A UNIFIED FRAMEWORK FOR VERTICAL FEDERATED LEARNING WITH PARTIALLY MISSING FEATURES AND LOCALLY INDEPENDENT INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Vertical Federated Learning enables collaborative training across clients holding disjoint feature subsets of data samples. Existing methods often treat a client's local features as either fully available or entirely missing, and thus fail to make full use of partially observed features. Moreover, standard VFL typically relies on joint inference involving all participating clients, limiting the ability of an individual client to make predictions independently. To address these limitations, we propose X-VFL, a unified framework that uses non-aligned data with partially missing features and supports locally independent inference for each client. Specifically, X-VFL introduces two modules: *Cross Completion* (XCom), which reconstructs missing local features for non-aligned samples, and *Decision Subspace Alignment* (DS-Align), which aligns local, completed, and joint representations in the decision subspace to improve independent inference. Furthermore, we establish convergence guarantees for X-VFL, proving an $O(1/\sqrt{T})$ rate for SGD-type and $O(1/T)$ for PAGE-type algorithms, where T denotes the number of training steps. Extensive experiments on real-world datasets demonstrate that X-VFL significantly outperforms existing methods across different feature missing rates, e.g., achieving an improvement of around 43 percentage points on the medical `MIMIC-III` dataset, validating X-VFL's practical effectiveness and robustness.

1 INTRODUCTION

Federated Learning (FL) has emerged as a transformative paradigm for privacy-preserving collaborative learning, enabling multiple parties to jointly train models without disclosing raw data (McMahan et al., 2017; Kairouz et al., 2021). Based on sample and feature space partitioning, FL is typically categorized into Horizontal FL (HFL) and Vertical FL (VFL) (Yang et al., 2019; Liu et al., 2024). In HFL, clients hold different datasets sharing the same feature space, whereas in VFL, all clients use the same data samples but each holds a different subset of features. This paper focuses on VFL, which is particularly useful in domains such as finance, healthcare, and e-commerce, where participating clients/parties often possess complementary feature sets (Joshi et al., 2022).

Despite its potential, real-world VFL deployment is constrained by two fundamental hurdles: *a rigid dependency on perfect data alignment* (Castiglia et al., 2022; Kang et al., 2022) and *the requirement for joint collaborative inference* (Li et al., 2022b; Ren et al., 2022). Empirically, only a tiny fraction of potential VFL pairs can be perfectly aligned (Wu et al., 2025). We formalize this critical scenario in Fig. 1. In a two-client setting, each client i possesses a feature set $\boldsymbol{X}_i = (\boldsymbol{X}_i^{al}, \boldsymbol{X}_i^{nl})$, where $\boldsymbol{X}_i^{al}$ represents aligned samples with complete local feature blocks, and $\boldsymbol{X}_i^{nl}$ denotes non-aligned samples where at least one client suffers from partially missing features. Such partial missingness is frequently induced by asynchronous

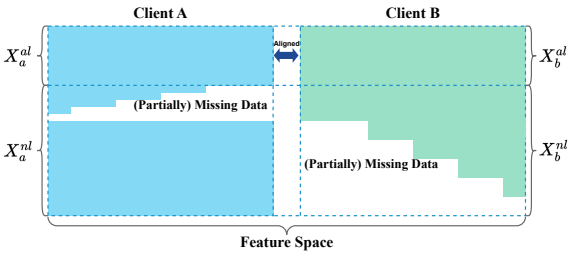


Figure 1: Dataset partition in VFL with two clients.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

data collection protocols or restricted data access. Existing VFL frameworks typically treat these incomplete partitions as total absence, causing substantial information waste and diminished model utility. Furthermore, the concatenation-based aggregation in vanilla VFL establishes a rigid dimensional dependency that prevents a single client from performing predictions when communication links fail or other parties are unavailable.

Recent studies have sought to address these challenges through various methodologies, such as knowledge distillation (Ren et al., 2022), self-supervised learning (Li et al., 2022b; Kang et al., 2022; He et al., 2024), and representation learning (Huang et al., 2023; Hong et al., 2026). Despite these gains, many frameworks are restricted to two-party settings (Li et al., 2022b; Kang et al., 2022) or fail to fully utilize unaligned data during training (Ren et al., 2022; Huang et al., 2023; He et al., 2024). Furthermore, current state-of-the-art methods predominantly address coarse-grained block-wise missingness (Valdeira et al., 2025), where feature partitions are either entirely present or absent. Consequently, the pervasive challenge of fine-grained intra-party partial missingness remains unaddressed.

To bridge these gaps, we propose X-VFL, a novel VFL framework designed to exploit non-aligned data samples with partial missingness while supporting locally independent inference of new data samples for each client. Specifically, we develop two key modules: *Cross Completion* (XCom) and *Decision Subspace Alignment* (DS-Align). The XCom module is developed to establish cross-complementary dependencies among disjoint local features contributed by different clients. Leveraging these relationships, local clients can complete their missing features through XCom, thereby effectively increasing the volume of data available for training and inference. The DS-Align module is designed to align the features across all clients within the decision subspace to support locally independent inference for each client, while maintaining performance comparable to collaborative inferences involving all clients. In addition, X-VFL employs an averaging-based aggregation strategy that maintains constant dimensionality across both local and joint embeddings. This design eliminates architectural dependencies and allows the top model to be applied to any subset of participants, enabling seamless independent inference without structural modifications.

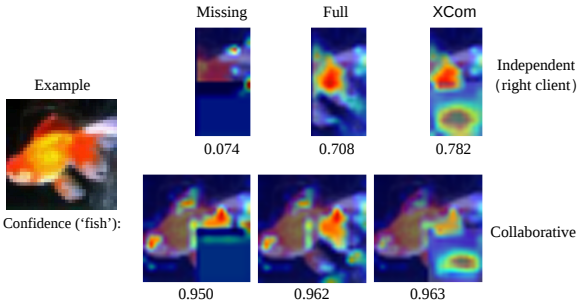
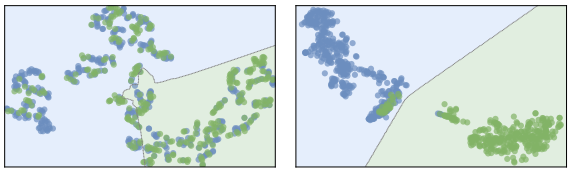


Figure 2: Effect of XCom.

Fig. 2 illustrates the effect of XCom on a 'fish' sample from the `TinyImageNet` dataset in a two-client VFL scenario where the image is vertically partitioned into left and right halves. The first row displays the independent inference results for the right client specifically when the features in its bottom-right region are missing. Without completion, the local model fails to identify the object, yielding a negligible confidence score of 0.074 and an uninformative attention heatmap. However, our XCom module effectively restores the local predictive power by reconstructing the missing information, achieving a confidence of 0.782 which slightly surpasses the 0.708 observed in the full-feature case. The second row illustrates the collaborative inference results involving both participants, where the joint model achieves a confidence of 0.950 under the missing scenario. With the integration of XCom, this collaborative confidence increases to 0.963, effectively matching the 0.962 full-feature baseline. Notably, the attention heatmaps for the XCom completed cases reveal that the visual focus is concentrated precisely on the reconstructed regions. This demonstrates that the information generated by XCom possesses clear semantic meaning and plays a pivotal role in the model's decision-making process by successfully restoring critical discriminative features.



(a) Vanilla (b) DS-Align

Figure 3: Effect of DS-Align.

The proposed DS-Align module unifies local individual features with completed and joint representations across all clients to ensure the decision space remains discriminative even when data is incomplete. Fig. 3 provides a t-SNE visualization (Van der Maaten & Hinton, 2008) of the embedding distributions and the resulting decision boundary for the gender classification task on the UTKFace dataset. As shown in the figure, the "Vanilla" setting in Fig. 3a results in a poorly defined decision boundary with significant overlapping regions, leading to frequent misclassifications between male and female classes. In contrast, our "DS-Align" setting (Fig. 3b) yields a much clearer decision boundary, ensuring better separation between classes. This confirms the effectiveness of DS-Align in inference under our X-VFL framework.

Our contributions. The main contributions of this work are as follows:

- We move beyond the conventional block-wise missingness and introduce a practical setting for partially missing features. We define a missing rate (R_{miss}) to quantify intra-party incompleteness, providing a more granular and realistic benchmark for VFL.
- We propose X-VFL, a novel VFL framework designed to handle the non-aligned data samples with (partially) missing features and to support locally independent inference for new data samples at each client. More importantly, our X-VFL introduces two key modules: Cross Completion (XCom) and Decision Subspace Alignment (DS-Align), which significantly enhance the ability of VFL to address more complex and practical scenarios.
- We provide theoretical convergence theorems for the algorithms used in training X-VFL, showing an $O(1/\sqrt{T})$ convergence rate for SGD-type algorithms and an $O(1/T)$ rate for PAGE-type algorithms, where T denotes the number of training update steps (see Theorems 1 and 2 in Section 3).
- Finally, we conduct extensive experiments on real-world datasets to demonstrate that X-VFL significantly outperforms existing VFL methods, e.g., achieving a 15% improvement in accuracy on CIFAR-10 and a 43% improvement on the medical dataset MIMIC-III (see Section 4), underscoring its practical superiority in complex, real-world environments.

1.1 Related work

VFL enables distributed learning across vertically partitioned data, where clients hold disjoint features for the same samples(Liu et al., 2024). Many frameworks adopt Split Neural Networks (SplitNNs)(Vepakomma et al., 2018; Thapa et al., 2022), where each client trains a local model up to a predefined cut layer. These intermediate outputs are transmitted to a central server holding the labels to complete forward and backward propagation. Gradients are then returned for local model updates until convergence.

Research on SplitNN-based VFL mainly focused on enabling locally independent inference and handling non-aligned data samples (Liu et al., 2024). For instance, Ren et al. (2022) introduced a knowledge distillation framework, IAVFL, that transfers insights from joint training to local models, enabling independent inference based solely on local features. Subsequent studies extended this concept through representation-level distillation to capture inter-client feature correlations and enhance independent inference capabilities (Lin et al., 2020; Li et al., 2022a; Huang et al., 2023; He et al., 2024). However, these distillation-based methods are restricted to aligned samples, thus providing limited improvement in inference performance.

Another promising direction leverages non-aligned data to improve model performance (Li et al., 2022b; Xu et al., 2023). Recent efforts have applied self-supervised learning (SSL) techniques to enhance local model representations using non-aligned data samples (Castiglia et al., 2022; Xu et al., 2023). By improving the representational quality of local models, SSL typically contributes to better global model performance in VFL frameworks. However, these methods still require collaboration among all clients during the inference, restricting their practical applicability. Besides, Kang et al. (2022) proposed a semi-supervised learning framework, FedCVT, that estimates the corresponding embeddings/representations for the missing features and infers the pseudo labels for the missing labels, thus increasing the amount of data embeddings available for training. Valdeira et al. (2025) proposed LASER-VFL, a VFL framework that enables training and inference with subsets of feature blocks by training a family of predictors with task-sampling. However, it only considers the fully missing feature setting and does not support partially missing features. Moreover, it lacks the ability

to complete/reconstruct missing features. Therefore, advancing practical VFL demands a novel framework that supports independent inference and efficiently integrates and completes non-aligned samples with partially missing features during both training and inference.

2 X-VFL FRAMEWORK

This section introduces X-VFL, a novel framework designed to overcome vanilla VFL limitations by handling partially missing features and enabling locally independent inference. In Section 2.1, we detail the problem setup and our two core modules, Cross Completion (XCom) and Decision Subspace Alignment (DS-Align), using a two-client setting for clarity. We then generalize the formulation and loss functions to a k-client setting in Appendix A. Finally, we present the operational details of both collaborative and independent inference modes.

2.1 FRAMEWORK DESCRIPTION

As illustrated in Fig. 4, our X-VFL is built upon two key modules: Cross Completion (XCom) and Decision Subspace Alignment (DS-Align). Each client hosts a local/bottom model f_i and an XCom module. The local model f_i processes its local features to high-level embeddings/representations, denoted as $\boldsymbol{E}_i = f_i(\boldsymbol{x}_i)$, while the XCom module reconstructs missing features using embeddings received from other clients. Once the embeddings, whether derived from existing features, reconstructed features, or a combination of both, are obtained, they are either directly fed into the top model h or averaged before being used as input. The final loss is computed by combining the decision cross-entropy (CE) loss and the loss from the DS-Align module, followed by standard backpropagation.

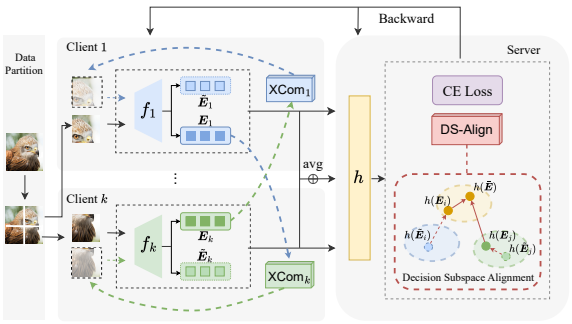


Figure 4: The framework of X-VFL.

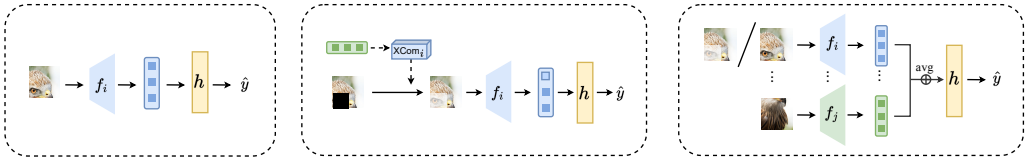
Crucially, instead of traditional concatenation ($\boldsymbol{E} = [\boldsymbol{E}_1, \boldsymbol{E}_2, \cdots, \boldsymbol{E}_k]$), X-VFL aggregates embeddings via averaging: $\boldsymbol{E}_{\text{avg}} = \text{Avg}(\boldsymbol{E}_1, \boldsymbol{E}_2, \cdots, \boldsymbol{E}_k)$. This paradigm shift naturally enables *independent inference*. Concatenation creates a dimensionality mismatch between individual and global embeddings, strictly mandating multi-client collaborative inference. In contrast, our averaging strategy ensures each local embedding $\boldsymbol{E}_i$ shares the exact dimensionality as $\boldsymbol{E}_{avg}$ allowing any single client to directly query the top model h for local predictions.

Cross Completion. The XCom module is designed to complete missing features for non-aligned data samples by leveraging information from other clients, thereby increasing the amount of usable training data and improving the inference performance.

To better illustrate this, we first consider a two-client setting with clients A and B (the general case with multiple k clients is described in Appendix A). Let f_a and f_b denote their bottom models, respectively. Their (partially) missing features can be completed using the other client's embeddings as follows:

$$\tilde{\boldsymbol{X}}_a = \mathsf{XCom}_a(\boldsymbol{E}_b), \ \tilde{\boldsymbol{X}}_b = \mathsf{XCom}_b(\boldsymbol{E}_a), \tag{1}$$

where $\boldsymbol{E}_b = f_b(\boldsymbol{X}_b)$ and $\boldsymbol{E}_a = f_a(\boldsymbol{X}_a)$, XCom_a and XCom_b are feature completers hosted by client A and client B, respectively. For the fully missing feature scenario, the completed features $\tilde{\boldsymbol{X}}_a$ and $\tilde{\boldsymbol{X}}_b$ are entirely adopted. For the partially missing feature scenario, only the components of $\tilde{\boldsymbol{X}}_a$ and $\tilde{\boldsymbol{X}}_b$ for missing features are adopted, while the remaining components should be replaced by their corresponding existing features in the original positions. The completed features $\tilde{\boldsymbol{X}}_a$ and $\tilde{\boldsymbol{X}}_b$ are then input into their corresponding bottom models, f_a and f_b, to generate their embeddings $\tilde{\boldsymbol{E}}_a$ and



(a) Independent inference. (b) Ind. infer. with missing features (c) Collaborative inference.

Figure 5: Different inference modes of X-VFL for independent and collaborative inference, with or without partially missing features.

$\tilde{\boldsymbol{E}}_b$, respectively. Then, the classification decision loss for two clients is formulated as follows:

$$\begin{aligned} L_{\text{decision}} =& \ell(h(\boldsymbol{E}_a), y) + \ell(h(\boldsymbol{E}_b), y) + \ell(h(\frac{\boldsymbol{E}_a + \boldsymbol{E}_b}{2}), y) \\ &+ \ell(h(\frac{\boldsymbol{E}_a + \tilde{\boldsymbol{E}}_b}{2}), y) + \ell(h(\frac{\tilde{\boldsymbol{E}}_a + \boldsymbol{E}_b}{2}), y), \end{aligned} \tag{2}$$

where ℓ denotes the classification loss, e.g., cross-entropy (CE) loss. $\ell(h(\boldsymbol{E}_a), y)$ and $\ell(h(\boldsymbol{E}_b), y)$ ensure that the model maintains the ability to make accurate predictions based on features from a single client, thereby supporting independent inference. The term $\ell(h(\frac{\boldsymbol{E}_a+\boldsymbol{E}_b}{2}), y)$ reinforces the model's capacity to leverage the combined features, enhancing collaborative inference performance. The terms involving reconstructed embeddings, i.e., $\ell(h(\frac{\boldsymbol{E}_a+\tilde{\boldsymbol{E}}_b}{2}), y)$ and $\ell(h(\frac{\tilde{\boldsymbol{E}}_a+\boldsymbol{E}_b}{2}), y)$, ensure that the reconstructed embeddings ($\tilde{\boldsymbol{E}}_a$ or $\tilde{\boldsymbol{E}}_b$) effectively compensate for the missing information, thus enabling robust inference in the presence of missing features. For aligned data, all terms should be activated. For non-aligned data, we consider two cases: 1) Client A has full local features while client B has (partially) missing features. In this case, the loss terms $\ell(h(\boldsymbol{E}_a), y)$ and $\ell(h(\frac{\boldsymbol{E}_a+\tilde{\boldsymbol{E}}_b}{2}), y)$ are activated; 2) The roles are reversed; client B has full local features, and client A has (partially) missing features. The loss terms $\ell(h(\boldsymbol{E}_b), y)$ and $\ell(h(\frac{\tilde{\boldsymbol{E}}_a+\boldsymbol{E}_b}{2}), y)$ are activated.

By explicitly leveraging reconstructed features, XCom effectively addresses the challenges of training and inference with (partially) missing features in VFL.

Decision Subspace Alignment. DS-Align unifies the alignment between reconstructed and existing features for aligned data, as well as between local individual features and the joint averaged features aggregated from all clients, within the decision subspace. This reinforces the effectiveness of XCom and further enhances the performance of independent inference.
To enforce consistency between reconstructed features and their corresponding existing features, ensuring accurate and reliable feature completion, DS-Align introduces the following alignment loss within the decision subspace:

$$L_{\text{DSAlign}_1} = \ell_s(h(\tilde{\boldsymbol{E}}_a), h(\boldsymbol{E}_a)) + \ell_s(h(\tilde{\boldsymbol{E}}_b), h(\boldsymbol{E}_b)), \tag{3}$$

where ℓ_s represents a similarity loss function, e.g., mean square error (MSE), which ensures that inference based on reconstructed embeddings aligns closely with the existing embeddings.

To enhance the performance of independent inference, DS-Align aligns local individual features and joint averaged features from all clients within the decision subspace. This alignment enables more accurate independent inference and is formulated as:

$$L_{\text{DSAlign}_2} = \ell_s(h(\boldsymbol{E}_a), h(\frac{\boldsymbol{E}_a + \boldsymbol{E}_b}{2})) + \ell_s(h(\boldsymbol{E}_b), h(\frac{\boldsymbol{E}_a + \boldsymbol{E}_b}{2})). \tag{4}$$

This dual-component approach reinforces both feature completion and independent inference, thus enhancing the overall performance and scalability of X-VFL.

Finally, by integrating XCom and DS-Align, we formulate the overall loss function for our X-VFL framework as follows:

$$L = L_{\text{decision}} + \lambda_1 L_{\text{DSAlign}_1} + \lambda_2 L_{\text{DSAlign}_2}. \tag{5}$$

Here, L_{decision} (see Eq. (2)) denotes the classification decision loss for both aligned and non-aligned data samples, while L_{DSAlign_1} and L_{DSAlign_2} (see Eq. (3) and Eq. (4)) are the two loss components

introduced by our DS-Align, weighted by the hyperparameters λ_1 and λ_2, respectively. Due to page limitations, we provide the detailed multiple-client setting for X-VFL in Appendix A.

Inference modes of X-VFL. As illustrated in Fig. 5, the X-VFL framework supports multiple inference modes to highlight the flexibility and scalability of X-VFL:

1. **Independent inference without missing features** (Fig. 5a): Each client independently generates predictions using only its local features and model, without requiring cross-client communication during inference.
2. **Independent inference with missing features** (Fig. 5b): Each client leverages its XCom module to reconstruct missing features and produce predictions independently.
3. **Collaborative inference with/without missing features** (Fig. 5c): All clients contribute their feature embeddings (obtained either from the original data or from the reconstructed data via XCom) to the central server, enabling collaborative inference.

3 Theoretical Results

In this section, we provide theoretical convergence theorems for our X-VFL framework which is formulated as the following optimization problem (see Eq. (5) and Eq. (11)):

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) := L_{\text{decision}} + \lambda_1 L_{\text{DSAlign}_1} + \lambda_2 L_{\text{DSAlign}_2}, \tag{6}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ denotes the X-VFL model parameters, $\lambda_1 \in \mathbb{R}$ and $\lambda_2 \in \mathbb{R}$ are two hyperparameters. To present the theorems, we first introduce the necessary notation and assumptions.

3.1 Notation and assumptions

Let $\Delta_0 := L(\boldsymbol{\theta}^0) - L^*$, where $L^* := \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$. Let $\nabla L(\boldsymbol{\theta})$ denote the gradient of function L at point $\boldsymbol{\theta}$, $\tilde{\nabla} L(\boldsymbol{\theta})$ and $\tilde{\nabla}_b L(\boldsymbol{\theta})$ denote its stochastic gradient and minibatch stochastic gradients with size b. Let $[n]$ denote the set $\{1, 2, \cdots, n\}$ and $\|\cdot\|$ denote the Euclidean norm for a vector. Let $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ denote the inner product of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. We use $O(\cdot)$ to hide absolute constants.

In order to prove the convergence results, one usually needs the following standard smoothness assumption and stochastic gradient variance assumption (see e.g., Ghadimi et al. (2016); Fang et al. (2018); Li et al. (2021); Li & Richtárik (2020)).

Assumption 1 (Average smoothness)**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ is average β-smooth if*

$$\mathbb{E}[\|\tilde{\nabla} L(\boldsymbol{\theta}_1) - \tilde{\nabla} L(\boldsymbol{\theta}_2)\|^2] \leq \beta^2 \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|^2, \quad \forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d. \tag{7}$$

Assumption 2 (Bounded variance)**.** *The stochastic gradient $\tilde{\nabla} L(\boldsymbol{\theta})$ is unbaised and has bounded variance*

$$\mathbb{E}[\|\tilde{\nabla} L(\boldsymbol{\theta}) - \nabla L(\boldsymbol{\theta})\|^2] \leq \sigma^2, \quad \forall \boldsymbol{\theta} \in \mathbb{R}^d. \tag{8}$$

3.2 Convergence results

Now we present the convergence theorems for training X-VFL to find a suitable target parameter $\hat{\boldsymbol{\theta}}$. The convergence theorems can also indicate the communication complexity and computation complexity of different algorithms (such as standard stochastic gradient descent (SGD) and the optimal variance-reduced PAGE (Li et al., 2021)) for training X-VFL.

Theorem 1 (Convergence for SGD-type algorithms)**.** *Suppose that Assumptions 1 and 2 hold. For SGD-type algorithms, e.g., update $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \tilde{\nabla} L(\boldsymbol{\theta}^t)$, after T steps, we have $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq \frac{\Delta_0}{(\eta - \beta\eta^2/2)T} + \frac{\beta\eta^2\sigma^2}{2\eta - \beta\eta^2}$. Choose learning rate $\eta \leq \min\{\frac{2}{\beta}, \sqrt{\frac{2\Delta_0}{\beta\sigma^2 T}}\}$, we obtain*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq O\left(\frac{1}{\sqrt{T}}\right). \tag{9}$$

This means that after $T = O(\frac{1}{\epsilon^4})$ steps, SGD-type algorithms can find a suitable parameter $\hat{\boldsymbol{\theta}}$ for X-VFL *such that $\mathbb{E}\|\nabla L(\hat{\boldsymbol{\theta}})\|^2 \leq \epsilon^2$, where ϵ denotes the convergence error.*

Note that each update step requires one round of communication in X-VFL, so the total number of communication rounds is $T = O(\frac{1}{\epsilon^4})$. Also, each step incurs a computational cost corresponding to a stochastic gradient computation $\tilde{\nabla} L(\boldsymbol{\theta}^t)$.

We note that the large number of communication rounds is mainly due to the variance in stochastic gradients. Thus we also show that the optimal variance-reduced PAGE method (Li et al., 2021) can largely reduce the number of communication rounds, i.e., from $T = O(\frac{1}{\epsilon^4})$ to $T = O(\frac{1}{\epsilon^2})$ by a factor of $\frac{1}{\epsilon^2}$. A simplified PAGE update at step t is $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \boldsymbol{g}^t$, where $\boldsymbol{g}^t = \tilde{\nabla}_b L(\boldsymbol{\theta}^t)$ with probability p and $\boldsymbol{g}^t = \boldsymbol{g}^{t-1} + \tilde{\nabla}_{b'} L(\boldsymbol{\theta}^t) - \tilde{\nabla}_{b'} L(\boldsymbol{\theta}^{t-1})$ with probability $1-p$.

It uses minibatch SGD update $\tilde{\nabla}_b L(\boldsymbol{\theta}^t)$ with probability p, and otherwise reuses the previous gradient $\boldsymbol{g}^{t-1}$ with a small adjustment (lower computation cost if $b' \ll b$). This update reduces the variance in stochastic gradients during training and thus significantly decreases the total number of communication rounds, i.e., by a factor of $\frac{1}{\epsilon^2}$.

Theorem 2 (Convergence for PAGE-type algorithms)**.** *Suppose that Assumptions 1 and 2 hold. For PAGE-type algorithms, after T steps, we have $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq \frac{2\Delta_0}{\eta T} + \frac{\sigma^2}{pbT} + \frac{\sigma^2}{b}$. Choose learning rate $\eta \leq \frac{1}{2\beta}$, minibatch sizes $b = \frac{2\sigma^2}{\epsilon^2}$, $b' \leq \sqrt{b}$ and probability $p = \frac{b'}{b+b'}$, we obtain*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq O\left(\frac{1}{T}\right). \tag{10}$$

This means that after $T = O(\frac{1}{\epsilon^2})$ steps, PAGE-type algorithms can find a suitable parameter $\hat{\boldsymbol{\theta}}$ for X-VFL *such that $\mathbb{E}\|\nabla L(\hat{\boldsymbol{\theta}})\|^2 \leq \epsilon^2$, where ϵ denotes the convergence error.*

Similarly, each update step requires one round of communication in X-VFL, so the total number of communication rounds is $T = O(\frac{1}{\epsilon^2})$. Also, each step incurs a computational cost corresponding to a minibatch stochastic gradients computation with expected size $pb+(1-p)b' = \frac{b'b}{b+b'} + \frac{bb'}{b+b'} \leq 2b'$.

4 Experiments

In this section, we present a comprehensive empirical evaluation of the proposed X-VFL framework using 6 real-world datasets, including three image datasets: `CIFAR-10` (Krizhevsky et al., 2009), `TinyImageNet` restricted to five classes (Deng et al., 2009), and `UTKFace` (Zhang et al., 2017), and three tabular datasets: `MIMIC-III` (Johnson et al., 2016), `Bank` (Liang et al., 2016), and `Avazu` (Kang et al., 2022). We compare X-VFL against five baseline methods: Vanilla Standalone, Vanilla VFL, IAVFL (Ren et al., 2022), FedCVT (Kang et al., 2022), and LASER-VFL (Valdeira et al., 2025). Comprehensive implementation details regarding dataset specifications, model architectures, hyperparameter configurations, and baseline definitions are provided in Appendix C.

We benchmark our method across diverse practical scenarios, including varying feature missing rates (see Section 4.1), sample overlap ratios (see Appendix F), and imbalanced training data across clients (see Appendix G). Additionally, we provide an ablation study (see Section 4.2) and evaluate the impact of embedding noise on model robustness (see Section 4.3). We also analyze computation and communication complexity to evaluate system efficiency (see Appendix I).

4.1 Results under varying feature missing rates

We evaluate the impact of feature missingness by varying the missing rate from 0 to 1.0 across both image and tabular datasets. The experimental results on `CIFAR-10`, `UTKFace`, `MIMIC-III`, and `Avazu` under the two-client setting are summarized in Fig. 6. Supplementary experiments on additional datasets and multi-client settings are deferred to Appendix D and Appendix E.

As shown in Fig. 6, X-VFL consistently outperforms all methods across all datasets and feature missing rates, in both independent and collaborative modes. For example, on `UTKFace` at a severe missing rate of 0.9, prior methods experience an accuracy drop exceeding 30% in independent mode, whereas X-VFL maintains remarkable stability with a performance degradation of less than

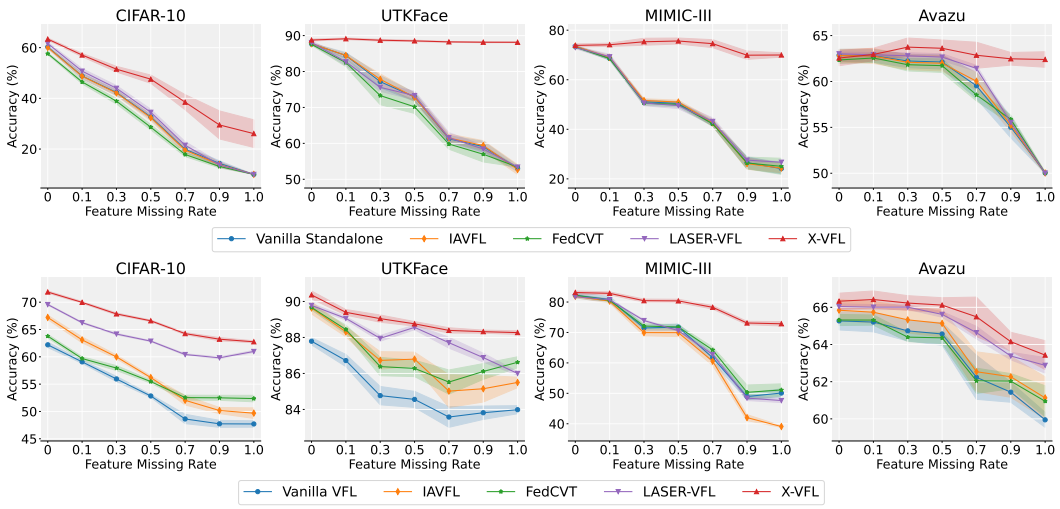


Figure 6: Performance comparison under varying feature missing rates across CIFAR-10, UTKFace, MIMIC-III, and Avazu datasets. The top and bottom rows correspond to the *independent inference* and *collaborative inference* modes, respectively.

0.5% relative to its complete-data baseline. Furthermore, X-VFL effectively bridges the performance discrepancy between independent and collaborative inference modes, as evidenced in Fig. 8 (see Appendix D). On UTKFace, X-VFL exhibits a negligible gap of less than 0.2% between independent and collaborative predictions, whereas prior baselines suffer a severe performance drop exceeding 20% when switching to independent prediction. An intriguing observation is that LASER-VFL achieves competitive gains in collaborative mode across several benchmarks by training a family of predictors via task-sampling, with the notable exception of MIMIC-III. A plausible explanation is that MIMIC-III consists entirely of discrete and categorical features, which constrains the model's ability to capture the intricate interactions inherent to categorical data. In contrast, X-VFL overcomes these structural limitations by effectively modeling cross-client correlations, thereby maintaining a substantial performance lead even on such challenging discrete datasets.

To verify the generalizability of X-VFL under diverse deployment constraints, we conduct extensive experiments on data overlap, inter-client imbalance, and system efficiency, with full details deferred to the appendix due to space limitations. Specifically, in Appendix F, we evaluate the impact of sample overlap under 20% and 80% alignment ratios. The results confirm that X-VFL consistently surpasses prior methods across all overlap ratios, achieving over 43% accuracy on CIFAR-10 at a low overlap ratio of 20% where baselines drop below 35%. In Appendix G, we examine an extreme 20% to 80% data imbalance setting between clients. While prior baselines suffer a drastic performance disparity exceeding 35% between data-rich and data-poor clients on MIMIC-III, X-VFL reduces this gap to only 8.42% and even enhances the accuracy of the data-poor client on CIFAR-10 by utilizing cross-client feature reconstruction. Finally, our theoretical and empirical efficiency analysis in Appendix I reveals that X-VFL scales linearly with the number of clients, incurring a modest training overhead of approximately $2\times$ in computation time and $1.35\times$ in communication volume.

4.2 Ablation Study

To evaluate the individual contributions of the XCom and DS-Align modules, we compare four configurations across CIFAR-10 and MIMIC-III: 1) Vanilla, using Vanilla standalone and Vanilla VFL for independent and collaborative inference, respectively; 2) W/ DS-Align, which employs the first three terms of the decision loss L_{decision}, alongside the local-to-joint alignment loss L_{DSAlign_2}; 3) W/ XCom, integrating the full decision loss L_{decision} with reconstructed embeddings $\tilde{\boldsymbol{E}}$ but omits the alignment losses. 4) Full (X-VFL), representing our complete framework as defined in Eq. (5).

As illustrated in Fig. 7 and Fig. 15 (see Appendix H), X-VFL consistently achieves the highest accuracy, demonstrating a clear synergy between feature completion and subspace alignment.

The XCom module serves as the primary driver for robustness, effectively preventing the catastrophic accuracy collapse observed in the Vanilla baseline as the missing rate increases. Although the marginal gain from DS-Align is smaller, its role in aligning local and joint representations provides a consistent refinement in accuracy across all scenarios. These findings confirm that while XCom restores missing information, DS-Align ensures latent space consistency, making the integration of both modules essential for robust VFL under extreme data incompleteness.

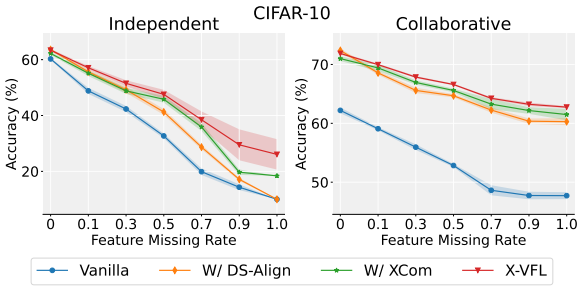


Figure 7: Ablation study of XCom and DS-Align.

4.3 Impact of Embedding Noise on X-VFL Performance

To mitigate the risk of raw data leakage via shared embeddings, vanilla VFL protocols typically inject noise into local embeddings before transmission. We therefore evaluate the performance of X-VFL under these privacy-preserving constraints. Our primary objective is to assess the accuracy impact of the additional noise introduced by the XCom module in X-VFL relative to vanilla VFL. Specifically, we apply Gaussian noise with scales $\sigma \in \{0.01, 0.1\}$ to L2-norm-clipped embeddings, where $\sigma = 0.1$ is recognized as a benchmark for providing substantial privacy protection (Xu et al., 2024). Table 1 summarizes the test accuracy for both frameworks across various feature missing rates and noise scales on the CIFAR-10 dataset.

As shown, while noise injection inevitably leads to a decline in accuracy for both frameworks, the performance of X-VFL remains significantly more stable as data incompleteness intensifies. At a missing rate of 1.0, Vanilla VFL suffers a catastrophic 18.80% reduction in accuracy under a noise scale of 0.1, whereas X-VFL exhibits a controlled decline of only 5.28%. Notably, the absolute accuracy of X-VFL under high-noise conditions consistently exceeds that of Vanilla VFL in a noise-free setting across nearly all scenarios. These results confirm that X-VFL effectively maintains high utility, successfully mitigating the joint impact of extreme data incompleteness and noise perturbations.

Table 1: Test accuracy (%) on CIFAR-10 across varying feature missing rates and Gaussian noise scales (σ).

Missing Rate	Method	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.1$
0	Vanilla VFL	62.15	61.18 (↓ 0.97)	60.24 (↓ 1.91)
	X-VFL	**71.86**	**70.52** (↓ 1.34)	**69.09** (↓ 2.77)
0.3	Vanilla VFL	55.97	54.13 (↓ 1.84)	53.77 (↓ 2.20)
	X-VFL	**67.85**	**65.66** (↓ 2.19)	**63.98** (↓ 3.87)
0.5	Vanilla VFL	52.90	48.94 (↓ 3.96)	48.99 (↓ 3.91)
	X-VFL	**66.59**	**64.55** (↓ 2.04)	**62.42** (↓ 4.17)
0.7	Vanilla VFL	48.43	40.26 (↓ 8.17)	41.00 (↓ 7.43)
	X-VFL	**64.25**	**62.51** (↓ 1.74)	**59.62** (↓ 4.63)
1.0	Vanilla VFL	47.76	29.85 (↓ 17.91)	28.96 (↓ 18.80)
	X-VFL	**62.75**	**61.02** (↓ 1.73)	**57.47** (↓ 5.28)

5 Conclusion

In this paper, we propose X-VFL, a novel VFL framework that addresses key challenges of Vanilla VFLs by effectively handling datasets with partially missing features and enabling locally independent inference at each client. In particular, X-VFL introduces two key modules: Cross Completion (XCom) and Decision Subspace Alignment (DS-Align). XCom is designed to complete missing features for non-aligned data samples by exploiting cross-client information, thereby effectively increasing the volume of data available for training and inference. DS-Align aligns local features with completed and global features across all clients within the decision subspace, enabling each client to perform locally independent inference. In addition, theoretical convergence theorems are established for different algorithms used in training X-VFL. Extensive experiments validate its superior performance over existing state-of-the-art methods.

AI USE STATEMENT

Parts of the manuscript were refined with the assistance of a large language model to enhance clarity. All content was subsequently reviewed and verified by the authors.

REFERENCES

Timothy Castiglia, Shiqiang Wang, and Stacy Patterson. Self-supervised vertical federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges, in Conjunction with NeurIPS 2022*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE Computer Society, 2009.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1):267–305, 2016.

Yuanqin He, Yan Kang, Xinyuan Zhao, Jiahuan Luo, Lixin Fan, Yuxing Han, and Qiang Yang. A hybrid self-supervised learning framework for vertical federated learning. *IEEE Transactions on Big Data*, 11(5):2210–2223, 2024.

Kihun Hong, Sejun Park, and Ganguk Hwang. Deep latent variable model based vertical federated learning with flexible alignment and labeling scenarios. In *International Conference on Learning Representations*, volume 2026, pp. 152525–152549, 2026.

Chung-ju Huang, Leye Wang, and Xiao Han. Vertical federated knowledge transfer via representation distillation for healthcare collaboration networks. In *Proceedings of the ACM Web Conference 2023*, pp. 4188–4199, 2023.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Madhura Joshi, Ankit Pal, and Malaikannan Sankarasubbu. Federated learning for healthcare domain - pipeline, applications and challenges. *ACM Transactions on Computing for Healthcare*, 3 (4), 2022.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Yan Kang, Yang Liu, and Xinle Liang. FedCVT: Semi-supervised vertical federated learning with cross-view training. *ACM Transactions on Intelligent Systems and Technology*, 13(4):1–16, 2022.

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

Wenjie Li, Qiaolin Xia, Hao Cheng, Kouyin Xue, and Shu-Tao Xia. Vertical semi-federated learning for efficient online advertising. *arXiv preprint arXiv:2209.15635*, 2022a.

Wenjie Li, Qiaolin Xia, Junfeng Deng, Hao Cheng, Jiangming Liu, Kouying Xue, Yong Cheng, and Shu-Tao Xia. VFed-SSD: Towards practical vertical federated advertising. *arXiv preprint arXiv:2205.15987*, 2022b.

Zhize Li and Peter Richtárik. A unified analysis of stochastic gradient methods for nonconvex federated optimization. *arXiv preprint arXiv:2006.07013*, 2020.

Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, pp. 6286–6295. PMLR, 2021.

Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research*, 252(2):561–572, 2016.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Zhenghang Ren, Liu Yang, and Kai Chen. Improving availability of vertical federated learning: Relaxing inference on non-overlapping data. *ACM Transactions on Intelligent Systems and Technology*, 13(4):1–20, 2022.

Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8485–8493, 2022.

Pedro Valdeira, Shiqiang Wang, and Yuejie Chi. Vertical federated learning with missing features during training and inference. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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

Zhaomin Wu, Zhen Qin, Junyi Hou, Haodong Zhao, Qinbin Li, Bingsheng He, and Lixin Fan. Vertical federated learning in practice: The good, the bad, and the ugly. *arXiv preprint arXiv:2502.08160*, 2025.

Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*, 2023.

Xiangrui Xu, Wei Wang, Zheng Chen, Bin Wang, Chao Li, Li Duan, Zhen Han, and Yufei Han. Finding the piste: Towards understanding privacy leaks in vertical federated learning systems. *IEEE Transactions on Dependable and Secure Computing*, 22(2):1537–1550, 2024.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

A MULTIPLE-CLIENT SETTING FOR X-VFL

In this appendix, we describe our X-VFL framework for the general case with multiple k clients. This generalization enhances the practicality of X-VFL and provides a more scalable and effective solution for broader applications. Similar to Section 2.1, let $f_1, \ldots, f_k$ denote the local models for the k clients. Then $\boldsymbol{E}_i = f_i(\boldsymbol{x_i})$ denotes the feature embeddings/representations for client i, $i \in [k]$.

For the general case of k clients, the overall loss function can be formulated as the same one as in Eq. (5):

$$L = L_{\text{decision}} + \lambda_1 L_{\text{DSAlign}_1} + \lambda_2 L_{\text{DSAlign}_2}. \tag{11}$$

Compared with two-client setting, now the classification decision loss for k clients is formulated as follows:

$$L_{\text{decision}} = \sum_{i\in\mathcal{M}} \ell(h(\boldsymbol{E}_i), y) + \ell(h(\frac{\boldsymbol{E}'}{m}), y) + \ell(h(\frac{\tilde{\boldsymbol{E}}' + \boldsymbol{E}'}{k}), y) + \sum_{i=1}^{k} \ell(h(\frac{\tilde{\boldsymbol{E}}_i + \boldsymbol{E}_{-i}}{k}), y), \tag{12}$$

where $\boldsymbol{E}' = \sum_{j\in\mathcal{M}} \boldsymbol{E}_j$, $\tilde{\boldsymbol{E}}' = \sum_{i\in\{1,\dots,k\}\setminus\mathcal{M}} \tilde{\boldsymbol{E}}'_i$, $\mathcal{M} \subseteq \{1,\dots,k\}$ denotes the subset of clients that hold full local features (i.e., clients without missing features), where $|\mathcal{M}| = m$ for some $m \le k$, $\boldsymbol{E}_{-i} = \sum_{j\in\{1,\dots,k\}\setminus\{i\}} \boldsymbol{E}_j$, and

$$\tilde{\boldsymbol{E}}_i = f_i(\tilde{\boldsymbol{X}}_i), \quad \tilde{\boldsymbol{X}}_i = \mathsf{XCom}_i(\frac{\boldsymbol{E}_{-i}}{k-1}), \quad 1 \le i \le k, \tag{13}$$

$$\tilde{\boldsymbol{E}}'_i = f_i(\tilde{\boldsymbol{X}}'_i), \quad \tilde{\boldsymbol{X}}'_i = \mathsf{XCom}_i(\frac{\boldsymbol{E}'}{m}), \quad 1 \le i \le k. \tag{14}$$

For aligned data, the following terms in Eq. (12) are activated: $\sum_{i\in\mathcal{M}} \ell(h(\boldsymbol{E}_i), y)$, $\ell(h(\frac{\boldsymbol{E}'}{m}), y)$, and $\sum_{i=1}^{k} \ell(h(\frac{\tilde{\boldsymbol{E}}_i + \boldsymbol{E}_{-i}}{k}), y)$. The term $\ell(h(\frac{\tilde{\boldsymbol{E}}' + \boldsymbol{E}'}{k}), y)$ is not activated, as $\tilde{\boldsymbol{E}}'$ is empty when $\mathcal{M}$ includes all clients (i.e., $k = m$). For non-aligned data, the actived terms are $\sum_{i\in\mathcal{M}} \ell(h(\boldsymbol{E}_i), y)$, $\ell(h(\frac{\boldsymbol{E}'}{m}), y)$, and $\ell(h(\frac{\tilde{\boldsymbol{E}}' + \boldsymbol{E}'}{k}), y)$.

The last two loss components introduced by our DS-Align module in the general case of k clients are formulated as follows:

$$L_{\text{DSAlign}_1} = \sum_{i=1}^{k} \ell(h(\tilde{\boldsymbol{E}}_i), h(\boldsymbol{E}_i)), \tag{15}$$

and

$$L_{\text{DSAlign}_2} = \sum_{i=1}^{k} \ell(h(\boldsymbol{E}_i), h(\frac{\sum_{i=1}^{k} \boldsymbol{E}_i}{k})). \tag{16}$$

B Missing Proofs

In this appendix, we provide the detailed proofs for our convergence Theorems 1 and 2 in Appendix B.1 and Appendix B.2, respectively.

B.1 Proof of Theorem 1

According to SGD-type update $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta\tilde{\nabla}L(\boldsymbol{\theta}^t)$ and the smoothness Assumption 1, we have

$$\mathbb{E}_t[L(\boldsymbol{\theta}^{t+1})] \le \mathbb{E}_t[L(\boldsymbol{\theta}^t) + \langle\nabla L(\boldsymbol{\theta}^t), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\rangle + \frac{\beta}{2}\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\|^2] \tag{17}$$

$$= \mathbb{E}_t[L(\boldsymbol{\theta}^t) - \eta\langle\nabla L(\boldsymbol{\theta}^t), \tilde{\nabla}L(\boldsymbol{\theta}^t)\rangle + \frac{\beta\eta^2}{2}\|\tilde{\nabla}L(\boldsymbol{\theta}^t)\|^2] \tag{18}$$

$$\le L(\boldsymbol{\theta}^t) - (\eta - \frac{\beta\eta^2}{2})\|\nabla L(\boldsymbol{\theta}^t)\|^2 + \frac{\beta\eta^2\sigma^2}{2}, \tag{19}$$

where $\mathbb{E}_t$ takes the expectation conditioned on all history before step t, and Eq. (19) uses Assumption 2.

Summing up Eq. (19) from $t = 0$ to $T - 1$ and rearranging terms, we get

$$(\eta - \frac{\beta\eta^2}{2}) \sum_{t=0}^{T-1} \mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq L(\boldsymbol{\theta}^0) - \mathbb{E}[L(\boldsymbol{\theta}^t)] + \frac{\beta\eta^2\sigma^2 T}{2} \tag{20}$$

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq \frac{\Delta_0}{(\eta - \beta\eta^2/2)T} + \frac{\beta\eta^2\sigma^2}{2\eta - \beta\eta^2}, \tag{21}$$

where $\Delta_0 := L(\boldsymbol{\theta}^0) - L^*$. Note that $\frac{\beta\eta^2\sigma^2}{2\eta-\beta\eta^2} \leq \frac{\Delta_0}{(\eta-\beta\eta^2/2)T}$ if $\eta \leq \sqrt{\frac{2\Delta_0}{\beta\sigma^2 T}}$. Thus it is not hard to obtain, by choosing the learning rate $\eta \leq \min\{\frac{2}{\beta}, \sqrt{\frac{2\Delta_0}{\beta\sigma^2 T}}\}$,

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq O\left(\sqrt{\frac{\beta\Delta_0\sigma^2}{T}}\right) = O\left(\frac{1}{\sqrt{T}}\right) = \epsilon^2. \tag{22}$$

This means that after $T = O(\frac{1}{\epsilon^4})$ steps, SGD-type algorithms can find a suitable parameter $\hat{\boldsymbol{\theta}}$ for X-VFL such that $\mathbb{E}\|\nabla L(\hat{\boldsymbol{\theta}})\|^2 \leq \epsilon^2$, where ϵ denotes the convergence error. □

B.2 Proof of Theorem 2

The variance in stochastic gradients leads to a large number of communication rounds (i.e., update steps T). We now show that the optimal variance-reduced PAGE-type method (Li et al., 2021) effectively reduces the variance and thus largely decreases the total number of communication rounds (Theorem 2).

According to simplified PAGE update step in Section 3.2 and Lemma 4 of PAGE (Li et al., 2021), we have the following variance reduction lemma:

Lemma 1. *Suppose that Assumptions 1 and 2 hold. For the probabilistic gradient estimator $\boldsymbol{g}^t$ defined in Section 3.2, we have*

$$\begin{aligned}\mathbb{E}_t[\|\boldsymbol{g}^t - \nabla L(\boldsymbol{\theta}^t)\|^2] \leq (1-p)\|\boldsymbol{g}^{t-1} - \nabla L(\boldsymbol{\theta}^{t-1})\|^2 \\ + \frac{(1-p)\beta^2}{b'}\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t-1}\|^2 + \frac{p\sigma^2}{b}.\end{aligned} \tag{23}$$

This lemma indicates that the variance is roughly reduced by a factor $1 - p$ after each update step. Then according to the descent Lemma 2 of PAGE (Li et al., 2021), we have

$$\begin{aligned}L(\boldsymbol{\theta}^{t+1}) \leq L(\boldsymbol{\theta}^t) - \frac{\eta}{2}\|\nabla L(\boldsymbol{\theta}^t)\|^2 - (\frac{1}{2\eta} - \frac{\beta}{2})\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\|^2 \\ + \frac{\eta}{2}\|\boldsymbol{g}^t - \nabla L(\boldsymbol{\theta}^t)\|^2.\end{aligned} \tag{24}$$

Taking the expectation and combining Eq. (24) with multiple Eq. (23), we obtain

$$\begin{aligned}&\mathbb{E}[L(\boldsymbol{\theta}^{t+1}) - L^* + \frac{\eta}{2p}\|\boldsymbol{g}^{t+1} - \nabla L(\boldsymbol{\theta}^{t+1})\|^2] \\ &\leq \mathbb{E}\Big[L(\boldsymbol{\theta}^t) - L^* - \frac{\eta}{2}\|\nabla L(\boldsymbol{\theta}^t)\|^2 - (\frac{1}{2\eta} - \frac{\beta}{2})\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\|^2 \\ &\quad + \frac{\eta}{2}\|\boldsymbol{g}^t - \nabla L(\boldsymbol{\theta}^t)\|^2 + \frac{\eta}{2p}\Big((1-p)\|\boldsymbol{g}^t - \nabla L(\boldsymbol{\theta}^t)\|^2 \\ &\quad + \frac{(1-p)\beta^2}{b'}\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\|^2 + \frac{p\sigma^2}{b}\Big)\Big] \qquad (25) \\ &= \mathbb{E}\Big[L(\boldsymbol{\theta}^t) - L^* + \frac{\eta}{2p}\|\boldsymbol{g}^t - \nabla L(\boldsymbol{\theta}^t)\|^2 - \frac{\eta}{2}\|\nabla L(\boldsymbol{\theta}^t)\|^2 + \frac{\eta\sigma^2}{2b} \\ &\quad - (\frac{1}{2\eta} - \frac{\beta}{2} - \frac{(1-p)\eta\beta^2}{2pb'})\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\|^2\Big] \qquad (26)\end{aligned}$$

$$\leq \mathbb{E}\Big[L(\boldsymbol{\theta}^t) - L^* + \frac{\eta}{2p}\|\boldsymbol{g}^t - \nabla L(\boldsymbol{\theta}^t)\|^2 - \frac{\eta}{2}\|\nabla L(\boldsymbol{\theta}^t)\|^2 + \frac{\eta\sigma^2}{2b}\Big], \tag{27}$$

where Eq. (27) holds due to $\frac{1}{2\eta} - \frac{\beta}{2} - \frac{(1-p)\eta\beta^2}{2pb'} \geq 0$ by choosing learning rate $\eta \leq \frac{1}{\beta(1+\sqrt{(1-p)/(pb')})}$.

Summing up Eq. (27) from $t = 0$ to $T-1$ and rearranging terms, we get

$$\frac{\eta}{2}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq \mathbb{E}\Big[L(\boldsymbol{\theta}^0) - L^* + \frac{\eta}{2p}\|\boldsymbol{g}^0 - \nabla L(\boldsymbol{\theta}^0)\|^2 + \frac{\eta\sigma^2 T}{2b}\Big] \tag{28}$$

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq \frac{2\Delta_0}{\eta T} + \frac{\sigma^2}{pbT} + \frac{\sigma^2}{b}, \tag{29}$$

where Eq. (29) uses $\Delta_0 := L(\boldsymbol{\theta}^0) - L^*$ and the bounded variance Assumption 2 for the initial gradient estimator $g^0 = \tilde{\nabla}_b L(\boldsymbol{\theta}^0)$.

Then for any convergence error ϵ, by choosing learning rate $\eta \leq \frac{1}{2\beta}$, minibatch sizes $b = \frac{2\sigma^2}{\epsilon^2}, b' \leq \sqrt{b}$, and probability $p = \frac{b'}{b+b'}$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla L(\boldsymbol{\theta}^t)\|^2 \leq O\left(\frac{1}{T}\right) = \epsilon^2. \tag{30}$$

This means that after $T = O(\frac{1}{\epsilon^2})$ steps, PAGE-type algorithms can find a suitable parameter $\hat{\boldsymbol{\theta}}$ for X-VFL such that $\mathbb{E}\|\nabla L(\hat{\boldsymbol{\theta}})\|^2 \leq \epsilon^2$, where ϵ denotes the convergence error. Compared with the SGD-type algorithms in Theorem 1, the variance-reduced PAGE-type algorithms in Theorem 2 significantly decrease the total number of communication rounds, i.e., from $T = O(\frac{1}{\epsilon^4})$ to $T = O(\frac{1}{\epsilon^2})$ by a factor of $\frac{1}{\epsilon^2}$. □

C Experimental Details

Table 2: Hyperparameter ranges for λ_1 and λ_2

Dataset	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$
CIFAR-10	$\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$	$\{1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}\}$
TinyImageNet	$\{1, 2, 5, 10, 20\}$	$\{5\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}, 5\times10^{-3}\}$
UTKFace	$\{1, 2, 5, 10, 20\}$	$\{5\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}, 5\times10^{-3}\}$
MIMIC-III	$\{5\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}, 5\times10^{-3}, 1\times10^{-2}\}$	$\{0.02, 0.05, 0.1\}$
Bank	$\{1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}\}$	$\{5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}\}$
Avazu	$\{1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}\}$	$\{1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}\}$

We now provide more details of the experimental setup in this appendix. We conduct the experiments on real-world datasets, including three image datasets: CIFAR-10 (Krizhevsky et al., 2009), TinyImageNet (restricted to five classes) (Deng et al., 2009), and UTKFace (Zhang et al., 2017), as well as three tabular datasets: MIMIC-III (Johnson et al., 2016), Bank (Liang et al., 2016), and Avazu (Kang et al., 2022). For the image datasets, features are vertically partitioned into disjoint subsets and assigned to different clients. For the tabular datasets, continuous features are normalized to the range $[0, 1]$, while categorical features are transformed into one-hot encodings (Bank) or embedding-based representations (Avazu). These preprocessed features are then partitioned into disjoint subsets and distributed to the clients. The server retains the class labels of the datasets.

To simulate real-world scenarios where clients have partially missing features, we mask a proportion of features in the non-aligned training and test samples. Various feature missing rates $R_{\text{miss}} = \{0, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ are applied to thoroughly assess the impact of missing features.

For the image datasets (CIFAR-10, TinyImageNet, and UTKFace), the client's local (bottom) model is implemented using a residual network with varying depths, characterized by different numbers of residual blocks. The corresponding server (top) model is a six-layer fully connected neural

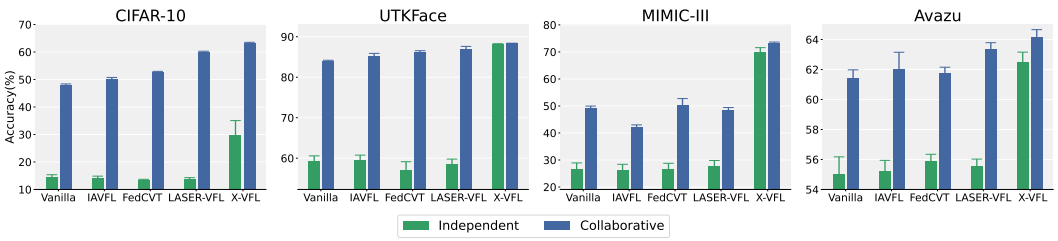


Figure 8: Performance comparison between independent and collaborative inference modes under feature missing rate $R_{\text{miss}} = 0.9$ on the `CIFAR-10`, `UTKFace`, `MIMIC-III`, and `Avazu` datasets.

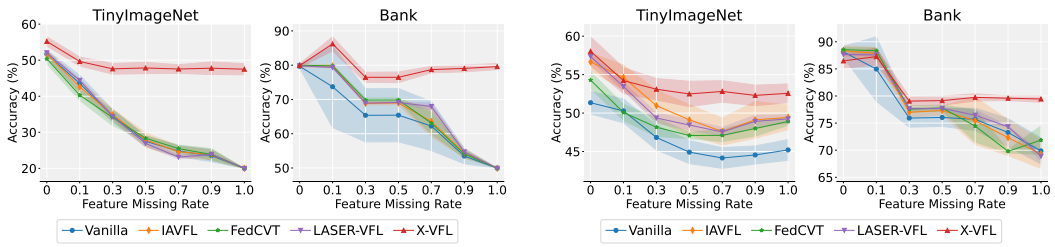
network (FCNN). For the tabular datasets (`MIMIC-III`, `Bank`, and `Avazu`), both the client and server models are implemented as three-layer FCNNs. The input dimensions of the server models were set to 2048 for image datasets and 512 for tabular datasets to accommodate differences in feature complexity and representation requirements. The batch size is set to 50. The hyperparameters λ_1 and λ_2 defined in the overall loss function (see Eq. (5)) are selected from the ranges listed in Table 2.

Vanilla Standalone refers to an independently trained model in which each client uses only its own local features for training and inference. It serves as a baseline for evaluating the capability of independent inference. Vanilla VFL enables collaborative training by concatenating local feature representations from all clients at a central server. During inference, predictions are generated based on these concatenated representations. This approach serves as a baseline for evaluating the capability of collaborative inference. For other baselines, although previous studies typically either cannot handle missing features or do not support locally independent inference, we identify two related works, IAVFL (Ren et al., 2022), FedCVT (Kang et al., 2022), and LASER-VFL (Valdeira et al., 2025) that are relevant for comparison, as discussed in Section 1.1. Concretely, we compare our X-VFL with these four prior methods for both independent Inference and collaborative inference modes, using test classification accuracy as the performance metric. All results are averaged over five random seeds to ensure robustness.

D Extended Evaluation under Varying Feature Missing Rates

In this section, we provide supplementary experimental results to further evaluate the robustness and generalizability of X-VFL under varying feature missing rates. We first investigate the performance discrepancy between independent and collaborative inference modes under an extreme missing rate of $R_{\text{miss}} = 0.9$, as depicted in Fig. 8. A decisive advantage of X-VFL lies in its capability to bridge the accuracy gap between these two operational regimes. On the `UTKFace` dataset, for instance, X-VFL exhibits a negligible performance drop of less than 0.2%, maintaining an accuracy of 88.15% in independent inference compared to 88.32% in collaborative inference. In sharp contrast, prior baseline methods suffer a severe degradation exceeding 20% when transitioning from collaborative to independent prediction. A consistent pattern is observed across all evaluated benchmarks, demonstrating that X-VFL ensures reliable predictive stability even when individual clients must operate in isolation under severe data incompleteness.

Furthermore, we extend our evaluation to the `TinyImageNet` and `Bank` datasets across the full spectrum of feature missing rates, with the corresponding results and mode gaps presented in Fig. 9 and Fig. 10. Across both image and tabular domains, X-VFL consistently outperforms competing approaches by a substantial margin at all missing rates. This superiority is especially pronounced in the independent inference mode, where baseline methods degrade rapidly as the missing rate increases. These empirical observations corroborate our primary conclusions in the main text, verifying that the proposed feature completion and subspace alignment mechanisms generalize effectively to diverse visual categories and real-world financial records.



(a) Independent inference mode

(b) Collaborative inference mode

Figure 9: Performance comparison under varying feature missing rates on the `TinyImagenet` and `Bank` datasets.

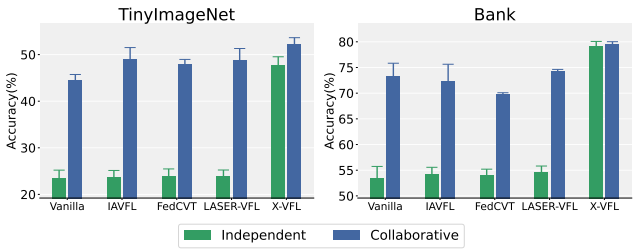


Figure 10: Performance difference between independent and collaborative inference modes under feature missing rate $R_{\text{miss}} = 0.9$ on the `TinyImagenet` and `Bank` datasets.

E Experimental Results for Multiple Clients

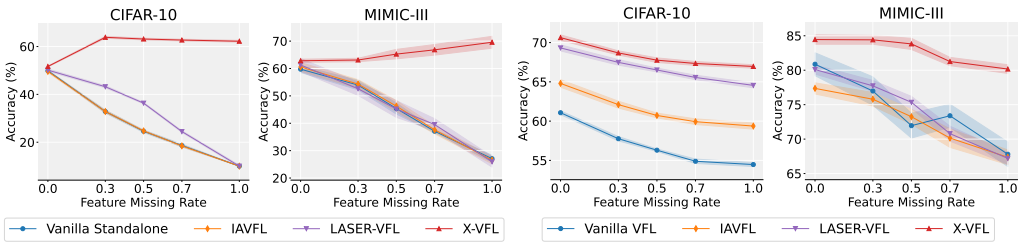
In this section, we provide the experimental results for the general k-client setting. We use `CIFAR-10` (Krizhevsky et al., 2009) as the image dataset and `MIMIC-III` (Johnson et al., 2016) as the tabular dataset. In particular, we conduct experiments with $k = 4$ clients. Regarding prior methods, although FedCVT (Kang et al., 2022) addresses non-aligned data in two-party VFL, we do not include it as a baseline in the k-client setting because generalizing its cross-view pseudo-labeling logic to multiple parties would lead to a combinatorial explosion of auxiliary classifiers and is not provided in its original formulation. Therefore, we include four prior methods (Vanilla Standalone, Vanilla VFL, IAVFL, LASER-VFL) in the k-client setting experiments, consistent with those used in the two-client setting, excluding only FedCVT. The implementation details remain the same as in the two-client case (see Appendix C), except that the batch size is set to 100.

Fig. 11 presents the results under varying feature missing rates, $R_{\text{miss}} = \{0, 0.3, 0.5, 0.7, 1.0\}$, for both independent and collaborative inference modes. A comparison of performance between these two modes at a missing rate $R_{\text{miss}} = 0.7$ is shown in Fig. 12. The results follow the same trend observed in the two-client case: our X-VFL significantly outperforms prior methods, particularly in the independent inference mode. Moreover, Fig. 12 highlights the advantage of X-VFL in effectively narrowing the accuracy gap between independent and collaborative inference modes.

F Results under varying overlap ratios of aligned data samples

To evaluate the effect of varying overlap ratios of aligned data samples, we test these methods with 20% and 80% aligned samples, under a fixed feature missing rate of $R_{\text{miss}} = 0.5$. The performance results on `CIFAR-10` and `MIMIC-III` are presented in Fig. 13a and Fig. 13b, respectively.

As illustrated, X-VFL consistently outperforms prior methods across all overlap ratios in both independent and collaborative inference modes. For instance, on the `CIFAR-10` dataset (Fig. 13a), X-VFL achieves over 43% accuracy in both independent and collaborative inference mode at a 20% overlap ratio, whereas the three prior methods all fall below 35%. Also, a similar trend is observed



(a) Independent inference mode (b) Collaborative inference mode

Figure 11: Performance comparison under varying feature missing rates on the CIFAR-10 and MIMIC-III datasets in the multiple clients setting.

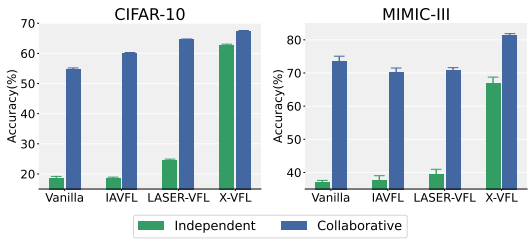


Figure 12: Performance comparison between independent and collaborative inference modes under feature missing rate of 0.7 on the CIFAR-10 and MIMIC-III datasets in the multiple clients setting.

on the MIMIC-III dataset (Fig. 13b), further demonstrating X-VFL's robustness and superior performance, particularly under low overlap ratios.

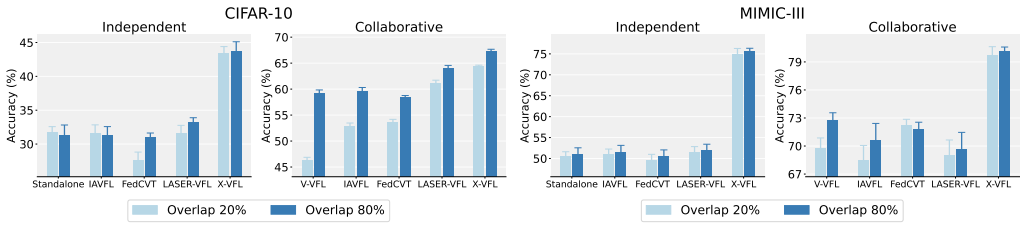
G Results under imbalanced training data across clients

To evaluate the impact of data imbalance between clients for these methods, we consider a 20% data imbalance setting, where client A and client B hold 80% and 20% of the total data, respectively, including both aligned and non-aligned samples. This setup enables a thorough evaluation of the impact of imbalanced training data on classification performance. The performance gap between the data-rich and data-poor clients also offers an interesting point of analysis. Experimental results for independent inference under a feature missing rate of $R_{\text{miss}} = 0.5$ are presented in Fig. 14.

As demonstrated in Fig. 14, prior methods exhibit substantial performance gaps between the two clients, particularly on the MIMIC-III dataset, where the data-rich client A consistently outperforms the data-poor client B by 36.06%, 35.85%, and 36.29%. These gaps highlight the inability of prior methods to effectively handle data imbalance, leading to significantly degraded performance for the data-poor client B. In contrast, our X-VFL substantially reduces this gap to just 8.42%, demonstrating its superior ability to balance the performance between data-rich and data-poor clients, by completing the missing features through the XCom module. A similar yet more compelling trend is observed on the CIFAR-10 dataset. While prior methods fail to adequately support the data-poor client B, X-VFL not only narrows the performance gap but also slightly improves the accuracy of the data-poor client B. This unexpected gain can be attributed to the design of the XCom module, which leverages the richer data from client A to enhance the feature representations for the data-poor client B by reconstructing its missing features, thus improving classification performance despite the severe data imbalance.

H Supplementary Ablation Study on MIMIC-III

In this section, we present the ablation study on the tabular dataset MIMIC-III across feature missing rates from 0.0 to 1.0, as illustrated in Fig. 15. The empirical curves highlight the indis-



(a) CIFAR-10 dataset (b) MIMIC-III dataset

Figure 13: Performance comparison under varying overlap ratios on the CIFAR-10 and MIMIC-III datasets.

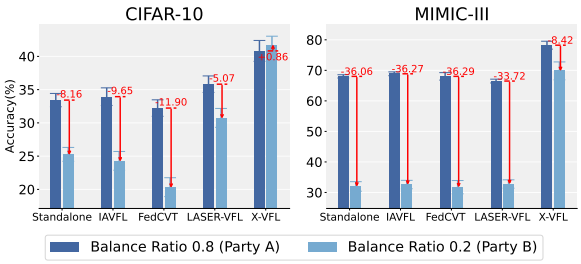


Figure 14: Performance comparison under data imbalance on the CIFAR-10 and MIMIC-III datasets.

pensable role of the XCom module when processing structured clinical features. In independent inference, configurations without feature completion, specifically Vanilla and W/ DS-Align, experience a catastrophic collapse, with accuracy plunging from roughly 73% to approximately 25% as feature absence reaches 1.0. In stark contrast, both W/ XCom and the full X-VFL framework maintain remarkable stability around 70% to 75% across all missing rates, proving that XCom effectively recovers critical diagnostic signals from cross-client representations.

A consistent trend is observed in collaborative inference, where Vanilla VFL and W/ DS-Align degrade sharply to approximately 50% accuracy under severe missingness, while XCom-enabled variants consistently retain performance above 72%. Moreover, the full X-VFL framework maintains a steady lead over the W/ XCom variant across all missing rates, confirming that DS-Align further optimizes representation consistency between local and joint decision spaces. These findings on MIMIC-III corroborate our conclusions on CIFAR-10, validating the necessity of combining feature-level completion with decision subspace alignment across both visual and tabular modalities.

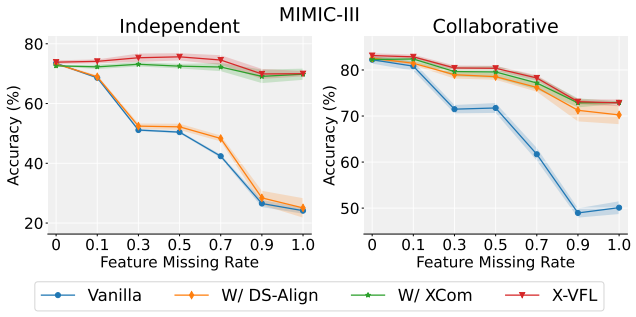


Figure 15: Ablation study of XCom and DS-Align modules on the MIMIC-III dataset.

Table 3: Efficiency comparison between Vanilla VFL and X-VFL. The values in parentheses denote the overhead ratio relative to the Vanilla VFL baseline.

Dataset	Method	Time (ms/batch)	Comm. Vol. [1] (MB/batch)
CIFAR-10	Vanilla VFL	11.41	1.56
	X-VFL	25.06 (2.19×)	2.14 (1.37×)
MIMIC-III	Vanilla VFL	3.65	0.39
	X-VFL	6.85 (1.87×)	0.53 (1.35×)

Comm. Vol. [1]: Communication Volume

I Computation and Communication Analysis

In this section, we analyze the efficiency of X-VFL from both theoretical and empirical perspectives to address the overhead concerns regarding the integration of XCom and DS-Align modules.

Theoretical Complexity Analysis. Compared to Vanilla VFL, X-VFL requires an additional exchange of aggregated embeddings to facilitate feature completion and alignment. Theoretically, while the framework involves bidirectional broadcasting, the communication complexity per client remains $\mathcal{O}(d)$, where d denotes the embedding dimension. This is because each client only needs to receive a single aggregated average embedding from the other $k-1$ participants rather than individual embeddings from every peer (see Appendix A for multiple-client setting). Consequently, the total communication volume scales linearly with the number of clients k, which ensures the scalability of the framework for large-scale federated systems. Computationally, the additional overhead arises from the extra forward passes through the XCom module and local backbones for feature completion, alongside the alignment of decision subspaces. Nevertheless, this overhead is mitigated as XCom is implemented as a lightweight CNN and DS-Align operates exclusively on low-dimensional decision subspaces.

Empirical Evaluation. To quantify these costs, we evaluate the computational latency and communication overhead per batch during the training phase on the CIFAR-10 and MIMIC-III datasets. We benchmark X-VFL against the Vanilla VFL baseline as detailed in Table 3. The empirical results show that X-VFL incurs a manageable increase in both training time and communication volume. Regarding computational latency, X-VFL exhibits an overhead ratio between 1.87× and 2.19× across the two datasets. This increase is primarily attributed to the extra forward passes through the bottom models and the XCom module to handle feature completion and subspace alignment. For communication volume, the overhead is notably lower, ranging from 1.35× to 1.37× that of the baseline. This demonstrates that the exchange of low-dimensional aggregated embeddings is highly efficient and does not significantly burden the network bandwidth. In summary, by investing a reasonable amount of training resources, X-VFL achieves significantly higher deployment flexibility and model utility in practical federated scenarios.