrix $\mathbf{A}_n^T\mathbf{A}_n$ is a scalar. Consider $\mathbf{A}_n = \mathbf{J}\mathbf{w}_n$, where $\mathbf{w}_n$ is the vector the active party is given access to instead of $\mathbf{w}_{\text{pas}}$. To conduct an attack, the adversary forms the equation $\hat{x}_{\text{LS},a}^{\text{PPS}} = (\mathbf{A}_n^T\mathbf{A}_n)^{-1}\mathbf{A}_n^T\mathbf{b}'$ resulting in the following MSE:

$$\text{MSE}(\hat{x}_{\text{LS},a}^{\text{PPS}}) = \mathbb{E}[x^2]\left(1 - \frac{\mathbf{A}_n^T\mathbf{A}}{\mathbf{A}_n^T\mathbf{A}_n}\right)^2. \tag{37}$$

Therefore, the optimization problem of interest here is [21]

$$\begin{aligned} \min_{\mathbf{w}_n \in \mathbb{R}^{k\times 1}} \quad & \frac{\mathbf{A}_n^T\mathbf{A}}{\mathbf{A}_n^T\mathbf{A}_n}\left(2 - \frac{\mathbf{A}_n^T\mathbf{A}}{\mathbf{A}_n^T\mathbf{A}_n}\right) \\ \text{s.t.} \quad & g(\mathbf{w}_n) = \frac{1}{k}(\mathbf{w}_n - \mathbf{w}_{\text{pas}})^T(\mathbf{w}_n - \mathbf{w}_{\text{pas}}) = \varepsilon \end{aligned} \tag{38}$$

Remark 12. For the special case of $d = 1, k = 2$, it can be easily verified that given the vectors $\mathbf{w}_n = [w_{1n}, w_{2n}]$ and $\mathbf{w}_{\text{pas}} = [w_1, w_2]$, we have

$$\text{MSE}(\hat{x}_{\text{LS},a}^{\text{PPS}}) = \mathbb{E}[x^2]\left(\frac{w_{n1} - w_1 + w_2 - w_{2n}}{w_{1n} - w_{2n}}\right)^2. \tag{39}$$

To maximise Equation (39), given the equality constraint $\frac{1}{2}\left((w_1 - w_{n1})^2 + (w_2 - w_{n2})^2\right) = \varepsilon$, define $w_1 - w_{n1} \triangleq r\cos(\theta)$, $w_2 - w_{n2} \triangleq r\sin(\theta)$, and $d_0 \triangleq (w_1 - w_2)/r$. Rewriting the objective and the constraint in terms of r, θ, we have

$$\begin{aligned} \max_{r,\theta} \quad & \left(\frac{\sin(\theta) - \cos(\theta)}{d_0 + \sin(\theta) - \cos(\theta)}\right)^2. \\ \text{s.t.} \quad & r^2 = 2\varepsilon \end{aligned} \tag{40}$$

Given the equality constraint, i.e., $r = \sqrt{2\varepsilon}$, the optimization occurs on a circle with the only variable being θ. By differentiating the objective in Equation (40) with respect to θ, it is verified that the necessary condition for the maximum value is $\theta \in \{\frac{3\pi}{4}, \frac{7\pi}{4}\}$. As a result, the maximizer of Equation (39) can be either $[\mathbf{w}_{\text{pas}} - \sqrt{\varepsilon}, \mathbf{w}_{\text{pas}} + \sqrt{\varepsilon}]$ or $[\mathbf{w}_{\text{pas}} + \sqrt{\varepsilon}, \mathbf{w}_{\text{pas}} - \sqrt{\varepsilon}]$. The designer can try both configurations and select the one which maximizes Equation (39) the most. From Section 5.1.2, observe that interpretability is not compromised if both parameters shift in the same direction. This is also easy to verify from Equation (39), where such a case results in zero MSE. Note that the same approach in this remark can be used to convert the problem in Equation (38) into an optimization over an n-Sphere with a fixed radius (due to the equality constraint), thus eliminating one variable and converting the problem into an unconstrained optimization over the surface of an n-Sphere, which could accelerate the optimization process.

Remark 13. For larger values of ε, there can be possibly an infinite number of solutions for Equation (38) that result in $\mathbf{A}_n^T\mathbf{A}_n = \mathbf{w}_n^T\mathbf{J}^T\mathbf{J}\mathbf{w}_n = 0$. This could occur for vectors with equal components that are within the ε-vicinity of $\mathbf{w}_{\text{pas}}$. Considering the equality constraint significantly affects the demanding task of discarding such solutions. Therefore, one possible direction to speed up the computation in Equation (38) is to replace the equality constraint with an inequality,

[21] Note that $\mathbb{E}[x^2]$ is removed in the optimization as it is a fixed positive parameter.

Remark 14. An interesting observation here is that, in the case of $d = 1$, the PPS does not depend on the statistics of the data. This undermines the idea of preserving the passive party's privacy as the active party can easily replicate the same steps to solve Equation (38). To overcome this situation, the guidelines provided in Remark 11 could be applied. Specifically, the passive party may either remain non-transparent about whether a PPS is in effect, or they could choose a random and secret $0 \leq \varepsilon' \leq \varepsilon$.

Remark 15. In this scenario, any transformation of the parameters similar to cases *i* and *ii* (here we have $\mathbf{R} \in \mathbb{R}$ is a scalar) would result in scaling all the parameters by a constant factor, meaning $\mathbf{w}_n = r\mathbf{w}_{\text{pas}}$. This ensures that the system of equations established by the adversary remains as an overdetermined system with a unique solution. Consequently, no matter which component of $\mathbf{w}_n$ the adversary selects, the attack equation would take the form $w_{n,i}x = rw_{\text{pas,i}}x = b'_i$, where $w_{n,i}$, $w_{\text{pas, i}}$, and b'_i are the ith components of $\mathbf{w}_n$, $\mathbf{w}_{\text{pas}}$, and $\mathbf{b}'$, respectively. It is evident that, in this case, not only a fixed scaling but any linear transformation of the parameters (which results in an overdetermined system with no solution) cannot keep the parameters secret indefinitely. As discussed in [12], the active party can collect valuable statistical information from received confidence scores (if the exact forms are provided) over time, which could be used to estimate the actual values of the parameters.

5.2.4 Case $\boldsymbol{k = 2, d > 1}$. In this case, the passive party's parameters are in the form of a matrix with dimensions $(2 \times d)$. Since $d \geq k$, the adversary is dealing with an underdetermined system, and therefore, we assume half* estimation is employed by the adversary. Noting that the pseudo-inverse of a vector is given by its transpose divided by its squared norm,[22] and assuming $\mathbf{A}_n = \mathbf{J}\mathbf{w}_n$, the resulting MSE is given as [23]

$$\text{MSE}(\hat{\mathbf{X}}^{\text{PPS}}_{\text{half}^*,a}) = \frac{1}{d}\text{Tr}(\mathbf{K}_{\frac{1}{2}\mathbf{1}}) + \frac{1}{d\|\mathbf{A}_n\|^2}(\mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}^T - 2\mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}_n^T + \mathbf{A}_n\mathbf{M}\mathbf{A}_n^T). \tag{41}$$

Equation (41) is independent of any transformed $\mathbf{J}$ used by the adversary. This is because

$$\mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}^T - 2\mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}_n^T + \mathbf{A}_n\mathbf{M}\mathbf{A}_n^T = \text{Tr}(\mathbf{A}^T\mathbf{A}\mathbf{K}_\mathbf{0} - 2\mathbf{A}_n^T\mathbf{A}\mathbf{K}_\mathbf{0} + \mathbf{A}_n^T\mathbf{A}_n\mathbf{M}).$$

To find a right parameter set $\mathbf{w}_n$ to be given to the active party, we solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}_n} \quad & \frac{1}{d\|\mathbf{A}_n\|^2}(2\mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}_n^T - \mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}^T - \mathbf{A}_n\mathbf{M}\mathbf{A}_n^T). \\ \text{s.t.} \quad & g(\mathbf{w}_n) = \varepsilon \end{aligned} \tag{42}$$

Remark 16. The optimization problems in Equations (38) and (42) for $d = 1$ and $k = 2$, respectively, are more general compared to the transforming matrices proposed in cases i and ii. However, since the implementation of the pseudo-inverse of a matrix is technically challenging, it restricts the extension of these cases to cases i and ii.

Remark 17. In Equation (41), two special cases are worth reviewing. First, if $\mathbf{w}_n = \mathbf{w}_{\text{pas}}$, we have

$$\begin{aligned} \text{MSE}(\hat{\mathbf{X}}^{\text{PPS}}_{\text{half}^*,a}) &= \frac{1}{d}\text{Tr}(\mathbf{K}_{\frac{1}{2}\mathbf{1}} - \frac{\mathbf{A}^T\mathbf{A}}{\|\mathbf{A}_n\|^2}(\mathbf{K}_\mathbf{0} - \mathbf{M})) \\ &= \frac{1}{d}\text{Tr}(\mathbf{K}_{\frac{1}{2}\mathbf{1}} - \mathbf{A}^+\mathbf{A}\mathbf{K}_{\frac{1}{2}\mathbf{1}}), \end{aligned} \tag{43}$$

[22] The pseudo-inverse of an all-zero vector is an all-zero vector transposed.

[23] As the steps to obtain Equation (41) are similar to the steps leading to Equation (29), they have been omitted for brevity.

which is the same as the first term in Equation (49) (MSE obtained from applying half* using the exact form of $\mathbf{c}'$). Take any d-dimensional vector, namely $\mathbf{w}'_1$ with non-zero components, and form $\mathbf{w}'_n = [\mathbf{w}'_1, \mathbf{w}'_1]^T$. Set $\varepsilon = g(\mathbf{w}'_n)$. In this case, while the constraint in (42) is satisfied, the objective in (42), $\mathbf{A}'_n = \mathbf{J}\mathbf{w}'_n = \mathbf{0}^T$, is not defined. Additionally, any point in the vicinity of $\mathbf{w}'_n$ holds zero value in terms of interpretability of the VFL model. Noting that for large enough ε, there is an infinite number of such undefined points, one should carefully discard them when solving Equation (42) numerically or enforce the algorithm to avoid such solution structures, which could be a demanding task in some cases. Alternatively, a relaxed version of the optimization problem in Equation (42) could be considered as follows:

$$\begin{aligned} \min_{\mathbf{w}_n} \quad & \frac{1}{d}(2\mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}_n^T - \mathbf{A}\mathbf{K}_\mathbf{0}\mathbf{A}^T - \mathbf{A}_n\mathbf{M}\mathbf{A}_n^T). \\ \text{s.t.} \quad & \|\mathbf{A}_n\|^2 \geq \varepsilon_1 \\ & g(\mathbf{w}_n) = \varepsilon. \end{aligned} \tag{44}$$

Applying the Lagrange multiplier's method, the necessary condition for a stationary point in Equation (44) is obtained as below

$$\frac{1}{d}(2\mathbf{J}^T\mathbf{A}\mathbf{K}_0 - \lambda\mathbf{w}_{\text{pas}}) = \mathbf{J}^T\mathbf{J}\mathbf{w}_n(\mathbf{M} + \mathbf{M}^T) - 2\lambda_1\mathbf{J}^T\mathbf{J}\mathbf{w}_n, \tag{45}$$

where λ and λ_1 are Lagrange multipliers of ε and ε_1, respectively. The solution to Equation (45) can be found [33] to be

$$\text{Vec}(\mathbf{w}_n) = \frac{1}{d}((\mathbf{M} + \mathbf{M}^T - 2\lambda_1\mathbf{I}) \otimes \mathbf{J}^T\mathbf{J})^{-1} \cdot \text{Vec}(2\mathbf{J}^T\mathbf{A}\mathbf{K}_0 - \lambda\mathbf{w}_{\text{pas}}). \tag{46}$$

Given that the inverse in Equation (46) exists, a lookup table could be created according to different values of λ, λ_1 (corresponding to different values of $\varepsilon, \varepsilon_1$) to be used according to the privacy interpretability concerns of the parties.

Remark 18 (Robustness of PPSs and potential attack). In the proposed PPSs, the passive party applies a transformation to its parameters that is determined by a secret design variable R. We explicitly assume that this transformation is *not* disclosed to the active party and that no auxiliary information (e.g., partial passive features obtained via collusion or other side channels) is available. Note that the auxiliary matrix J used in constructing the linear reconstruction systems is fixed by definition. Furthermore, as long as R remains secret, different choices of matrix T yield identical reconstruction performance in terms of MSE, and therefore do not reduce the privacy guaranties enforced by the PPS. Consequently, although the active party may adapt the architecture of its AM or RAM models, such adaptations do not overcome the privacy protection provided by the PPS. The reconstruction error is fundamentally governed by the distortion introduced through the secret transformation R. Note that an active party may continuously collect prediction outputs over time and attempt to jointly estimate both model parameters and passive features by leveraging a large number of observations and imposing feasibility constraints on the feature domain. Intuitively, such an approach would require a substantial number of prediction queries to achieve a meaningful estimation accuracy, with the required number growing with both the model dimension and the number of passive parameters. A systematic analysis of the sample complexity required for joint parameter and feature estimation under these conditions is beyond the scope of this article and is left for future work.

Remark 19 (Adversarial Choice of PPS Parameters). Under the honest-but-curious assumption, the passive party should anticipate that the active party may construct AM or RAM models to assess confidence scores or decide whether to continue collaboration. Accordingly, the passive party's

role is to select the PPS parameters to balance privacy protection and interpretability. In particular, the distortion budget ε governing parameter perturbation is application-dependent and cannot be universally prescribed. Instead, ε may be calibrated by enforcing stability on downstream decisions or explanations (e.g., ensuring that model outputs or rankings remain within acceptable tolerances on a validation set), while maximizing the reconstruction error of passive features. This calibration philosophy parallels the deployment of differential privacy mechanisms, where privacy parameters are typically chosen based on policy, regulatory constraints, and empirical utility considerations rather than closed-form optimality.

Remark 20 (Interpretability–privacy tradeoff). As discussed in Section 5.1.2, revealing the passive-party parameters allows the active party to interpret how its local features interact with the passive features in the learned model. When the PPS introduces distortion, the resulting parameters deviate from their true values, reducing the fidelity of such interpretations. Operationally, higher distortion improves privacy by increasing reconstruction MSE, but decreases interpretability by weakening the correspondence between the released parameters and the true influence of passive features. The PPS therefore enables a tunable tradeoff: low distortion preserves interpretability but yields weaker privacy protection, whereas higher distortion increases privacy at the cost of interpretability. This tradeoff can be adjusted according to the requirements of the deployed VFL application.

6 Experimental Results

In this section, we evaluate the performance of adversary's inference attack on passive features according to the different setups proposed in the previous sections. The performance of inference attacks and PPSs are evaluated according to the MSE per feature in Equation (7), which can be estimated empirically by $\frac{1}{Nd}\sum_{i=1}^{N}\|\mathbf{X}_i-\hat{\mathbf{X}}_i\|^2$ with N denoting the number of samples under study and d denoting the number of passive features. Since agnostic inference attack can be applied on samples both in the training and prediction phase, we set $N=n_t+n_p$ for each dataset, and name this averaging over N as *average over time*. This is to distinguish from another type of averaging, namely, *average over space*, which is explained via an example as follows. Assume that the Bank data, which has 19 features, is considered. Also, consider the case that we are interested in obtaining the MSE when the active and passive parties have 14 and 5 features, respectively. Since these 19 features are not i.i.d., the MSE depends on which 5 (out of 19) features are allocated to the passive party. In order to resolve this issue, we average the MSE over some different possibilities of allocating 5 features to the passive party. More specifically, we average the MSE over a moving window of size 5 features, i.e., MSE is obtained for 19 scenarios where the feature indices of the passive party are $[1:5],[2:6],[3:7],\ldots,\{19\}\cup[1:4]$. Afterwards, these 19 MSE's are summed and divided by 19, which denotes the MSE when $d=5$.

In the following, in Section 6.1, we first review the experimental results related to the agnostic inference attack studied in this article. Then, in Section 6.2, we focus on the performance of the proposed PPSs against this type of attack. Finally, in Section 6.3, we discuss about the computational and system-level overhead introduced by the PPSs.

6.1 Experimental Results on Agnostic Inference Attack

In Figure 4, the resulted MSE from agnostic inference attack via AM (orange solid line denoted by AM) and refined AM (green solid+ line denoted by $\text{RAM}_{n_p,\alpha}$) are compared with half* estimation via VFL model (blue solid• line denoted by VFL) and half estimation (red solid⋆ line) for four different datasets. AM is trained based on the loss function defined in Equation (8) using the training dataset available at the active party, while RAM is trained based on the loss function defined in Equation (9) with the addition of $n_p=100$ confidence scores in their exact form received from CA. The RAM in

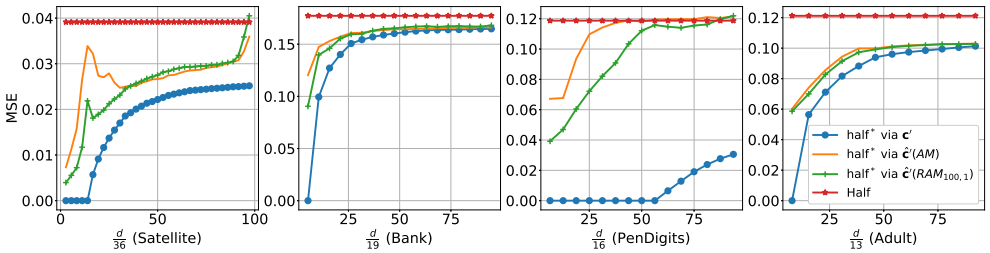


Fig. 4. MSE per feature obtained from agnostic inference attack via $\hat{\mathbf{c}}'$ obtained from AM, $\hat{\mathbf{c}}'$ obtained from refined AM (RAM), half* via the exact form of $\mathbf{c}'$ obtained from the VFL model and half estimation.

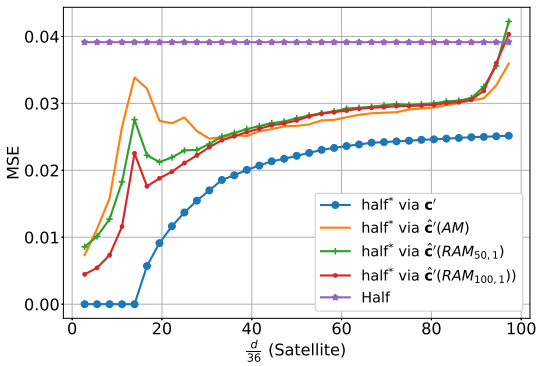


Fig. 5. MSE per feature resulted from agnostic inference attack via $\hat{\mathbf{c}}'$ obtained from AM, $\text{RAM}_{(n_p,\alpha)=(50,1)}$, $\text{RAM}_{(n_p,\alpha)=(100,1)}$, half* applied on $\mathbf{c}'$ and half estimation for the Satellite dataset.

this figure is trained with $\alpha = 1$ as defined in Equation (9). As this is an evaluation for agnostic attacks, the MSE is calculated by considering both the training and prediction samples. It should be noted that since the scores obtained from AM are estimated values of the scores obtained from the VFL model, it is possible that even in the case of $d < k$, the estimated passive features are outside the feasible region (for $d > k$, the system is underdetermined and there is no guarantee that the solution will be in the feasible region). In such cases, the individual features that are outside the feasible region are mapped to either 0 or 1, whichever is closest. For the half estimation, all the passive features are estimated as half. The results show that even without the exact values of the target's confidence scores, it is still possible to achieve a good performance on the inference attack using AM. For the bank, satellite, and adult datasets, the attack performance using AM is close to the half* method via the VFL model in which the adversary has access to the exact values of the target's confidence scores. Moreover, refining AM using Equation (9) with only 100 scores significantly improves the attack performance, especially when the number of passive features is small. This is mainly because the active party has the advantage of a richer dataset. Finally, while the attack using AM is not effective for the pendigit dataset, the attack using RAM significantly improves the performance.

In Figure 5, performance of agnostic inference attack for satellite dataset when the adversary obtains the estimated confidence scores via AM (orange solid), RAM using 50 prediction confidence scores (green solid+) and RAM using 100 prediction confidence scores (red solid·) are compared

with half estimation and half* via the VFL model. It can be seen that incorporating more confidence scores during the refinement process of AM prior to the attack improves the attack performance. The results show that even using as few as 50 confidence scores can still lead to a significant improvement in the attack performance when the number of passive features is low.

Remark 21. The experimental results in this section demonstrate that while the use of confidence scores by the active party can certainly enhance the performance of the inference attack, it is the presence of passive parameters that offers the initial opportunity for exploiting the passive features. It is important to note that tampering with either the confidence scores or the passive parameters can have adverse effects on either the accuracy of the decisions or the interpretability of the results, respectively. In many scenarios, it may be considered more crucial to maintain accuracy in decision-making than the interpretability of the outcomes. Furthermore, as it will be clearer in Section 6.2, for some cases, a small perturbation of the passive parameters can result in a significant improvement in preserving privacy while still allowing for some level of interpretability in the VFL model and keeping decisions accurate (delivering the confidence scores in the clear).

The results shown in Figure 4 reveal that the effectiveness of an agnostic inference attack is highly dependent on the nature of the data. For instance, it can be seen that an attack on the bank and adult datasets results in a higher gain compared to the pendigit dataset. This leads to the question of under what conditions the adversary can achieve a lower MSE through this type of inference attack.

We address this question experimentally. We can consider an ideal scenario in which the features of a dataset are mutually independent. In this scenario, training a classifier on one set of features would not reveal any information about the other set of features, thus rendering the inference attack infeasible. However, in real-world situations, there is always a degree of dependency among the features of a dataset.

Due to lack of technical tools to measure the dependency of two or more random variables directly, instead, here we use covariance of the features of a dataset as an indirect metric. This is represented by the heat map of the covariance matrix $\mathbf{K}_{\boldsymbol{\mu}}$ shown in Figure 6. This matrix is calculated as $\mathbf{K}_{\boldsymbol{\mu}} \triangleq \mathbb{E}[(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^T]$, where $\mathbf{X}$ is a d_t-dimensional vector that represents the features (both passive and active features) and $\boldsymbol{\mu}$ is the mean vector.

It is observed that for the grid dataset, the corresponding covariance matrix is nearly diagonal, indicating that the features are nearly uncorrelated. As shown in Figure 7, the agnostic inference attack on this dataset is ineffective as it is almost similar to estimating the features as half.

Remark 22. Figure 8 presents the average model accuracy of AM versus the number of active features in the datasets listed in Table 1. The results show that the active party benefits more from VFL in the grid and pendigits datasets compared to the bank, satellite, and adult datasets. This is because as the number of active features increases, the active party can achieve higher accuracy on the grid and pendigit datasets. The results from Figure 8 in conjunction with the results from Figures 4, 6, and 7 suggest that datasets that are more suitable for VFL collaboration and result in higher accuracy from VFL collaboration are less susceptible to agnostic inference attacks. This is a crucial aspect to consider for the active party when deciding to participate in a VFL collaboration, especially if the active party incurs costs for both VFL training and prediction.

6.2 Experimental Results on Privacy-Preserving Schemes

In this section, we evaluate the effectiveness of the PPSs introduced in Section 5. Our assessment consists of two stages: first, we analyze each of the cases studied in Section 5.2 separately by keeping

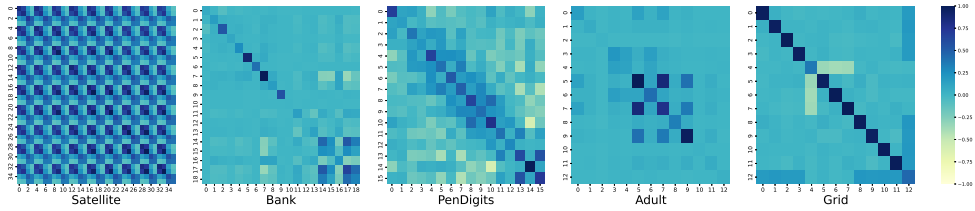


Fig. 6. Heat map of covariance matrix for the datasets in Table 1.

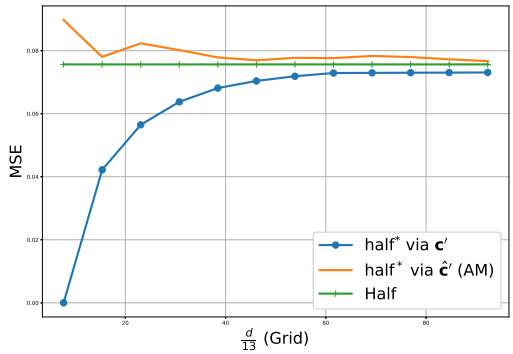


Fig. 7. MSE per feature resulted from agnostic inference attack via $\hat{\mathbf{c}}'$ obtained from AM, half* applied on $\mathbf{c}'$ and half estimation.

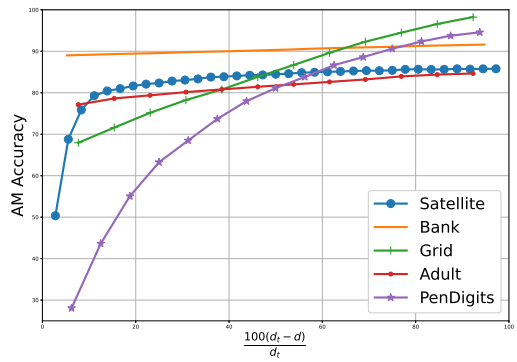


Fig. 8. Accuracy of AM versus active features for the datasets in Table 1.

either d and/or k constant and varying the value of ε. Next, we examine the opposite scenario by fixing ε and varying d and k.

To measure the interpretability of the passive parameters at the active party, we use the function $g(\cdot)$, where a higher value of $g(\cdot)$ indicates lower interpretability. On the other hand, the privacy of the passive party is measured by the amount of MSE incurred by the active party in the case of an attack, with higher MSE values indicating greater privacy for the passive party.

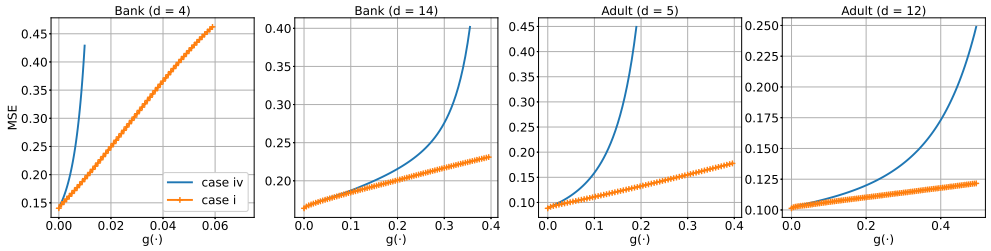


Fig. 9. Illustration of PI tradeoff for case i and case iv for the datasets bank and adult.

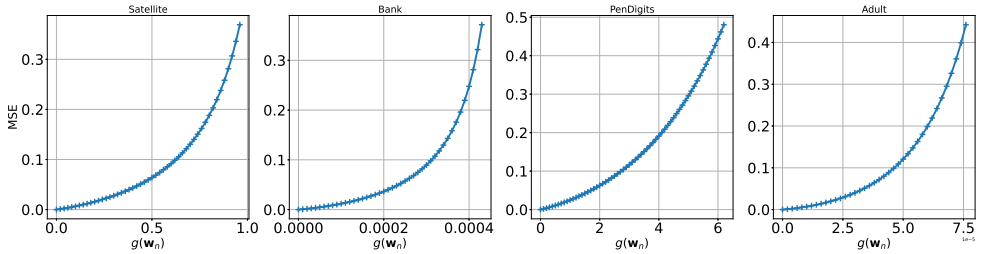


Fig. 10. Illustration of PI tradeoff for case iii for the datasets bank, adult, satellite, and pendigits.

In Figure 9, the PI tradeoffs for the datasets bank and adult are illustrated for three different values of passive features. The orange-solid+ and blue-solid• lines are the PI tradeoffs obtained via the approaches in case i and case iv, respectively. It is worth noting two key aspects regarding the results. *First*, the inclusion of the results of case i is merely intended to provide a comparison with the more general PPS approach in case iv, and they are not applicable for datasets with $k = 2$. As seen from the plots, while the results of case i deviate quickly from those of case iv, they remain close to each other in the region of interest, which is approximately 0.18 and 0.14 for the bank and adult datasets, respectively, as indicated in Figure 4 (see the MSE obtained from half estimation). Furthermore, as the value of d increases, the results from case i and case iv converge, as shown in the rightmost plots in Figure 9. *Second*, both the passive and active parties prefer that a small amount of perturbation in the passive parameters results in a noticeable increase in the MSE from inference attacks, while still preserving a reasonable level of interpretability. This way, both parties meet their concerns with a minimal passive parameter perturbation. This is equivalent to having a concave PI tradeoff curve. However, as observed from Figure 9 and the results throughout this section, the PI tradeoff achieved by the proposed PPSs for different datasets and under different cases can sometimes be convex, sometimes concave, and in some cases, neither. It remains an open question whether the optimal PI tradeoff curve in the region of interest follows a certain shape, preferably concave.

In Figure 10, the PI tradeoff is demonstrated for the datasets bank, adult, satellite, and pendigits, when $d = 1$. To obtain the PI curve for the bank and adult datasets, we utilized the result from Remark (12), while for the remaining datasets, the optimization problem in (38) was solved. The approach in this case is considered more comprehensive as it does not impose any limitations on the mapping between the passive parameters and the parameters that the active party has given access to. Note that, as shown in Figure 4, the MSE for the half estimation on the satellite, bank, pendigits, and adult datasets is approximately 0.04, 0.17, 0.12, and 0.12, respectively. This leads to

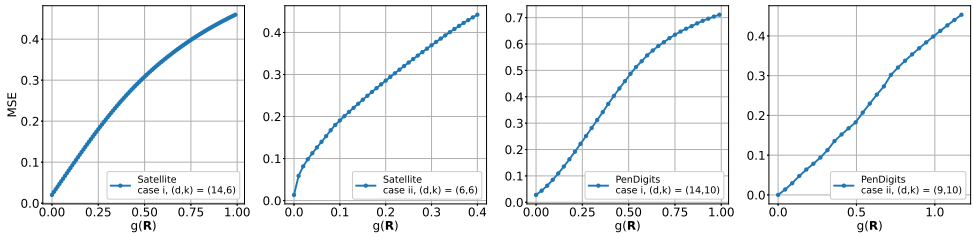


Fig. 11. Illustration of PI tradeoff for case i for the datasets satellite and pendigits.

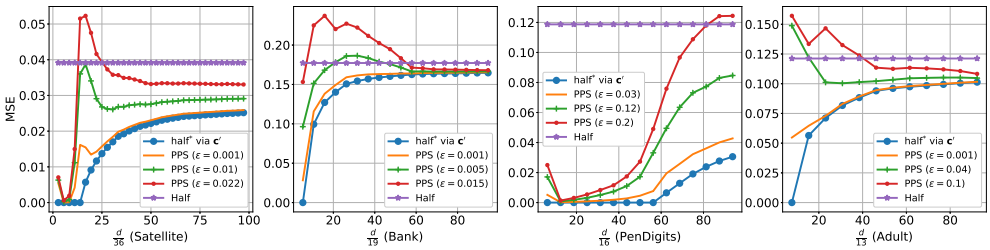


Fig. 12. MSE resulted from inference attack when PPSs (with different interpretability levels) proposed in Section 5.2 are in place, compared with half* and half estimation.

the observation that the PI curve below this MSE level for each of these datasets is almost linear. This observation is promising in the sense that the PPS is capable of delivering a balance between privacy and interpretability concerns of the parties, despite the general non-concave PI tradeoff curve.

In Figure 11, the PI tradeoff for case i and ii is depicted for the satellite and pendigits datasets. It's important to keep in mind that in this instance, there is no widely accepted general solution that can serve as a benchmark for comparison. As a result, it remains a potential area of research to identify mappings from $\mathbf{W}_{\text{pas}}$ to $\mathbf{W}_n$ that result in an improvement in the PI tradeoff region.

In Figure 12, the effectiveness of the PPSs outlined in Section 5.2 is compared to the baseline methods half* and half estimation. Each plot in the figure corresponds to one of the datasets, with three different levels of interpretability depicted in each plot. For instance, in the top-left plot related to the satellite dataset, the interpretability levels are $\varepsilon = 0.001, 0.01, 0.022$. Unlike Figures 9, 10, and 11, in this figure the performance of all four cases for PPS are evaluated against varying numbers of passive features with a fixed interpretability level in each curve. The results indicate that, for a fixed ε, the gap between the MSE and that of half* is not consistent. This is because as the number of passive parameters increases, finding the optimal solution for the system $\mathbf{A}_n\mathbf{X} = \mathbf{b}'$ becomes more challenging. For the satellite and pendigit datasets, the MSE does not seem to be affected as the value of ε varies for the case of $d = 2$. However, as seen in the results in Figure 13 (MSE per feature of inference attack versus different values of ε for $d = 2$), the amount of ε required to produce a noticeable change in the MSE is not comparable to the other cases.

Remark 23 (On baseline defenses and experimental scope). Throughout this article, we adopt a worst-case utility assumption in which the active party receives confidence scores in their exact form. This choice is motivated by applications that rely on soft scores for downstream tasks and allows us to study privacy risks that cannot be mitigated by score-level obfuscation.

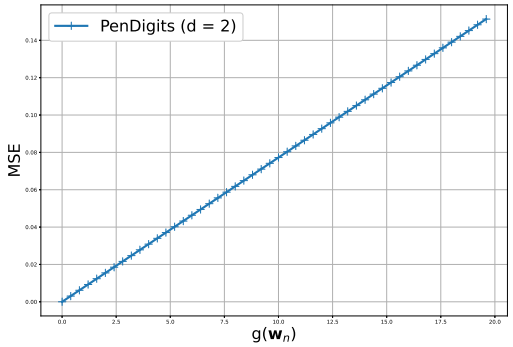


Fig. 13. MSE per feature of inference attack versus different values of ε for $d = 2$ (Pendigits).

Consequently, defenses based on rounding, clipping, or injecting noise into confidence scores are intentionally excluded from the experimental evaluation. While adding random noise to model parameters may appear as a simple alternative, such perturbations lack a principled design criterion unless calibrated through an explicit objective. In contrast, the proposed PPSs introduce structured parameter distortions obtained via optimization, enabling a controlled and interpretable privacy–interpretability tradeoff. Black-box configurations, where passive parameters are entirely hidden, trivially eliminate reconstruction attacks but also remove interpretability at the active party; they therefore serve as conceptual reference points rather than experimental baselines.

Remark 24 (On Gaussian noise baselines). A natural baseline is to perturb the passive-party parameters with i.i.d. Gaussian noise under the same distortion budget ε. However, in the present setting, such a comparison is not particularly informative. First, the passive parameter block is low-dimensional, and therefore the induced privacy metric (reconstruction MSE) is highly sensitive to individual noise realizations. Unlike our optimization-based PPS, which is deterministic once ε is fixed, a naive Gaussian mechanism yields unstable privacy–interpretability outcomes unless one averages over many independent perturbation draws. Second, our evaluation already involves averaging over samples and over different passive-feature allocations. A fair Gaussian comparison would therefore require an additional Monte Carlo averaging layer, substantially increasing variance reduction requirements and computational cost. Moreover, the Gaussian variance cannot be directly matched to the same distortion budget ε in closed form; instead, it must be calibrated empirically through iterative procedures (e.g., bisection), where each candidate variance requires repeated evaluations of the full pipeline. Finally, Gaussian perturbation is geometry-agnostic and does not exploit the structure of the reconstruction system induced by the passive parameters. As a result, in this low-dimensional regime, its outcomes are weakly diagnostic: observed differences are often dominated by stochastic variability rather than reflecting meaningful privacy–utility tradeoffs. For these reasons, we do not include this baseline and instead focus on the proposed PPS, which directly optimizes distortion under the reconstruction model.

6.3 Overhead Analysis and Practical Considerations

In the following, we discuss the system-level overhead introduced by the proposed PPSs, with particular emphasis on their practical implementation and deployment cost.

Computational overhead: All PPSs proposed in this article are applied *offline*, after completion of VFL model training and prior to releasing the passive party's parameters to the active party. For PPS designs formulated as optimization problems—including those with orthogonality constraints

on Stiefel manifolds—the optimization is carried out locally at the passive party and involves only the learned model parameters and precomputed second-order statistics.

In our implementation, the PPS optimization is implemented in Python using standard numerical libraries (NumPy and SciPy), together with manifold-optimization tools for orthogonality-constrained problems. In particular, Stiefel-manifold-based designs are solved using existing manifold solvers (e.g., smooth projected or Riemannian gradient-based methods), which operate directly on matrix variables and exploit closed-form expressions for the objective and its gradient. Because the optimization variables are of dimension proportional to the number of passive features and classes, and because no data-dependent iterations are involved, the resulting wall-clock runtime is on the order of milliseconds for all datasets considered.

Communication overhead: The proposed PPSs do not introduce additional communication rounds during VFL training or prediction. The only additional step is the one-time release of the modified passive parameters in place of the original ones. Therefore, from a system perspective, the communication pattern of the underlying VFL protocol remains unchanged.

6.4 Guidelines for VFL System Designers

At a high level, agnostic attacks are most concerning in deployments where (i) the active party has access to labels, (ii) features across parties exhibit moderate or strong correlation, and (iii) the application requires soft confidence scores for downstream tasks. In such settings, the active party can derive surrogate models (AM/RAM) that enable reconstructions comparable in accuracy to those obtained under non-agnostic attacks.

The proposed PPSs are intended for scenarios where passive-party interpretability is permitted or required, but where exact parameter disclosure poses privacy risks. They are applied once after model training and adjust only the passive-party parameters, allowing the VFL workflow and prediction interface to remain unchanged. For practitioners, the key operational guideline is to select the distortion level that achieves the desired privacy target (in terms of reconstruction MSE) while retaining sufficient interpretability for audit or compliance purposes.

7 Conclusion

In this article, we examine the intricacies of a white-box VFL setting where two parties are involved in constructing a classifier collaboratively. The active party has access to the labels while the passive party contributes by sharing a separate set of features for the same samples. This collaboration takes place under the supervision of a third trusted entity known as the CA. Previous studies, such as those in [11, 12], have explored various inference attack methods that mainly rely on the active party's exploitation of the confidence scores during the prediction phase. However, in this study, we investigate a novel approach of inference attack, which does not require the active party to have access to the target's confidence score. The adversary uses their available training set, including labels and active party features, to construct a classifier that is then used to estimate the target's confidence score. Once the estimation is obtained, the active party uses one of the previously studied inference attack methods to carry out the attack. This attack is referred to as "agnostic" due to the absence of the target's confidence score at the active party, which as a result puts all the samples used in the training phase at the risk of privacy exposure. Our findings show that the performance of the adversary's model can be significantly enhanced by incorporating the confidence scores received from CA. This results in more accurate estimates of the target's confidence score, which, in turn, can lead to more successful inference attacks. Since this type of attack does not rely on the availability of an attack target's confidence score, it is therefore unaffected of any PPSs applied to the confidence scores, such as rounding or adding noise.

To counter the aforementioned potential for privacy breaches in a VFL setting, various PPSs have been proposed with a focus on protecting the passive parameters. To that end, we consider the development of PPSs that intentionally distort the passive parameters in a systematic manner. The active party is given access to these distorted parameters instead of the original versions. In designing these PPSs, we carefully consider the need for interpretability by the active party while still safeguarding the privacy of the passive party. This, in turn, gives rise to a tradeoff between the interpretability of the VFL model for the active party and the privacy concerns of the passive party. Our approach moves away from a black or white viewpoint in VFL and instead aims to find a suitable balance between the needs of both parties to mitigate the risk of collaboration. Our goal is not to completely obscure the active party or completely protect the passive party, but rather to strike a balance between the two in order to minimize potential privacy risks.

References

[1] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469. DOI: http://dx.doi.org/10.1109/TIFS.2020.2988575

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*. Aarti Singh and Jerry Zhu (Eds.), Vol. 54, PMLR, 1273–1282. Retrieved from https://proceedings.mlr.press/v54/mcmahan17a.html

[3] Songtao Lu, Yawen Zhang, and Yunlong Wang. 2020. Decentralized federated learning for electronic health records. In *Proceedings of the 2020 54th Annual Conference on Information Sciences and Systems (CISS)*. 1–5. DOI: http://dx.doi.org/10.1109/CISS48834.2020.1570617414

[4] Wenqi Li, Fausto Milletari, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M. Jorge Cardoso, et al. 2019. Privacy-preserving federated brain tumour segmentation. arXiv:1910.00962. Retrieved from http://arxiv.org/abs/1910.00962

[5] Francoise Beaufays, Kanishka Rao, Rajiv Mathews, and Swaroop Ramaswamy. 2019. Federated learning for emoji prediction in a mobile keyboard. arXiv:1906.04329. Retrieved from https://arxiv.org/abs/1906.04329

[6] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. arXiv:1811.03604. Retrieved from http://arxiv.org/abs/1811.03604

[7] Kai Yang, Tao Fan, Tianjian Chen, Yuanming Shi, and Qiang Yang. 2019. A quasi-Newton method based vertical federated learning framework for logistic regression. arXiv:1912.00513. Retrieved from http://arxiv.org/abs/1912.00513

[8] Yichuan Wang, Yuying Tian, Xinyue Yin, and Xinhong Hei. 2020. A trusted recommendation scheme for privacy protection based on federated learning. *CCF Transactions on Networking* 3, 4 (12 2020), 218–228. DOI: http://dx.doi.org/10.1007/s42045-020-00045-8

[9] Rong Yu and Peichun Li. 2021. Toward resource-efficient federated learning in mobile edge computing. *IEEE Network* 35, 1 (2021), 148–155. DOI: http://dx.doi.org/10.1109/MNET.011.2000295

[10] Yong Cheng, Yang Liu, Tianjian Chen, and Qiang Yang. 2020. Federated learning for privacy-preserving AI. *Communications of the ACM* 63, 12 (Nov. 2020), 33–36. DOI: http://dx.doi.org/10.1145/3387107

[11] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. 2020. Feature inference attack on model predictions in vertical federated learning. arXiv:2010.10152. Retrieved from https://arxiv.org/abs/2010.10152

[12] Borzoo Rassouli, Morteza Varasteh, and Deniz Gunduz. 2022. Privacy against inference attacks in vertical federated learning. arXiv:2207.11788. Retrieved from https://arxiv.org/abs/2207.11788 DOI: http://dx.doi.org/10.48550/ARXIV.2207.11788

[13] Andrew C. Yao. 1982. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*. 160–164. DOI: http://dx.doi.org/10.1109/SFCS.1982.38

[14] Ivan Damgård and Mads Jurik. 2001. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Public Key Cryptography*. Kwangjo Kim (Ed.), Springer Berlin Heidelberg, Berlin, 119–136.

[15] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. (2017). Retrieved April 2026 from http://archive.ics.uci.edu/ml

[16] Andrew W. Lo and A. Craig MacKinlay. 2015. Data-snooping biases in tests of financial asset pricing models. *The Review of Financial Studies* 3, 3 (04 2015), 431–467. DOI: http://dx.doi.org/10.1093/rfs/3.3.431

[17] github. 2020. GitHub. (2020). Retrieved April 2026 from https://github.com/mrtzvrst

[18] Dongning Guo, S. Shamai, and S. Verdu. 2005. Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory* 51, 4 (2005), 1261–1282. DOI: http://dx.doi.org/10.1109/TIT.2005.844072

[19] Paul W. Cuff and Lanqing Yu. 2016. Differential privacy as a mutual information constraint. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* . Retrieved from https://api.semanticscholar.org/CorpusID:9204999

[20] Xue Jiang, Xuebing Zhou, and Jens Grossklags. 2022. Comprehensive analysis of privacy leakage in vertical federated learning during prediction. *Proceedings on Privacy Enhancing Technologies* 2022, 2 (04 2022), 263–281. DOI: http://dx.doi.org/10.2478/popets-2022-0045

[21] R. Penrose. 1955. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society* 51, 3 (1955), 406–413. DOI: http://dx.doi.org/10.1017/S0305004100030401

[22] Michael Bücker, Gero Szepannek, Alicja Gosiewska, and Przemyslaw Biecek. 2020. Transparency, auditability and eXplainability of machine learning models in credit scoring. arXiv:stat.ML/2009.13384. Retrieved from https://arxiv.org/abs/2009.13384

[23] Oguz Koc, Omur Ugur, and A. Sevtap Kestel. 2023. The impact of feature selection and transformation on machine learning methods in determining the credit scoring. arXiv:q-fin.RM/2303.05427. Retrieved from https://arxiv.org/abs/2303.05427

[24] Siqi Li, Yilin Ning, Marcus Eng Hock Ong, Bibhas Chakraborty, Chuan Hong, Feng Xie, Han Yuan, Mingxuan Liu, Daniel M. Buckland, Yong Chen, and Nan Liu. 2023. FedScore: A privacy-preserving framework for federated scoring system development. *Journal of Biomedical Informatics* 146 (Oct. 2023), 104485. DOI: http://dx.doi.org/10.1016/j.jbi.2023.104485

[25] Matthew E. Shipe, Stephen A. Deppen, Farhood Farjah, and Eric L. Grogan. 2019. Developing prediction models for clinical use using logistic regression: An overview. *Journal of Thoracic Disease* 11, Suppl 4 (Mar. 2019), S574–S584. DOI: http://dx.doi.org/10.21037/jtd.2019.01.25

[26] Y. Wu, K. Wang, Z. Zhang, W. Lin, H. Chen, and C. Li. 2018. Privacy-preserving group nearest neighbor search. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*. 277–288.

[27] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A system for scalable privacy-preserving machine learning. In *Proceedings of the IEEE Symposium on Security and Privacy*. 19–38. DOI: http://dx.doi.org/10.1109/SP.2017.12

[28] A. R. Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha. 2020. Crypt?: Crypto-assisted differential privacy on untrusted servers. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. 603–619. DOI: http://dx.doi.org/10.1145/3318464.3389769

[29] Alan Edelman, Tomás A. Arias, and Steven T. Smith. 1998. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications* 20, 2 (1998), 303–353. DOI: http://dx.doi.org/10.1137/S0895479895290954

[30] Nachuan Xiao, Xin Liu, and Ya-xiang Yuan. 2021. A penalty-free infeasible approach for a class of nonsmooth optimization problems over the Stiefel manifold. arXiv:2103.03514. Retrieved from https://arxiv.org/abs/2103.03514 DOI: http://dx.doi.org/10.48550/ARXIV.2103.03514

[31] Nachuan Xiao, Xin Liu, and Ya xiang Yuan. 2022. A class of smooth exact penalty function methods for optimization problems with orthogonality constraints. *Optimization Methods and Software* 37, 4 (2022), 1205–1241. DOI: http://dx.doi.org/10.1080/10556788.2020.1852236

[32] T. N. E. Greville. 1966. Note on the generalized inverse of a matrix product. *SIAM Review* 8, 4 (1966), 518–521. DOI: http://dx.doi.org/10.1137/1008107

[33] K. B. Petersen and M. S. Pedersen. 2008. The Matrix Cookbook. (Oct. 2008). Retrieved April 2026 from http://www2.imm.dtu.dk/pubdb/p.php?3274

Appendices

A Derivation of (13)

$$\begin{aligned}
\mathrm{MSE}(\hat{\mathbf{X}}_{\mathrm{LS},a}) &= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\hat{\mathbf{X}}_{\mathrm{LS},a}\|^2\right] \\
&= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{b}'+\hat{\mathbf{c}}'-\mathbf{c}')\|^2\right] \\
&= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{X}+\hat{\mathbf{c}}'-\mathbf{c}')\|^2\right] \\
&= \frac{1}{d}\mathbb{E}\left[\|(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\hat{\mathbf{c}}'-\mathbf{c}')\|^2\right] \\
&= \frac{1}{d}\mathbb{E}\left[\mathrm{Tr}(\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\hat{\mathbf{c}}'-\mathbf{c}')(\hat{\mathbf{c}}'-\mathbf{c}')^T\right] \\
&= \frac{1}{d}\mathrm{Tr}\left(\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-2}\mathbf{A}^T\mathbf{K}_{\mathbf{c}',\hat{\mathbf{c}}'}\right), \qquad (47)
\end{aligned}$$

B Derivation of (18)

$$\begin{aligned}
\mathrm{MSE}(\hat{\mathbf{X}}_{\mathrm{half}^*,a}) &= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\hat{\mathbf{X}}_{\mathrm{half}^*,a}\|^2\right] \\
&= \frac{1}{d}\mathbb{E}\left[\|(\mathbf{X}-\hat{\mathbf{X}}_{\mathrm{half}^*}-\mathbf{A}^+(\hat{\mathbf{c}}'-\mathbf{c}'))\|^2\right] \\
&= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\hat{\mathbf{X}}_{\mathrm{half}^*}\|^2\right]+\frac{1}{d}\mathbb{E}\left[\|\mathbf{A}^+(\hat{\mathbf{c}}'-\mathbf{c}')\|^2\right]-\frac{2}{d}\mathbb{E}\left[(\hat{\mathbf{c}}'-\mathbf{c}')^T\mathbf{A}^{+T}(\mathbf{I}-\mathbf{A}^+\mathbf{A})(\mathbf{X}-\frac{1}{2}\mathbf{1}_d)\right] \qquad (48) \\
&= \frac{1}{d}\mathrm{Tr}\left((\mathbf{I}-\mathbf{A}^+\mathbf{A})\,\mathbf{K}_{\frac{1}{2}\mathbf{1}}\right)+\frac{1}{d}\mathbb{E}\left[\|\mathbf{A}^+(\hat{\mathbf{c}}'-\mathbf{c}')\|^2\right] \qquad (49) \\
&= \frac{1}{d}\mathrm{Tr}\left((\mathbf{I}-\mathbf{A}^+\mathbf{A})\,\mathbf{K}_{\frac{1}{2}\mathbf{1}}\right)+\frac{1}{d}\mathrm{Tr}\left(\mathbf{A}^{+T}\mathbf{A}^+\mathbf{K}_{\mathbf{c}',\hat{\mathbf{c}}'}\right), \qquad (50)
\end{aligned}$$

where the third term in Equation (48) is 0 due to $\mathbf{A}^{+T}(\mathbf{I}-\mathbf{A}^+\mathbf{A})=\mathbf{0}$. The first term in Equation (49) is due to [12, Theorem 2]. It is important to note that the second term in Equation (48) is always positive, resulting in a positive value for the second term in Equation (18).

C Derivation of (21)

$$\begin{aligned}
\mathrm{MSE}(\hat{\mathbf{X}}_{\mathrm{LS}}^{\mathrm{PPS}}) &= \mathbb{E}[\|\mathbf{X}-\hat{\mathbf{X}}_{\mathrm{LS}}^{\mathrm{PPS}}\|^2] \\
&= \mathbb{E}[\|\mathbf{X}-\mathbf{A}_n^+\mathbf{b}'\|^2] \\
&= \mathbb{E}[\|(\mathbf{I}-\mathbf{A}_n^+\mathbf{A})\mathbf{X}\|^2] \qquad (51) \\
&= \mathbb{E}[\|(\mathbf{I}-\mathbf{R}^T\mathbf{A}^+\mathbf{A})\mathbf{X}\|^2] \qquad (52) \\
&= \mathrm{Tr}((\mathbf{I}-\mathbf{R}^T\mathbf{A}^+\mathbf{A})\mathbf{K}_{\mathbf{0}}(\mathbf{I}-\mathbf{A}^+\mathbf{A}\mathbf{R})) \qquad (53) \\
&= \mathrm{Tr}((\mathbf{I}+\mathbf{A}^+\mathbf{A}-\mathbf{A}^+\mathbf{A}\mathbf{R}-\mathbf{R}^T\mathbf{A}^+\mathbf{A})\mathbf{K}_{\mathbf{0}}) \qquad (54) \\
&= \mathrm{Tr}(\mathbf{I}+\mathbf{A}^+\mathbf{A})-2\mathrm{Tr}(\mathbf{R}\mathbf{K}_{\mathbf{0}}\mathbf{A}^+\mathbf{A}), \qquad (55)
\end{aligned}$$

where in Equations (51)–(55) we have $\mathbf{b}'=\mathbf{A}\mathbf{X}$, $(\mathbf{A}\mathbf{R})^+=\mathbf{R}^{-1}\mathbf{A}^+=\mathbf{R}^T\mathbf{A}^+$, because of orthonormality of $\mathbf{R}$, $(\mathbf{A}^+\mathbf{A})^2=\mathbf{A}^+\mathbf{A}$, $\mathbf{K}_{\boldsymbol{\mu}}\triangleq\mathbb{E}[(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^T]$ with $\boldsymbol{\mu}=\mathbf{0}$, $\mathrm{Tr}(\mathbf{R}^T\mathbf{A}^+\mathbf{A}\mathbf{K}_{\mathbf{0}})=\mathrm{Tr}(\mathbf{A}^+\mathbf{A}\mathbf{R}\mathbf{K}_{\mathbf{0}})$, respectively.

D Derivation of (29)

$$
\begin{aligned}
\mathrm{MSE}(\hat{\mathbf{X}}^{\mathrm{PPS}}_{\mathrm{half}^*,a}) &= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\hat{\mathbf{X}}^{\mathrm{PPS}}_{\mathrm{half}^*,a}\|^2\right]\\
&= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\mathbf{R}^T\mathbf{A}^+\mathbf{A}\mathbf{X}-\frac{1}{2}(\mathbf{I}-\mathbf{R}^T\mathbf{A}^+\mathbf{A}\mathbf{R})\mathbf{1}_d\|^2\right]\\
&= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\frac{1}{2}\mathbf{1}_d-\mathbf{R}^T\mathbf{A}^+\mathbf{A}(\mathbf{X}-\frac{1}{2}\mathbf{R}\mathbf{1}_d)\|^2\right]\\
&= \frac{1}{d}\mathbb{E}\left[\|\mathbf{X}-\frac{1}{2}\mathbf{1}_d\|^2+\mathrm{Tr}(\mathbf{A}^+\mathbf{A}(\mathbf{X}-\frac{1}{2}\mathbf{R}\mathbf{1}_d)(\mathbf{X}-\frac{1}{2}\mathbf{R}\mathbf{1}_d)^T)-2\mathrm{Tr}(\mathbf{R}^T\mathbf{A}^+\mathbf{A}(\mathbf{X}-\frac{1}{2}\mathbf{R}\mathbf{1}_d)(\mathbf{X}-\frac{1}{2}\mathbf{1}_d)^T)\right]\\
&= \frac{1}{d}\mathrm{Tr}(\mathbf{K}_{\frac{1}{2}\mathbf{1}})+\frac{1}{d}\mathbb{E}\left[\mathrm{Tr}(\mathbf{A}^+\mathbf{A}(\mathbf{X}-\frac{1}{2}\mathbf{R}\mathbf{1}_d)\cdot(\mathbf{X}+\frac{1}{2}\mathbf{R}\mathbf{1}_d-2\mathbf{R}\mathbf{X})^T)\right] \qquad (56)\\
&= \frac{1}{d}\mathrm{Tr}(\mathbf{K}_{\frac{1}{2}\mathbf{1}})+\frac{1}{d}\mathbb{E}\left[\mathrm{Tr}(\mathbf{A}^+\mathbf{A}(\mathbf{X}\mathbf{X}^T-2\mathbf{X}\mathbf{X}^T\mathbf{R}^T))\right]-\frac{1}{d}\mathbb{E}\left[\mathrm{Tr}(\mathbf{A}^+\mathbf{A}(\frac{1}{4}\mathbf{R}\mathbf{1}_d\mathbf{1}_d^T\mathbf{R}^T-\mathbf{R}\mathbf{1}_d\mathbf{X}^T\mathbf{R}))\right] \qquad (57)\\
&= \frac{1}{d}\mathrm{Tr}\left(\mathbf{K}_{\frac{1}{2}\mathbf{1}}+\mathbf{A}^+\mathbf{A}\left(\mathbf{K}_\mathbf{0}-2\mathbf{K}_\mathbf{0}\mathbf{R}^T\frac{1}{4}\mathbf{R}\mathbf{1}_d\mathbf{1}_d^T\mathbf{R}^T+\mathbf{R}\mathbf{1}_d\boldsymbol{\mu}^T\mathbf{R}\right)\right)\\
&= \frac{1}{d}\mathrm{Tr}(\mathbf{K}_{\frac{1}{2}\mathbf{1}})-\frac{1}{d}\mathrm{Tr}\left(\mathbf{A}^+\mathbf{A}\left(2\mathbf{K}_\mathbf{0}\mathbf{R}^T-\mathbf{K}_\mathbf{0}-\mathbf{R}\mathbf{M}\mathbf{R}^T\right)\right), \qquad (58)
\end{aligned}
$$

where in Equation (56) we use $\mathbf{R}^T\mathbf{R}=\mathbf{I}$, and $(\mathbf{A}^+\mathbf{A})^T\mathbf{A}^+\mathbf{A}=(\mathbf{A}^+\mathbf{A})^2=\mathbf{A}^+\mathbf{A}$. In Equation (57). we use $\mathrm{Tr}(\mathbf{A}^+\mathbf{A}\mathbf{X}\mathbf{1}_d^T\mathbf{R}^T)=\mathrm{Tr}(\mathbf{R}\mathbf{1}_d\mathbf{X}^T\mathbf{A}^+\mathbf{A})=\mathrm{Tr}(\mathbf{A}^+\mathbf{A}\mathbf{R}\mathbf{1}_d\mathbf{X}^T)$ and in Equation (58), we have $\mathbf{M}\triangleq\mathbf{1}_d(\boldsymbol{\mu}-\frac{1}{4}\mathbf{1}_d)^T$.

E Derivation of (35)

$$
\begin{aligned}
\mathrm{MSE}(\hat{\mathbf{X}}^{\mathrm{PPS}}_{\mathrm{LS},a}) &= \frac{1}{d}\mathbb{E}\left[\left\|\mathbf{X}-(\mathbf{A}_n^T\mathbf{A}_n)^{-1}\mathbf{A}_n^T\mathbf{b}'\right\|^2\right]\\
&= \frac{1}{d}\mathbb{E}\left[\left\|\mathbf{X}-(\mathbf{A}_n^T\mathbf{A}_n)^{-1}\mathbf{A}_n^T\mathbf{A}\mathbf{X}\right\|^2\right]\\
&= \frac{1}{d}\mathbb{E}\left[\left\|\mathbf{X}-(\mathbf{R}^T\mathbf{A}^T\mathbf{A}\mathbf{R})^{-1}\mathbf{R}^T\mathbf{A}^T\mathbf{A}\mathbf{X}\right\|^2\right]\\
&= \frac{1}{d}\mathbb{E}\left[\left\|(\mathbf{I}-\mathbf{R}^T)\mathbf{X}\right\|^2\right]\\
&= \frac{2}{d}\mathrm{Tr}\left(\mathbf{K}_\mathbf{0}-\mathbf{R}\mathbf{K}_\mathbf{0}\right). \qquad (59)
\end{aligned}
$$

In Equation (59), we use the identity $\mathrm{Tr}(\mathbf{R}^T\mathbf{K}_0)=\mathrm{Tr}(\mathbf{R}\mathbf{K}_0)$.

Received 29 August 2025; revised 27 January 2026; accepted 14 March 2026