Equilibrium-Driven Vertical Federated Learning with Selective Privacy Protection

Yuanzhe Peng[1], Wenwei Zhao[2], Zhuo Lu[2], Jie Xu[1]

[1]University of Florida, Gainesville, FL, USA
[2]University of South Florida, Tampa, FL, USA
{pengy1, jie.xu}@ufl.edu, {wenweizhao, zhuolu}@usf.edu

Abstract

Vertical Federated Learning (VFL) enables multiple clients with feature-partitioned data to collaboratively train models while preserving privacy by transmitting embeddings instead of raw data. However, such embeddings can still expose sensitive attributes (e.g., gender or race) unrelated to the target task, making them vulnerable to attribute inference attacks. Most existing privacy strategies may provide extra protection, but at the cost of reduced accuracy and excessive privacy budget. In this paper, we propose a novel equilibrium-driven VFL framework with selective privacy protection for sensitive attributes that are difficult to isolate from embeddings, thereby enhancing local privacy with minor accuracy compromise. We introduce two key innovations: (1) a *NashCoder*, which incorporates a surrogate head to jointly optimize accuracy and privacy; (2) an adaptive decomposition strategy based on Shapley values, which dynamically decomposes the global objective for distributed optimization from an equilibrium perspective. We theoretically analyze our framework and empirically evaluate it on three public datasets against five baselines, demonstrating significant improvements in the accuracy-privacy trade-off under various privacy settings. Extensive experimental results support our theoretical analysis.

1 Introduction

Federated Learning (FL) is a distributed learning paradigm that enables multiple clients to collaboratively train models while preserving data privacy. FL can be categorized into two main types: Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL). In HFL, data is horizontally partitioned across different clients, with each client holding a different sample space but sharing the same feature space. In contrast, VFL involves vertical partitioning across different clients, with each client holding a different feature space but sharing the same sample space. Specifically, clients with distinct but complementary feature spaces collaboratively train models for mutual benefit. For instance, in healthcare, different hospitals hold various medical features for the same patients and collaborate on diagnostics. In multimodal scenarios, different modalities for the same samples are held by distinct clients (e.g., cloud service providers) due to privacy concerns (Peng et al. 2024), while they collaborate on a global target task, as shown in Fig. 1 (a). VFL initially protects privacy by (1) splitting the feature space so that each client holds only a partial dataset, and (2) training local models that map raw data into meaningful representations (embeddings) for transmission.

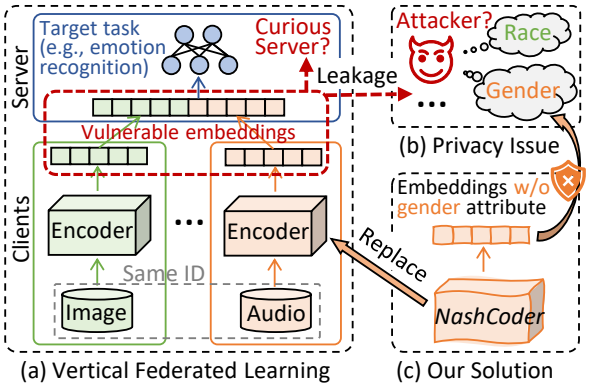

Figure 1: We aim to (1) design *NashCoder* to obfuscate sensitive attributes, and (2) balance the trade-off between global accuracy and local privacy from an equilibrium perspective.

However, an honest-but-curious server or a potential attacker may gain access to these embeddings and maliciously infer sensitive attributes (e.g., gender or race) that are unrelated to the target task by feeding the embeddings into an adversarial classifier. This type of threat is known as an attribute inference attack (Liu et al. 2022a) and is the focus of this work. Notably, the sensitive attributes are commonly predefined and privately known to each client when aiming to enhance privacy. However, they are not in the form of features that can be easily isolated or removed, and are inevitably reflected in the embeddings to some extent for two reasons. First, low-level features are entangled, e.g., the target task (emotion) prediction can be informally expressed as $\hat{y} = f(\boldsymbol{x})$, while the sensitive attribute (gender) prediction is $\hat{z} = g(\boldsymbol{x})$, both based on the input $\boldsymbol{x}$. Thus, sensitive attributes cannot be easily removed by dropping a few selected dimensions from $\boldsymbol{x}$, as one might do by deleting specific private columns from a CSV file. Second, $g(\cdot)$ is unknown in practice, making it nearly impossible to isolate the sensitive attribute. Thus, embeddings inherently carry privacy leakage risks and are vulnerable to attribute inference attacks.

Most existing privacy-enhancing methods in VFL, such as Differential Privacy (DP) (Tran et al. 2023), Homomorphic Encryption (HE) (Gong et al. 2023), and Secure Multiclient Computation (MPC) (Huang et al. 2023; Li, Yao, and Liu 2023), provide additional protection, but often at the cost of degraded target task accuracy and excessive privacy overhead, as they fail to identify which attributes are worth protecting. In contrast, adversarial learning-based privacy protection has gained significant attention in non-FL contexts (Wang et al. 2023; Li et al. 2021, 2020). This strategy enables a more tailored approach by enhancing privacy only for sensitive attributes without affecting other useful information, thereby incurring minor accuracy compromise for the downstream target task.

Challenges. However, VFL poses several new challenges. First, clients hold distinct feature spaces and may inherently possess different sensitive attributes, which we refer to as *heterogeneous privacy*, such as race for image clients, political affiliation for text clients, and gender for audio clients. Second, enhancing local privacy inevitably degrades global target task accuracy, reflecting the "no free lunch" principle and leading to an *accuracy-privacy trade-off*. Third, these challenges must be addressed in a distributed setting, where clients perform local optimization without a global view, making it difficult to balance the trade-off, as *privacy is adjusted locally while accuracy is achieved globally*.

Solutions. We propose an equilibrium-driven VFL framework with selective privacy protection for sensitive attributes to enhance privacy with minor accuracy compromise. To facilitate efficient distributed optimization, we decompose the global objective, which captures the accuracy-privacy trade-off, into multiple local objectives tailored to each client. Since each client can only optimize its local model with conflicting objectives, the problem naturally lends itself to a game-theoretic interpretation. We introduce two key innovations: (1) a *NashCoder*, which incorporates a surrogate head to jointly optimize for accuracy and privacy; and (2) a Shapley-based adaptive decomposition strategy that dynamically reshapes local objectives in a non-uniform yet fair manner, thereby constructing a better equilibrium model.

Contributions. (1) We formulate a novel and practical VFL problem that accounts for heterogeneous privacy and accuracy-privacy trade-off. (2) We design *NashCoder* to selectively obfuscate sensitive attributes, enhancing local privacy with minor global accuracy compromise. (3) We propose a Shapley-based adaptive decomposition strategy that dynamically reshapes local objectives to construct a better equilibrium model in a distributed setting. (4) We provide theoretical analysis and evaluate our framework on three public datasets against five baselines, demonstrating a better accuracy-privacy trade-off across various settings. Extensive experimental results support our theoretical analysis.

2 Related Work

There are two lines of VFL: one assigns an active party (label holder) and passive parties that transmit partial gradients, while the other, adopted in this work and widely used in recent VFL studies, transmits embeddings to enable flexible feature space partitioning and support arbitrary server models for feature fusion (Peng, Lu, and Xu 2024; Peng, Bian, and Xu 2024). Most existing VFL studies focus on two primary problems: (1) How to improve efficiency? Methods include compression techniques such as sparsification and quantization (Castiglia et al. 2022), asynchronous coordination (Castiglia, Wang, and Patterson 2023), and feature selection (Castiglia et al. 2023). (2) How to improve model effectiveness? Approaches include self supervised (He et al. 2024) and semi supervised (Sun et al. 2023) methods.

However, these works either rely on the flawed assumption that embeddings are privacy-preserving (which is not true) or adopt generic privacy strategies such as DP (Tran et al. 2023), HE (Gong et al. 2023), and MPC (Huang et al. 2023), some of which target inference attacks (Luo et al. 2021). Such one size fits all privacy enhancing strategies often degrade downstream task performance and incur excessive privacy overhead, as per the no free lunch principle.

Several adversarial learning based methods share a similar privacy enhancing motivation to obfuscate sensitive attributes (Wang et al. 2023; Li et al. 2021, 2020). However, they are limited to non-FL settings and are not applicable to VFL setting, where privacy is adjusted locally while accuracy is achieved globally. Put differently, in VFL, the conflicting objectives among clients make distributed optimization challenging, as each client must balance global accuracy gains with local privacy costs without a global view.

3 Problem Formulation

To clearly formulate our problem, we clarify the usage of several symbols: $[\cdot]$ denotes a set of elements, $\circ$ represents the composition of functions, and $\{\cdot\}$ denotes sets of features, labels, or models depending on the context.

We consider a VFL system consisting of K clients and a server. The dataset $\boldsymbol{x} \in \mathbb{R}^{N \times M}$ is vertically partitioned among the K clients, where N is the number of samples and M is the feature dimension. We define the dataset for client k ($k \in [K]$) as $\boldsymbol{x}_k \in \mathbb{R}^{N \times M_k}$ ($M = \sum_{k=1}^{K} M_k$). The i-th row of $\boldsymbol{x}$ corresponds to a sample x^i ($i \in [N]$), where each sample x^i is composed of feature subsets held by each client k, denoted as x_k^i, such that $x^i = \{x_1^i, \ldots, x_K^i\}$. Each x^i is associated with a target task label y^i and privacy task labels $\{z_k^i\}_{k=1}^K$, each corresponding to client-specific sensitive attributes. For example, if the global target task is emotion recognition, sensitive attributes may vary across different clients: race for images, political affiliation for text, and gender for audio. These attributes are unrelated to the global target task and are not in the form of easily isolated features, thus posing privacy risks (e.g., vulnerability to attribute inference attacks). Each client k locally holds an encoder model, parameterized by θ_k, implementing an embedding function $h_k(\cdot)$. The server holds the server head model, parameterized by θ_0, and a loss function $\ell(\cdot)$ for the target task, which combines embeddings received from clients.

Threat Model. A potential external attacker (or curious server) may gain access to the embeddings and maliciously infer sensitive attributes (e.g., gender or race) unrelated to the target task, as illustrated in Fig. 1(b). This is known as

an attribute inference attack, which can be formulated as a classification problem where the attacker feeds the embeddings into an adversarial classifier for prediction. However, in practice, clients lack knowledge of the attacker's model and only know which attributes (e.g., gender) are sensitive to them. To mimic the potential attacker's behavior, each client k locally trains a surrogate head g_k to approximate the attribute inference attack and employs a privacy loss function $\ell_k(\cdot)$ to measure the inference ability, as part of our approach detailed in Section 4.2. A similar threat model has been adopted in several privacy-related works in non-FL contexts (Li et al. 2020, 2021; Wang et al. 2023).

Unlike standard VFL formulations, we explicitly account for heterogeneous privacy and accuracy-privacy trade-off, both posing key challenges in real-world distributed settings, and integrate them into the global VFL objective:

$$\max_{\Theta} F(\Theta) := u(\Theta) - \alpha \cdot \textstyle\sum_{k\in[K]} c_k(\theta_k), \qquad (1)$$

where $\Theta = \{\theta_0, \theta_1, \ldots, \theta_K\}$ represents the global model, α controls the trade-off between accuracy and privacy, $u(\cdot)$ is a predefined accuracy utility function that quantifies the global model's performance on the target task, and $c_k(\cdot)$ is a predefined privacy cost function that captures the potential privacy risk for client k, with details deferred to Section 4.2. Our objective is to train the global model Θ in a distributed manner that maximizes the global value function $F(\Theta)$, defined as the difference between the globally achieved accuracy utility and the sum of local privacy costs.

While the global objective is well-defined and practical in real-world scenarios, solving it is challenging due to the distributed nature of feature-partitioned data. To enable distributed training, we decompose the global value function into a set of local value functions, one for each client k:

$$f_k(\Theta) := q_k \cdot u(\Theta) - \alpha \cdot c_k(\theta_k), \quad \forall k \in [K], \qquad (2)$$

where q_k is a decomposition weight that satisfies $\sum_k q_k = 1$. This decomposition allows us to express the global value function as $F(\Theta) := \sum_{k\in[K]} f_k(\Theta)$, enabling each client to focus on maximizing its local value, i.e., $\max_{\Theta} f_k(\Theta)$. Let $\boldsymbol{q} = \{q_1, \ldots, q_K\}$ denote the decomposition vector. The choice of decomposition for $\boldsymbol{q}$ can significantly influence the final training outcome, highlighting the importance of an effective decomposition strategy, detailed in Section 4.3.

A Novel Equilibrium Perspective. Although the global value function $F(\Theta)$ in Eq. (1) can be decomposed into multiple local value functions $f_k(\Theta)$ in Eq. (2), each client k cannot directly maximize its local value $f_k(\Theta)$ since it only maintains its local model θ_k, rather than the entire model Θ, i.e., the absence of a global view. To explicitly capture this dependency, we express the local value function $f_k(\Theta)$ as $f_k(\theta_k, \Theta_{-k})$, where $\Theta_{-k} = \{\theta_{k'}\}_{k'\neq k}$ denotes the rest of the model excluding θ_k. This lack of global control naturally leads to a game-theoretic interpretation, in which each client k is a player aiming to maximize its local value $f_k(\theta_k, \Theta_{-k})$ by optimizing its local model θ_k under Θ_{-k}. In this multi-agent setting, a Nash Equilibrium (NE) provides a natural solution concept, representing a stable state in which no client can improve its local value $f_k(\theta_k, \Theta_{-k})$ by unilaterally adjusting its local model θ_k.

Definition 3.1 (Equilibrium Model)**.** A global model Θ^{NE} is an equilibrium model if, for all $k \in [K]$,

$$f_k(\theta_k^{\text{NE}}, \Theta_{-k}^{\text{NE}}) \geq f_k(\theta_k', \Theta_{-k}^{\text{NE}}), \quad \forall \theta_k' \neq \theta_k^{\text{NE}}. \qquad (3)$$

Here, we focus on the equilibrium conditions among the K clients, excluding the server. Since all clients adjust their local models based on the same server head θ_0, i.e., their objectives remain aligned with θ_0, we can focus on optimizing the local models $\{\theta_1, \ldots, \theta_K\}$ in a distributed manner.

4 Methodology

4.1 Overview of our VFL Framework

To illustrate the core idea of VFL, we first describe an idealized training algorithm, disregarding practical communication and computation constraints. The training proceeds in R global rounds, where each round r consists of local updates performed independently by each client.

Idealized Training. At the start of round r, each client k initializes its local model as $\theta_k^{r,t_0} = \theta_k^{r-1}$, using the converged local model from the previous round, and trains it using local dataset $\boldsymbol{x}_k$. The local training process involves iteratively applying (stochastic) gradient descent, updating $\theta_k^{r,t}$ at each local step t, and continuing until convergence. Since each client k requires embeddings from other clients $\{k'\}_{k'\neq k}$ and server head θ_0 to compute its local partial gradients, information exchange is essential. However, communication occurs only at the end of each round r. Consequently, during local updates in round r, client k only has access to the embeddings from other clients $\{k'\}_{k'\neq k}$ from the last round. Let $h_k(\theta_k^r; \boldsymbol{x}_k)$ denote the embeddings generated by client k using its locally converged model θ_k^r. We define $\Phi^r = \{h_k(\theta_k^r; \boldsymbol{x}_k)\}_{k=1}^K$ as the concatenated embeddings from all clients at the end of round r. During local updates in round r, client k updates its local encoder θ_k based on fresh embeddings $h_k(\theta_k^{r,t}; \boldsymbol{x}_k)$ computed using its current encoder $\theta_k^{r,t}$, and stale embeddings from other clients $\{k'\}_{k'\neq k}$ from the last round, denoted as $\tilde{\Phi}_{-k}^r = \{h_{k'}(\theta_{k'}^{r-1}; \boldsymbol{x}_{k'})\}_{k'\neq k}$. At the end of each round r, clients upload their fresh embeddings to the server, which then redistributes the concatenated embeddings Φ^r and server head θ_0^r to all clients for the next round of training. Equivalently, by the end of each round r, each client k updates its local model θ_k^r by solving $\theta_k^r = \arg\max_{\theta_k} f_k(\theta_k, \Theta_{-k}^{r-1})$, while the global model Θ^r is updated by composing the locally updated models $\{\theta_k^r\}$.

Practical Training. While conceptually clear, the above idealized training paradigm poses significant computational and communication challenges. Each global round can be time-consuming, as it may require many local updates for model convergence. Besides, sharing embeddings over the entire dataset causes substantial communication overhead, which is also impractical. To mitigate computational and communication overhead, we adopt two practical training strategies: (1) Fixed number of local updates: We perform Q local updates per round to improve efficiency while maintaining effective model training. (2) Mini-batch training: Instead of updating on the entire dataset, training is performed

on a sampled mini-batch $\mathcal{B}$. These two efficiency improvement strategies have been widely adopted in recent VFL works (Castiglia et al. 2022, 2023; Castiglia, Wang, and Patterson 2023) and are orthogonal to our work, which focuses on heterogeneous privacy and accuracy-privacy trade-off.

4.2 *NashCoder*: Selective Privacy Protection

We now present *NashCoder*, which replaces the standard encoder in VFL and leverages adversarial training to selectively obfuscate (client-specific) sensitive attributes that are otherwise difficult to isolate and eliminate. We illustrate the proposed training methodology in Fig. 2.

The training of *NashCoder* is guided by the following two conflicting objectives, expressed from an information perspective: (1) *Global accuracy objective:* The output embedding h_k should capture as much useful information as possible to collaborate with other clients and achieve high accuracy on the global target task. (2) *Local privacy objective:* The output embedding h_k should convey as little sensitive attribute information as possible to enhance local privacy.

For the local privacy objective, as discussed in the threat model in Section 3, since the client lacks knowledge of the attacker's model and only knows which attribute is sensitive to itself, each client k locally trains a surrogate head g_k to approximate the behavior of a potential attacker (e.g., adversarial classifier) with the following objective:

$$\min_{g_k} \frac{1}{N} \sum_{i=1}^{N} \ell_k \left(g_k \circ h_k(\theta_k; x_k^i), z_k^i \right), \tag{4}$$

where $\ell_k(\cdot)$ is the local privacy task loss measuring the surrogate head's ability to infer sensitive attribute.

To jointly optimize global accuracy and local privacy in a distributed setting, the client locally trains the *NashCoder* θ_k and the surrogate head g_k under a given q_k and α by solving the following problem: $\min_{\theta_k}[q_k \cdot \ell(\cdot) - \alpha \cdot \min_{g_k} \ell_k(\cdot)]$, where $\ell(\cdot)$ represents the global target task loss. Furthermore, in practice, the globally achieved accuracy provides utility to all participants, modeled by a server-defined utility function $u(\cdot)$, which quantifies the real-world utility (e.g., monetary benefits) of the global model's performance. Similarly, local privacy leakage imposes a local cost on each client k, modeled by a client-defined cost function $c_k(\cdot)$, which captures real-world costs (e.g., monetary loss) as a function of the privacy attack's effectiveness. For training purposes, we let $u(\cdot)$ depend on the target task loss instead of the accuracy, and $c_k(\cdot)$ depend on the local privacy task loss instead of the attack accuracy. By combining the local privacy cost and the decomposed global accuracy utility, each client solves the following min-max optimization problem: $\min_{\theta_k}[\alpha \cdot \max_{g_k} c_k(g_k, \theta_k) - q_k \cdot u(\theta_k, \Theta_{-k})]$, and local updates alternate between θ_k and g_k in practice. Thus, given a q_k (e.g., $q_k = \frac{1}{K}$, $\forall k$) and α, we train θ_k using the following objective to address heterogeneous privacy and balance trade-off between global accuracy and local privacy:

$$\min_{\theta_k} \frac{1}{N} \sum_{i=1}^{N} \Big[\alpha \cdot \underbrace{c_k(\ell_k(g_k \circ h_k(\theta_k; x_k^i); z_k^i))}_{\text{Local privacy cost}} - \underbrace{q_k \cdot u(\ell(\tilde{\theta}_0 \circ \{h_k(\theta_k; x_k^i), \tilde{\Phi}_{-k}\}; y^i))}_{\text{Decomposed global accuracy utility}} \Big]. \tag{5}$$

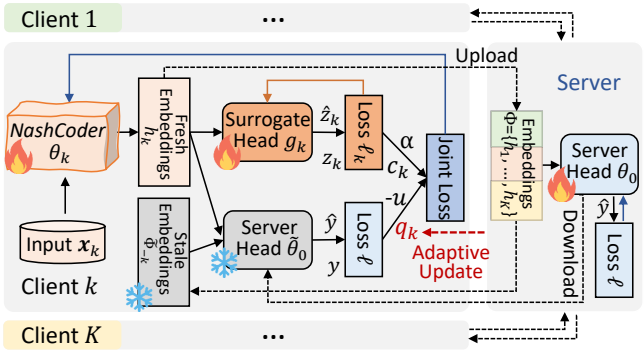

Figure 2: (1) Each client k selectively obfuscates client-specific sensitive attributes via *NashCoder* θ_k by incorporating a surrogate head g_k for local adversarial training. (2) The server adaptively updates the decomposition weight q_k for each client k. Downloads occur at the start of each global round, and uploads occur at the end. The spark denotes training, while the snowflake denotes a fixed state.

Note that $u(\cdot)$ and $c_k(\cdot)$ are predefined by the problem and not designed by our algorithm; their role is simply to convert different losses into meaningful values for joint accuracy and privacy training. As a toy example for illustration, we may set $u(x) = -x$, where the globally achieved accuracy utility is the negative of the target task loss; similarly for c_k. Besides, $\tilde{\theta}_0$ and $\tilde{\Phi}_{-k}$ are only shared at the start of each round for efficiency, which is widely adopted by VFL paradigms (Castiglia et al. 2022, 2023). Such efficiency improvements are orthogonal to our main contribution prioritizing accuracy-privacy trade-off.

4.3 Adaptive Decomposition

So far, we have assumed a fixed decomposition vector $\boldsymbol{q}$ and a given q_k for each client k in Eq. (5). However, the choice of $\boldsymbol{q}$ clearly impacts training outcomes, and it remains unclear which decomposition should be used. The simplest strategy is a uniform decomposition, i.e., $q_k = \frac{1}{K}$ for each client k, which promotes equal contributions to the global model performance, as shown in Fig. 3 (left). However, this method is clearly suboptimal, as different clients contribute unequally to the global target task depending on the importance of their provided embeddings. Another decomposition strategy that naturally arises is the weight-based approach, but it is still essentially a fixed decomposition, and it remains unclear how to fairly assign different q_k values to each client k in practice. Notably, accuracy is achieved globally and privacy is adjusted locally in VFL; thus, the trade-off balance differs from the non-FL setting and naturally lends itself to a game-theoretic interpretation. Different decomposition strategies for $\boldsymbol{q}$ lead to different games and result in different equilibrium models. Our goal is to find a better decomposition strategy to construct a better equilibrium model by reshaping the value functions $\{f_k\}_{k\in[K]}$ in Eq. (2), thereby enhancing overall outcomes reflected in a better trade-off.

To this end, we propose an adaptive decomposition strategy from an equilibrium perspective, which dynamically ad-

justs the decomposition weights $\{q_k\}_{k\in[K]}$ based on each client's contribution. Specifically, we use the *Shapley value* to evaluate each client's marginal contribution to the global target task at the end of each round in a non-uniform yet fair manner, which guides the decomposition of $\boldsymbol{q}$ in the next round. For clarity, we omit $\ell(\cdot)$ and other unnecessary notations below. To ensure $\sum_{k=1}^{K} q_k = 1$, we use a normalized utility for any subset of clients $S \subseteq [K]$, computed as: $\hat{u}(\Theta^r[S]) = \frac{u(\Theta^r[S])-u(\Theta^r[\emptyset])}{u(\Theta^r)-u(\Theta^r[\emptyset])}$. Here, $u(\Theta^r[S])$ is the accuracy utility achieved by coalition S, and $u(\Theta^r[\emptyset])$ is the utility achieved by the empty coalition. The marginal contribution of each client k to the global model performance is quantified by how much the accuracy utility increases when adding k to a coalition S. For any coalition S, the marginal contribution of client k is $\hat{u}(\Theta^r[S \cup \{k\}]) - \hat{u}(\Theta^r[S])$, Thus, the decomposition weight q_k is computed by: $q_k^{r+1} = \sum_{S\subseteq[K]\setminus\{k\}} \frac{|S|!(K-|S|-1)!}{K!} \cdot (\hat{u}(\Theta^r[S\cup\{k\}]) - \hat{u}(\Theta^r[S]))$.

An important motivation for using the Shapley value is that the decomposition should be entirely determined by the server, based solely on each client's marginal contribution rather than client-reported local values f_k that incorporate client-defined privacy costs c_k; that is, the decomposition is privacy-cost-agnostic to avoid imbalances caused by c_k. Moreover, most existing studies (Liu et al. 2024, 2022b; Castiglia et al. 2022) suggest that the number of clients K is typically small in VFL, as the number of splittable features (modalities) is limited in practice (unlike HFL, which involves many clients). Thus, the computational cost of calculating exact q_k on a powerful server is relatively minor. We offer a sampling-based approximation for q_k to address potential scalability issues in extreme cases where K is large. However, we do not claim this efficiency improvement as our core contribution, as it is orthogonal to our main objective of balancing the trade-off from an equilibrium view.

4.4 Theoretical Analysis

We first demonstrate that, under common L-smoothness assumptions (L_1, L_2, L_3), our idealized training with adaptive decomposition converges to an equilibrium model.

Theorem 4.1 (Convergence). *Define* $C \triangleq \sum_k \sum_{S\subseteq[K]\setminus\{k\}} \frac{|S|!(K-|S|-1)!}{K!}$. *If* L_1, L_2, L_3 *satisfy* $2L_1L_3C + L_2 < 1$, *then the global model* Θ *and decomposition vector* $\boldsymbol{q}$ *obtained by our distributed algorithm converge to stable* $\Theta^\dagger$ *and* $\boldsymbol{q}^\dagger$. *Moreover,* $\Theta^\dagger$ *is an equilibrium model under* $\boldsymbol{q}^\dagger$.

Furthermore, given any decomposition vector $\boldsymbol{q}$ and trade-off parameter α, we bound the gap between the corresponding NE and the global optimum under common smoothness assumptions parameterized by constants L, L_0, L_c, and L_q, where $L_c = \max_{k\in[K]} L_{c,k}$ and $L_q = \max_{k\in[K]} L_{q,k}$, which depend on the privacy cost functions $\{c_k\}_{k\in[K]}$ and the decomposition weights $\{q_k\}_{k\in[K]}$, respectively.

Theorem 4.2 (Bound between NE and the global optimum). *If* $1 - L_q(K-1) > 0$, *let* Θ^{OPT} *denote the global optimum and* $\Theta^{NE}(\boldsymbol{q})$ *denote the NE solution of* $F(\Theta)$ *under* $\boldsymbol{q}$. *Then, the gap between* $F(\Theta^{OPT})$ *and* $F(\Theta^{NE}(\boldsymbol{q}))$ *is upper bounded by:* $\|F(\Theta^{OPT}) - F(\Theta^{NE}(\boldsymbol{q}))\| \le \alpha L(L_0 + \frac{L_c K}{1-L_q(K-1)})$.

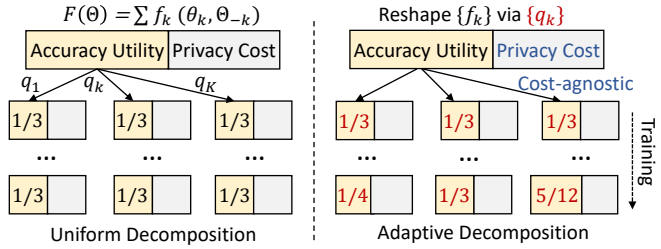

Figure 3: We construct a better NE by adaptively reshaping the value functions through global objective decomposition.

This upper bound aligns with the intuition that the NE solution approaches the global optimum as $\alpha \to 0$ (i.e., all clients care only about global accuracy utility and ignore local privacy cost, as in standard VFL). Detailed assumptions and proofs of Theorems 4.1 and 4.2 are provided in the Appendix (Peng 2025) due to page limit. The "Theory vs. Practice" discussion is deferred to Section 5.4, showing that extensive experimental results under various privacy settings align with our theoretical analysis.

5 Experiments

We aim to investigate the following three questions in our experiments to validate our framework and theoretical analysis. The corresponding results and analyses are presented in Sections 5.2, 5.3, and 5.4, respectively: **Q1:** Can we achieve a better trade-off when sensitive attributes are homogeneous across clients? **Q2:** Can we achieve a better trade-off when sensitive attributes are heterogeneous? **Q3:** How and why do key components and parameters affect performance?

5.1 Experimental Setup

Datasets: We adopt three publicly available datasets covering different VFL applications and tasks. (1) *IEMOCAP* consists of visual, text, and audio modalities recorded from multiple actors during conversations (Busso et al. 2008). The target task is set as emotion recognition, while the privacy task is set as gender prediction, as gender is unrelated to the target task but can be maliciously inferred. (2) *MIMIC-III* is a collection of medical records of patients admitted to ICUs. Each sample has up to 76 features, such as vital signs and medications within a period (Johnson et al. 2016). The target task is set as the prediction of in-hospital mortality (IHM). The privacy task is set as predicting a length of stay (LOS) longer than two days, as this attribute can be maliciously inferred for strategic insurance pricing. (3) *CelebA* contains a large-scale collection of facial images, each labeled with 40 binary attributes (Liu et al. 2015). The predictions of "smiling" and "gray hair" are set as different global target tasks, while the predictions of "male" and "young" are set as different privacy tasks (client-specific sensitive attributes).

Baselines: (1) *Standard VFL* transmits embeddings instead of feature-partitioned data but is vulnerable to potential attacks. (2) *DP-Gaussian* applies DP in VFL by injecting Gaussian noise to enhance privacy (Wang et al. 2024). (3) *DP-Laplacian* adopts DP in VFL by injecting Laplacian noise (Ovi et al. 2023). (4) *C-VFL* employs a compression

strategy by pruning embeddings, which enhances privacy as a byproduct (Castiglia et al. 2022). (5) *Ours with uniform decomposition* ($q_k = \frac{1}{K}, \forall k$) serves as an important baseline to highlight the advantages of our adaptive decomposition.

Implementation and Metrics: We run all experiments on NVIDIA GeForce RTX 4090 GPUs. Essential implementation details and metrics are provided below. For some imbalanced datasets, we use the F1 score instead of accuracy. In the following, (↑) and (↓) indicate that higher or lower values correspond to better performance. For IEMOCAP, we partition the data across three modalities for three clients. The privacy task metric is the F1 score for gender prediction (↓), and the target task metric is the weighted F1 score for emotion recognition (↑). For MIMIC-III, we partition up to 76 features across two clients and increase the number of clients K for scalability experiments in Section 5.4. The privacy task metric is the F1 score for LOS>2 (↓), and the target task metric is the F1 score for IHM (↑), since only 16% of samples are positive (i.e., most patients did not experience IHM). For CelebA, each image is vertically split into two parts for two clients, which is commonly used for feature partitioning in VFL (Sun et al. 2023; Castiglia et al. 2022). We use the potential attacker's accuracy for predicting "male" and "young" as metrics for local privacy task performance (↓), and accuracy for predicting "smiling" and "gray hair" as metrics for two different global target tasks (↑). Notably, the privacy metrics, evaluated from the potential attacker's perspective, align with those used in adversarial-learning-based methods in non-FL contexts (Li et al. 2020, 2021; Wang et al. 2023), and are widely adopted in attribute inference attacks (Liu et al. 2022a).

5.2 Results of Homogeneous Sensitive Attributes

To answer **Q1**, we first set the sensitive attributes of different clients to be homogeneous (note that clients are unaware of this homogeneity and c_k may differ). For each dataset, we present two types of results. First, we compare our method against the baselines to observe the *overall trend* in the accuracy–privacy trade-off. Each method (except the standard VFL, which ignores privacy) is repeated five times under different values of α. In other words, each point with the same marker represents the performance of a method under a fixed α. The horizontal axis indicates the global target task performance collaboratively achieved by all clients, while the vertical axis indicates the average privacy performance across all clients. As shown in Fig. 4 on IEMOCAP and MIMIC-III, our method consistently approaches the optimal accuracy–privacy trade-off compared with the baselines. The lower right corner corresponds to better target task accuracy from the server's view and worse privacy task accuracy from the surrogate's view, as our method selectively obfuscates sensitive attributes with less accuracy compromise. Notably, our adaptive decomposition achieves better performance by assigning higher weights to clients with greater contributions in a non-uniform yet fair manner.

Second, we show each client's *individual performance* under a fixed α. The horizontal axis still represents the target task performance, while the vertical axis indicates each client's individual privacy task performance (solid, hollow, and bordered markers denote different clients). As shown in Fig. 5, our method achieves a better trade-off, as clients not only collaboratively improve global accuracy utility but also reduce local privacy costs to maximize their individual value. Put differently, we construct a better equilibrium model by adaptively reshaping the value functions.

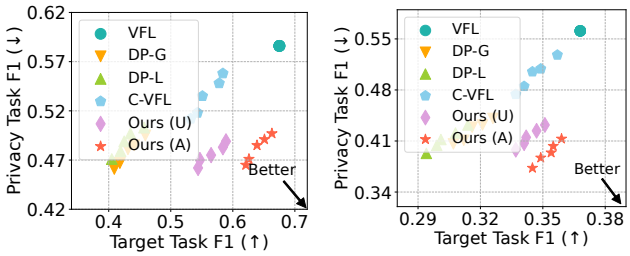

Figure 4: Overall trade-off trend between global accuracy and (average) local privacy on IEMOCAP and MIMIC-III.

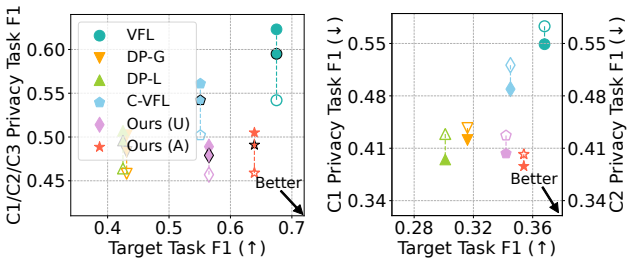

Figure 5: Individual trade-off between accuracy and privacy.

5.3 Results of Heterogeneous Sensitive Attributes

To answer **Q2**, we conduct experiments on CelebA by assigning heterogeneous privacy tasks ("male" and "young") to two clients (each with half of the facial image) associated with the same target task. Besides, we perform experiments on two different target tasks ("smiling" and "gray hair") to demonstrate broad applicability. As shown in Fig. 6, our method achieves a better accuracy-privacy trade-off than baselines. Thus, it is suitable for personalized privacy objectives commonly seen in real-world distributed settings, where accuracy is achieved globally and privacy is adjusted locally. Here, the server adjusts decomposition weights solely based on each client's utility contribution.

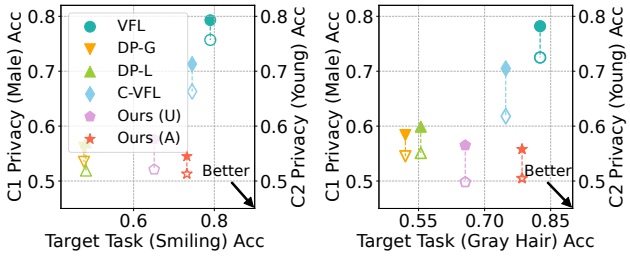

Figure 6: Client-level accuracy-privacy trade-off when sensitive attributes are heterogeneous under two target tasks.

5.4 Impact of Key Components and Parameters

To answer **Q3**, we conduct more experiments including ablation studies. Results on MIMIC-III are presented, with similar trends observed on the other two datasets.

Impact of Adversarial Training. As shown in Fig. 7 (a), our method selectively obfuscates sensitive attributes without affecting other useful information via adversarial training. Thus, for the target task, our method closely approaches the performance of standard VFL, which is impractical in the real world as it ignores privacy cost. Besides, although the surrogate head is a shallow model and efficient to train in practice, we further investigate its computational cost. As shown in Fig. 7 (b), the smaller the shadow areas, the better, i.e., privacy can be enhanced via local adversarial training.

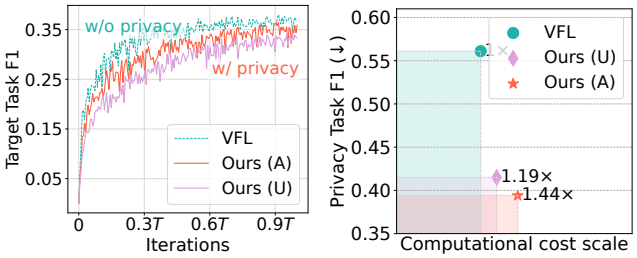

Figure 7: (a) Convergence (b) Impact of adversarial training.

Impact of Adaptive Decomposition. As shown in Fig. 8 (left), our method with adaptive decomposition achieves a better accuracy-privacy trade-off compared to uniform decomposition. This is because the server dynamically adjusts q_k to reshape the value functions f_k for each client k based on their past contributions in a non-uniform yet fair manner, thereby helping construct a better equilibrium model in a distributed manner and improving overall outcomes.

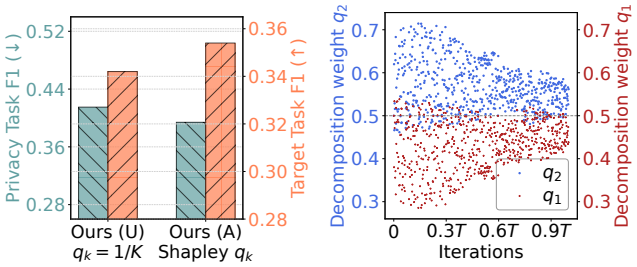

Figure 8: Impact of adaptive decomposition and dynamic q_k.

Impact of α. As shown in Fig. 9 (left), increasing α causes the value function to assign greater weight to privacy cost. However, changes in α are applied equally to all clients, i.e., α can only balance the trade-off between accuracy and privacy, but it cannot reshape the value functions to construct a better equilibrium model, as our method does.

Impact of K. As shown in Fig. 9 (right), increasing K slightly degrades the global target task performance, as a fixed number of feature dimensions is distributed across more clients, causing each client to maintain a narrower feature space and rely on a larger proportion of stale information during local updates. Moreover, a larger K also decreases the F1 score of the privacy task, as fewer (split) features make it harder for the surrogate model to infer sensitive attributes. Besides, since the number of splittable features (modalities) is limited in practice, K is typically small in VFL. As a result, scalability becomes a relatively minor concern compared to the accuracy and privacy trade off.

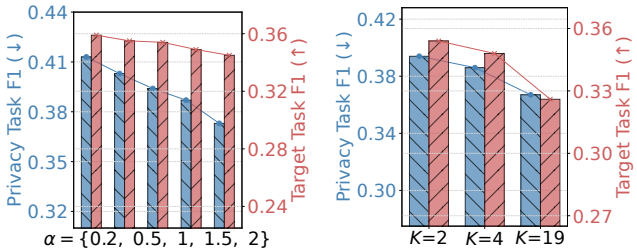

Figure 9: Impact of α (left) and impact of K (right).

Theory vs. Practice. Extensive experimental results in Figs. 4 to 6 under homogeneous and heterogeneous sensitive attributes, the convergence and improved privacy performance via adversarial training in Fig. 7, the adaptive decomposition results in Fig. 8, and the evaluation across different trade-off parameters α and numbers of clients K in Fig. 9 collectively demonstrate the effectiveness of our framework.

Importantly, these results align with our theoretical analysis in Theorem 4.1, which shows that the global model Θ and decomposition vector $\boldsymbol{q}$, obtained through our distributed algorithm, converge to stable values $\Theta^\dagger$ and $\boldsymbol{q}^\dagger$, where $\Theta^\dagger$ is an equilibrium model under $\boldsymbol{q}^\dagger$. Such an equilibrium model achieves a better trade-off between global accuracy and local privacy. Furthermore, we show in Theorem 4.2 that our NE solution approaches the global optimum as $\alpha \to 0$ (i.e., when all clients care only about global accuracy and ignore local privacy costs, as in standard VFL), thereby highlighting the practicality of our distributed algorithm in real-world scenarios, where clients not only benefit from global accuracy gains via collaboration but also incur local privacy costs that must be carefully balanced in a distributed setting.

6 Conclusion

In conclusion, we formulate a novel and practical VFL problem that explicitly accounts for heterogeneous privacy and the accuracy-privacy trade-off, and propose an equilibrium-driven VFL framework to solve it via global objective decomposition. We introduce two key innovations: (1) *NashCoder*, which incorporates a surrogate head to jointly optimize global accuracy and local privacy; and (2) an adaptive decomposition strategy based on Shapley values, which dynamically decomposes the global objective for distributed optimization from an equilibrium perspective. We theoretically analyze our framework and empirically evaluate it on three public datasets against five baselines, demonstrating significant improvements in the accuracy-privacy trade-off under various privacy settings. Extensive experimental results further support our theoretical analysis.

Acknowledgments

The work of Yuanzhe Peng and Jie Xu is partially supported by NSF under grants 2505381 and 2515982. The work of Wenwei Zhao and Zhuo Lu is partially supported by NSF under grant 2319781. Any opinions, findings, and conclusions expressed in this paper are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

Busso, C.; Bulut, M.; Lee, C.-C.; Kazemzadeh, A.; Mower, E.; Kim, S.; Chang, J. N.; Lee, S.; and Narayanan, S. S. 2008. IEMOCAP: Interactive Emotional Dyadic Motion Capture Database. *Language Resources and Evaluation*, 42: 335–359.

Castiglia, T.; Wang, S.; and Patterson, S. 2023. Flexible Vertical Federated Learning with Heterogeneous Parties. *IEEE Transactions on Neural Networks and Learning Systems*.

Castiglia, T.; Zhou, Y.; Wang, S.; Kadhe, S.; Baracaldo, N.; and Patterson, S. 2023. LESS-VFL: Communication-Efficient Feature Selection for Vertical Federated Learning. In *International Conference on Machine Learning*, 3757–3781. PMLR.

Castiglia, T. J.; Das, A.; Wang, S.; and Patterson, S. 2022. Compressed-VFL: Communication-Efficient Learning with Vertically Partitioned Data. In *International Conference on Machine Learning*, 2738–2766. PMLR.

Gong, M.; Zhang, Y.; Gao, Y.; Qin, A. K.; Wu, Y.; Wang, S.; and Zhang, Y. 2023. A Multi-Modal Vertical Federated Learning Framework Based on Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*.

He, Y.; Kang, Y.; Zhao, X.; Luo, J.; Fan, L.; Han, Y.; and Yang, Q. 2024. A Hybrid Self-Supervised Learning Framework for Vertical Federated Learning. *IEEE Transactions on Big Data*.

Huang, Y.; Wang, W.; Zhao, X.; Wang, Y.; Feng, X.; He, H.; and Yao, M. 2023. EFMVFL: An Efficient and Flexible Multi-party Vertical Federated Learning without a Third Party. *ACM Transactions on Knowledge Discovery from Data*, 18(3): 1–20.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, A Freely Accessible Critical Care Database. *Scientific Data*, 1–9.

Li, A.; Duan, Y.; Yang, H.; Chen, Y.; and Yang, J. 2020. TIPRDC: Task-Independent Privacy-Respecting Data Crowdsourcing Framework for Deep Learning with Anonymized Intermediate Representations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 824–832.

Li, A.; Guo, J.; Yang, H.; Salim, F. D.; and Chen, Y. 2021. DeepObfuscator: Obfuscating Intermediate Representations with Privacy-Preserving Adversarial Learning on Smartphones. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 28–39.

Li, S.; Yao, D.; and Liu, J. 2023. FedVS: Straggler-Resilient and Privacy-Preserving Vertical Federated Learning for Split Models. In *International Conference on Machine Learning*, 20296–20311. PMLR.

Liu, Y.; Kang, Y.; Zou, T.; Pu, Y.; He, Y.; Ye, X.; Ouyang, Y.; Zhang, Y.-Q.; and Yang, Q. 2024. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36: 3615–3634.

Liu, Y.; Wen, R.; He, X.; Salem, A.; Zhang, Z.; Backes, M.; De Cristofaro, E.; Fritz, M.; and Zhang, Y. 2022a. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. In *31st USENIX Security Symposium*, 4525–4542.

Liu, Y.; Zhang, X.; Kang, Y.; Li, L.; Chen, T.; Hong, M.; and Yang, Q. 2022b. FedBCD: A Communication-Efficient Collaborative Learning Framework for Distributed Features. *IEEE Transactions on Signal Processing*, 70: 4277–4290.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision*, 3730–3738.

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

Ovi, P. R.; Dey, E.; Roy, N.; and Gangopadhyay, A. 2023. Mixed Quantization Enabled Federated Learning to Tackle Gradient Inversion Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5046–5054.

Peng, Y. 2025. Appendix. URL: https://bit.ly/4qQepzj.

Peng, Y.; Bian, J.; and Xu, J. 2024. FedMM: Federated Multi-Modal Learning with Modality Heterogeneity in Computational Pathology. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1696–1700. IEEE.

Peng, Y.; Lu, Z.; and Xu, J. 2024. Joint Horizontal and Vertical Federated Learning for Multimodal IoT. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2371–2376.

Peng, Y.; Wu, Y.; Bian, J.; and Xu, J. 2024. Hybrid Federated Learning for Multimodal IoT Systems. *IEEE Internet of Things Journal*.

Sun, J.; Xu, Z.; Yang, D.; Nath, V.; Li, W.; Zhao, C.; Xu, D.; Chen, Y.; and Roth, H. R. 2023. Communication-Efficient Vertical Federated Learning with Limited Overlapping Samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5203–5212.

Tran, L.; Castiglia, T.; Patterson, S.; and Milanova, A. 2023. Privacy Tradeoffs in Vertical Federated Learning. In *Federated Learning Systems (FLSys) Workshop@MLSys 2023*.

Wang, G.; Gu, B.; Zhang, Q.; Li, X.; Wang, B.; and Ling, C. X. 2024. A Unified Solution for Privacy and Communication Efficiency in Vertical Federated Learning. *Advances in Neural Information Processing Systems*, 36.

Wang, Z.; Wang, H.; Jin, S.; Zhang, W.; Hu, J.; Wang, Y.; Sun, P.; Yuan, W.; Liu, K.; and Ren, K. 2023. Privacy-Preserving Adversarial Facial Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8212–8221.