Joint Horizontal and Vertical Federated Learning for Multimodal IoT*

Yuanzhe Peng
pengy1@ufl.edu
University of Florida
Gainesville, FL, USA

Zhuo Lu
zhuolu@usf.edu
University of South Florida
Tampa, FL, USA

Jie Xu
jie.xu@ufl.edu
University of Florida
Gainesville, FL, USA

ABSTRACT

Multimodal Federated Learning (FL) integrates two crucial research areas in IoT scenarios: utilizing complementary multimodal data to enhance downstream inference performance and conducting decentralized training to safeguard privacy. However, existing studies primarily focus on applying FL methods after multimodal feature fusion, without fundamentally addressing multimodal FL across both feature and sample spaces. A notable tradeoff persists between the computational demands of multimodal information and the limited computing resources in IoT systems. To tackle this challenge, we propose a Joint Horizontal and Vertical (JHV) FL algorithm tailored for multimodal IoT systems. JHV employs vertical FL to distribute computing tasks across multimodal IoT devices (feature space) and horizontal FL to allocate tasks across multiple silos (sample space). Experimental results on two public multimodal datasets show that JHV outperforms three baseline methods, demonstrating its effectiveness for multimodal IoT systems, especially in rapid and accurate downstream tasks like classification and prediction.

CCS CONCEPTS

• **Computing methodologies** → **Distributed algorithms**.

KEYWORDS

Federated Learning, Multimodal Internet of Things, Testbed.

ACM Reference Format:
Yuanzhe Peng, Zhuo Lu, and Jie Xu. 2024. Joint Horizontal and Vertical Federated Learning for Multimodal IoT. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA*. ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

The Internet of Things (IoT) has rapidly evolved, connecting numerous devices and sensors to enable seamless data exchange. In multimodal IoT, these devices capture various data types from the same sample [17]. Each IoT device, within its respective silo (e.g., home or factory), may have one or multiple sensors gathering different data modalities. For instance, as shown in Fig. 1, each IoT device may have a single sensor type, such as a camera for images or a microphone for audio. These collected multimodal data are then uploaded to the edge server for feature extraction and downstream inference. Compared to single-modal data, multimodal data provide more informative complementary features, thereby improving inference performance [6], which is a key research direction.

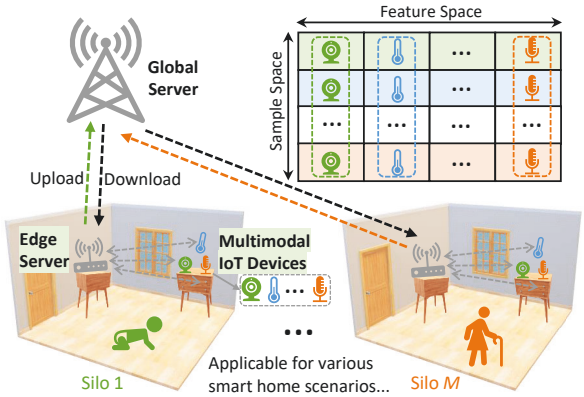

Figure 1: A multimodal IoT system involves problem decomposition across feature space and sample space.

Federated Learning (FL) has emerged as another pivotal research direction in IoT due to its capability to expand the sample space through distributed training with privacy protection [20]. This decentralized approach enables devices to collaboratively train models without centrally aggregating sensitive raw data. In multimodal IoT scenarios, characterized by data heterogeneity and privacy concerns, FL is particularly promising as it aids in developing robust and generalized models across multiple silos [8].

However, current research on FL in multimodal IoT scenarios faces limitations due to the edge server's limited computing resources (e.g., memory or storage) and the underdeveloped computing capabilities of distributed IoT devices. While many IoT devices now possess computational power, most FL approaches for multimodal IoT systems primarily treat them as sensors for data collection, relying on centralized data processing and model training at the edge server [21]. In other words, these methods apply FL after the multimodal feature fusion stage in a straightforward manner, thereby overlooking the potential development of edge computing resources within distributed IoT devices. Essentially, most existing methods for multimodal IoT systems treat multimodal inputs as "single-modal" inputs with richer features and higher dimensions, failing to fundamentally disentangle multimodalities in the

*This work is supported in part by NSF under grants 2033681, 2006630, 2044991, 2319780 and 2319781.

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11…$15.00
https://doi.org/10.1145/3636534.3698245

multimodal FL problem [19]. Thus, we can summarize two challenges hindering the effectiveness of FL in multimodal IoT scenarios. Firstly, edge servers in each silo often face constraints in computing resources such as memory and storage, which impede real-time parallel processing of multimodal data [8]. Secondly, the limited data samples within each silo may cause the local model to perform well within its own silo but result in poor performance across the entire dataset spanning multiple silos.

In light of the above discussion, one key question arises: *How can we fundamentally disentangle multimodalities and leverage the edge computing resources of IoT devices to alleviate the computing burden on the edge server with privacy protection?*

To address this question, we propose a Joint Horizontal and Vertical (JHV) FL algorithm. This approach leverages vertical FL (VFL) to distribute computing resources across multimodal IoT devices (feature space), addressing the first challenge, and horizontal FL (HFL) to distribute computing resources across multiple silos (sample space), addressing the second challenge. This innovative algorithm requires careful consideration of both stale information usage from the VFL component and perturbed gradients from the HFL component, aspects that are not fully understood theoretically and practically. In this paper, we not only theoretically analyze the convergence of JHV but also empirically validate it on a real-world testbed. Extensive experimental results from two public multimodal datasets demonstrate that JHV outperforms three baselines, making it practical for multimodal IoT systems requiring rapid and accurate downstream inference tasks such as classification and prediction.

Our key contributions can be summarized as follows:

- We formulate the multimodal FL problem and fundamentally disentangle it across both the feature space and sample space.
- We propose the JHV algorithm, which distributes computing resources across multimodal IoT devices (feature space) and multiple silos (sample space).
- We theoretically analyze the convergence of the proposed JHV algorithm for non-convex objectives.
- We empirically validate the proposed JHV algorithm on a deployed real-world testbed and evaluate its scalability and generalization on a powerful computing cluster. Extensive experimental results from two multimodal datasets demonstrate its effectiveness and align with our theoretical analysis.

2 PROBLEM FORMULATION

We investigate a multimodal IoT system with M silos, where each silo indexed by m contains N_m samples ($m \in [M]$). The total number of samples across M silos is denoted as $N = \sum_{m=1}^{M} N_m$. Each silo represents a home (or factory) equipped with an edge server and K multimodal IoT devices. These devices are equipped with sensors capable of collecting various data modalities such as images and audio. These K IoT devices collectively have J sensors, where $J \geq K$, capable of capturing data across different modalities corresponding to the same sample. When $K = J$, each IoT device has only one type of sensor (modality, e.g., image or audio), illustrated in Fig. 1.

In the m-th silo, the local dataset $\mathbf{x}_m \in \mathbb{R}^{N_m \times J}$ is vertically partitioned across K IoT devices along the modality axis (feature space). Each IoT device k ($k \in [K]$) may contain a varying number of modalities. However, for simplicity, we assume each IoT device possesses the same number of modalities, specifically $\frac{J}{K}$ modalities per device. The i-th row of $\mathbf{x}_m$ represents a data sample x_m^i. For each sample $x_m^i = [x_m^{1,i}, \cdots, x_m^{k,i}, \cdots, x_m^{K,i}]$, the k-th IoT device holds a disjoint subset of features, denoted as $x_m^{k,i}$. Each x_m^i is associated with a corresponding label y_m^i. Let $\mathbf{y}_m$ denote the vector of all sample labels within the m-th silo. Additionally, $\mathbf{x}_m^k$ represents the local and partial dataset of the k-th IoT device within the m-th silo, where the i-th row corresponds to the data features $x_m^{k,i}$. The objective at the m-th silo level is to minimize:

$$f_m(\Theta_m; \mathbf{x}_m; \mathbf{y}_m) := \frac{1}{N_m} \sum_{i=1}^{N_m} \mathcal{L}\left[\theta_m^0, h_m^1\left(\theta_m^1; x_m^{1,i}\right), \cdots, h_m^K\left(\theta_m^K; x_m^{K,i}\right); y_m^i\right], \quad (1)$$

where $\Theta_m = [\theta_m^0, \theta_m^1, \cdots, \theta_m^k, \cdots, \theta_m^K]$ represents the m-th silo-level model, and $\mathcal{L}(\cdot)$ denotes a loss function that combines the embeddings $\{h_m^k(\theta_m^k; x_m^{k,i})\}_{k=1}^K$ from all IoT devices. For simplicity, we designate $k = 0$ as the edge server (i.e., head [2]) and define $h_m^0(\theta_m^0; x_m^i) := \theta_m^0$ for all x^i, where $h_m^0(\cdot)$ is equivalent to the identity function. In the m-th silo, the partial derivative for the coordinate partition θ_m^k of the k-th IoT device can be expressed as follows:

$$\nabla_k f_m(\Theta_m; \mathbf{x}_m; \mathbf{y}_m) := \frac{1}{N_m} \sum_{i=1}^{N_m} \nabla_{\theta_m^k} \mathcal{L}\left[\theta_m^0, h_m^1\left(\theta_m^1; x_m^{1,i}\right), \cdots, h_m^K\left(\theta_m^K; x_m^{K,i}\right); y_m^i\right]. \quad (2)$$

The stochastic partial derivative of the coordinate partition θ_m^k for the k-th IoT device can be expressed as follows:

$$\nabla_k f_m(\Theta_m; \mathcal{B}_m) := \frac{1}{B_m} \sum_{i \in \mathcal{B}_m} \nabla_{\theta_m^k} \mathcal{L}\left[\theta_m^0, h_m^1\left(\theta_m^1; x_m^{1,i}\right), \cdots, h_m^K\left(\theta_m^K; x_m^{K,i}\right); y_m^i\right], \quad (3)$$

where $\mathcal{B}_m$ denotes a randomly sampled mini-batch of size B_m. We may omit $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}_m$ and $\mathbf{y}_m$ from $f(\cdot)$ or $f_m(\cdot)$ for brevity. Additionally, we define $h_m^k(\theta_m^k; \mathbf{x}_m^{k,\mathcal{B}_m}) := \{h_m^k(\theta_m^k; x_m^{k,\mathcal{B}_m^1}), \cdots, h_m^k(\theta_m^k; x_m^{k,\mathcal{B}_m^{B_m}})\}$ as the set of embeddings of k-th IoT device associated with the mini-batch $\mathcal{B}_m$, where $\mathcal{B}_m^i$ represents the i-th sample in the mini-batch $\mathcal{B}_m$. We consider $\nabla_k f_{\mathcal{B}_m}(\Theta_m)$ and $\nabla_k f_{\mathcal{B}_m}[\theta_m^0, \cdots, h_m^K(\theta_m^K; \mathbf{x}_m^{K,\mathcal{B}_m})]$ equivalent and use them interchangeably.

Thus, the global objective is to minimize the following:

$$f(\Theta) := \frac{1}{N} \sum_{m=1}^{M} N_m f_m(\Theta), \quad (4)$$

where $\Theta = [\theta^0, \theta^1, \cdots, \theta^k, \cdots, \theta^K]$ denotes the global full model, and $\theta^k = \frac{1}{N} \sum_{m=1}^{M} N_m \theta_m^k$ denotes the partial model on the k-th (type of) IoT device. This objective evaluates how well the model fits the entire dataset across K IoT devices and M silos, distinguishing it from any existing HFL-type [15] or VFL-type [13] problem.

3 JHV ALGORITHM

3.1 VFL across K IoT Devices

At the beginning of each communication round (i.e., $t \bmod Q = 0$) within the m-th silo, designated as t_0, a mini-batch $\mathcal{B}_m$ is randomly sampled from $\mathbf{x}_m$. Each IoT device conducts block coordinate stochastic gradient descent on its local model parameters (i.e., θ_m^k) in parallel for Q local iterations. Specifically, for the k-th IoT device to compute the stochastic partial gradient regarding its features across

partial modalities, it needs the embeddings computed by all other devices k' ($k' \neq k$), as well as its own k-th device embeddings $h_m^k(\theta_m^{k,t})$. Within each silo m, these embeddings from different IoT devices are shared with the edge server and then distributed to all IoT devices. We define $\Phi_m^{-k,t_0} := \{(h_m^{k'}(\theta_m^{k',t_0})) \mid k' \neq k\}_{k'=0}^{K-1}$ as the set of embeddings from other devices $k' \neq k$; thus, the set of embeddings used by the k-th device is $\Phi_m^{k,t} = \{\Phi_m^{-k,t_0}; h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m})\}$, which inevitably contains stale information Φ_m^{-k,t_0} during $t > t_0$ iterations in this round. For each iteration t, the k-th device updates θ_m^k by computing the stochastic partial derivatives $\nabla_k f_{\mathcal{B}_m}(\Phi_m^{k,t}; \mathbf{y}_m^{\mathcal{B}_m})$ and applying a gradient step with step size η. It is noteworthy that each IoT device relies on a stale view of the silo-level model to compute its partial gradient during multiple local iterations by reusing the embeddings received at the start of each communication round, which may affect convergence. However, in Section 3.3, we provide a theoretical analysis demonstrating that our proposed JHV algorithm converges despite devices performing multiple local iterations with inevitably stale information at this stage.

In addition to the inevitable use of stale information, another significant deviation of JHV from previous VFL algorithms, such as FedBCD [14], is its adoption of the edge server model (referred to as the "head" [2], denoted as θ_m^0) with trainable parameters, facilitating the integration of arbitrary multimodal fusion networks. Moreover, while we assume that both the edge server and all IoT devices have the labels $\mathbf{y}_m$ in each silo, we also consider scenarios where this may not be the case. If labels are only available to one party (e.g., the edge server), it can still provide enough information for other devices to compute gradients for certain model architectures [13].

3.2 HFL across M Silos

In IoT scenarios, the local datasets within each silo (e.g., homes or factories) are influenced by available samples and various modalities, resulting in significant variations in both sample size and multimodal data distribution across different silos. Non-Independent and Identically Distributed (Non-IID) datasets may enable a silo-level model to fit well locally but not generalize to the entire dataset across M silos. To develop a global model that performs well across entire multimodal IoT systems, we employ HFL across all M silos, ensuring it operates concurrently with VFL conducted within each silo. Specifically, at each communication point where $t \bmod Q = 0$, the global server aggregates partial models across K IoT devices from M silos. For the k-th IoT device, this aggregation is computed as $\theta^{k,t} = \frac{1}{N}\sum_{m=1}^{M} N_m \theta_m^{k,t}$, and then the updated models $\theta^{k,t}$ are broadcast back to all silos. Importantly, this aggregation involves different types of IoT devices representing various modalities (features). A key distinction of our proposed JHV from existing multimodal FL methods is our fundamental disentanglement of multimodal inputs, rather than treating them incrementally as a "single-modal" input with richer features and higher dimensions.

While HFL already provides privacy benefits by avoiding the sharing of raw data, we offer additional and more stringent solutions. Within each silo, IoT devices share only embeddings and compute partial derivatives related to their local models, thereby further mitigating privacy concerns associated with the transmission of raw data. Furthermore, we can enhance security against sophisticated attacks using methods such as secure multi-party computation [4] or homomorphic encryption [1].

Algorithm 1: JHV

Initialize: $\theta_m^{0,t=0}$, $\theta_m^{k,t=0}$, $\forall k \in [K], \forall m \in [M]$;
for $t = 0, 1, \cdots, T-1$ **do**
  if $t \mod Q = 0$ **then**
    # *VFL*
    for $m = 1, 2, \cdots, M$ *in parallel* **do**
      for $k = 1, 2, \cdots, K$ *in parallel* **do**
        IoT device uploads $h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m})$;
      $\Phi_m^{t_0} \leftarrow \{\theta_m^{0,t}, h_m^1(\theta_m^{1,t}), \cdots, h_m^K(\theta_m^{K,t})\}$;
      Edge server sends $\Phi_m^{t_0}$ to all K devices;
    # *HFL*
    for $k = 0, 1, \cdots, K$ *in parallel* **do**
      Global server computes $\theta^{k,t} = \frac{1}{N}\sum_{m=1}^{M} N_m \theta_m^{k,t}$;
      Global server sends $\theta^{k,t}$ to all M silos;
    for $m = 1, 2, \cdots, M$ *in parallel* **do**
      for $k = 0, 1, \cdots, K$ *in parallel* **do**
        $\theta_m^{k,t} \leftarrow \theta^{k,t}$;
  for $m = 1, 2, \cdots, M$ *in parallel* **do**
    for $k = 0, 1, \cdots, K$ *in parallel* **do**
      $\Phi_m^{k,t} \leftarrow \{\Phi_m^{-k,t_0}; h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m})\}$;
      $\theta_m^{k,t+1} \leftarrow \theta_m^{k,t} - \eta \nabla_k f_{\mathcal{B}_m}(\Phi_m^{k,t}; \mathbf{y}_m^{\mathcal{B}_m})$;

3.3 Convergence Analysis

Considering that the proposed Algorithm 1 raises two concerns: (1) multiple local iterations using inevitably stale information and (2) significant variations in both sample size and multimodal data distribution across multiple silos, which have not been fully understood from a theoretical point, we provide the convergence analysis.

Assumption 1. *There exist positive constants $L < \infty$ and $L_k < \infty$, for $m \in [M]$, $k \in [K]$, such that for all Θ and Θ', the objective function satisfies:*

$$\|\nabla f_m(\Theta) - \nabla f_m(\Theta')\| \leq L\|\Theta - \Theta'\|, \tag{5}$$

$$\|\nabla_k f_m(\Theta) - \nabla_k f_m(\Theta')\| \leq L_k\|\Theta - \Theta'\|. \tag{6}$$

Assumption 2. *The stochastic partial derivatives are unbiased for each mini-batch $\mathcal{B}$:*

$$\mathbb{E}\left[\nabla_k f_m(\Theta; \mathcal{B})\right] = \nabla_k f_m(\Theta). \tag{7}$$

Assumption 3. *There exist constants σ_k such that the variance of the stochastic partial derivatives is bounded for a mini-batch $\mathcal{B}$ of size B:*

$$\mathbb{E}\left[\|\nabla_k f_m(\Theta; \mathcal{B}) - \nabla_k f_m(\Theta)\|^2\right] \leq \frac{\sigma_k^2}{B}. \tag{8}$$

Assumption 4. *There exists a constant δ such that the expected squared Euclidean norm of $\nabla_k f_m(\Theta; \mathcal{B})$ is uniformly bounded for all*

IoT devices $\mathcal{K}$ of size K:

$$\mathbb{E}\left[\|\nabla_k f_m(\Theta;\mathcal{B})\|^2\right] \leq \frac{\delta^2}{K}. \tag{9}$$

Theorem 1. Suppose Assumptions 1–4 hold, $\eta \leq \frac{1}{\max\{L,L_k\}}$, then the average squared gradient over P global rounds (i.e., $T = Q \times P$ iterations) of Algorithm 1 is bounded:

$$\begin{aligned}&\tfrac{1}{P}\textstyle\sum_{t=0}^{P-1}\mathbb{E}\left[\left\|\nabla f\left(\Theta^t\right)\right\|^2\right]\\ &\leq \tfrac{2[f(\Theta^0)-f(\Theta^*)]}{\eta P} + 2\eta^2 Q^2 \textstyle\sum_{k=0}^{K} L_k^2\left(\tfrac{K+1}{M}\tfrac{\sigma_k^2}{B}\right.\\ &\left.+(\tfrac{K+1}{M}+1)\tfrac{\delta^2}{K}\right)+\eta L \tfrac{1}{M}\textstyle\sum_{k=0}^{K}(\tfrac{\sigma_k^2}{B}+\tfrac{\delta^2}{K}),\end{aligned} \tag{10}$$

where $f(\Theta^*)$ is the optimal value of the global objective (4).

Remark 1. The convergence error in Theorem 1 arises from parallel updates on coordinate blocks in Algorithm 1, which depend on the communication frequency (Q), the number of IoT devices (K), and the number of silos (M). The first term is determined by the disparity between the initial model and the optimal model, diminishing as the number of global rounds (P) approaches infinity. The remaining terms denote errors arising from (1) multiple local iterations using stale information in the VFL component, and (2) variance in stochastic gradients in the HFL component. We further explore these aspects in Section 4.3.4 through extensive experiments.

4 EXPERIMENTS

4.1 Real-World Testbed

We developed a prototype of our proposed JHV using a hardware testbed consisting of two silos deployed in our lab under real-world network conditions. Each silo is equipped with the following devices: an NVIDIA Jetson TX2 (TX2), featuring an NVIDIA Pascal GPU with 256 CUDA cores and 8GB LPDDR4 memory, running Ubuntu 18.04; an NVIDIA Jetson Xavier NX (NX), featuring a 384-core NVIDIA Volta GPU with 48 Tensor Cores and 8GB 128-bit LPDDR4x memory, also running Ubuntu 18.04; and an edge server with an Intel Core i7-11370 CPU, running Ubuntu 20.04. These devices offer varying computational capabilities. The NVIDIA Jetson devices can locally train deep learning models and communicate with the edge server via 802.11ac WLAN. Due to limitations of available lab devices, an edge server within the first silo handles both its own training tasks and the model aggregation function of the global server. Additionally, we strategically deployed heterogeneous IoT devices with varying computational capabilities within each silo to accommodate natural variations in computation times across different modalities in multimodal scenarios. The hardware system setup is illustrated in Fig. 2.

4.2 Experimental Setup

4.2.1 Datasets. *ModelNet10*[1] comprises images of Computer-Aided Design (CAD) models depicting various objects [18]. Each CAD model is represented by 12 images captured from different camera views. Notably, these images are not generated through data augmentation techniques such as flipping or noise addition, making ModelNet10 widely utilized as a multimodal dataset. Similar

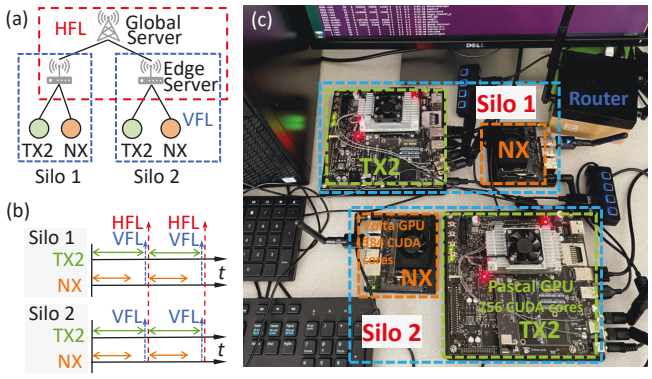

Figure 2: Our real-world testbed deployment includes two silos with heterogeneous devices: (a) Deployment diagram, (b) Training methodology, and (c) Lab deployment environment.

to [12, 16], we split the dataset into 3,991 training samples and 908 testing samples for the task of multi-class classification across 10 object categories, such as bed, dresser, sofa, table, monitor, etc. *MIMIC-III*[2] (Medical Information Mart for Intensive Care) dataset contains anonymized information of patients admitted to critical care units in a hospital [10]. We follow the data processing steps outlined in [5] to obtain 14,681 training samples and 3,236 test samples. Each sample comprises 48 time steps corresponding to 48 hours, with each time step having 76 features, such as demographic information, vital signs, medications, etc. The objective is to predict in-hospital mortality as a binary classification task.

4.2.2 Implementation and Reproducibility. For the ModelNet10 dataset, during the training process within each silo, we vertically partition the local data along the 12-view axis into two vertical partitions (feature space). The TX2 and NX each contain 6 of the 12 views. Subsequently, each IoT device trains a ResNet18 model with a penultimate layer. The concatenated embeddings (features) are then fed into the classifier layer (head) at the edge server, which employs cross-entropy loss for class prediction. We employ 5-fold cross-validation for hyperparameter selection, such as performing grid search for the learning rate within the range $[0.0001, 0.002]$.

For the MIMIC-III dataset, our preprocessing procedure partitions the dataset into various prediction cases, with our experiments specifically targeting the prediction of in-hospital mortality. During the training process within each silo, we vertically partition the local data along the 76-features axis into K vertical partitions (e.g., when $K = 2$, each partition contains 38 of the 76 features). Each device trains an LSTM model with a linear layer. The concatenated embeddings (features) are then fed into the classifier layer (head) at the edge server, which utilizes cross-entropy loss for class prediction. We utilize 5-fold cross-validation for hyperparameter selection, including a grid search for the learning rate within the range $[0.001, 0.02]$. Due to the imbalanced nature of the MIMIC-III dataset, consisting of only 16% positive samples (indicating that most patients did not die in the hospital for the in-hospital mortality prediction task), we assess the generalization performance on the

[1] https://modelnet.cs.princeton.edu/

[2] https://mimic.mit.edu/

test dataset using the F1 score as an evaluation metric. The F1 score represents the harmonic mean of precision and recall, calculated for the global model across the entire test dataset.

4.2.3 Baselines. Our baseline experiments cover three categories, each potentially associated with several existing multimodal FL methods. To vividly demonstrate the efficacy of our proposed JHV, we use blue and red dashed boxes to represent VFL and HFL, respectively, and gray dashed boxes to represent sample size, highlighting their training differences, as depicted in Fig. 3.

Baseline 1: Local training with multimodal data [6], [7]. This baseline corresponds to the traditional multimodal learning approach for IoT systems. However, the computing resources of the edge server in the IoT system are limited and cannot process all multimodal inputs in parallel. For example, when processing image or video data, the GPU memory might become fully utilized, leading to delays in processing other modal data, especially those requiring real-time processing. Additionally, this baseline fails to expand the training sample space while ensuring privacy.

Baseline 2: VFL with multimodal data [14], [3]. This baseline corresponds to a form of multimodal FL methods that explores VFL for distributed training of multimodal data. However, these methods do not effectively address the challenge of limited samples within each silo (home) in IoT scenarios.

Baseline 3: HFL with multimodal data [21], [19]. This baseline corresponds to a form of multimodal FL that does not involve disentangling the training of multimodal data across feature space. Here, the multimodal input can be perceived as a "single modal input" with richer information and higher dimensions. Besides, the computing resources of the edge server within each silo are limited, thus all multimodal inputs cannot be processed in parallel.

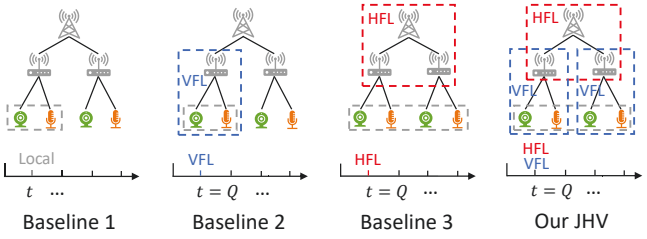

Figure 3: Comparison between JHV and three baselines.

4.3 Results and Analysis

4.3.1 Convergence Performance: To fairly compare the convergence performance between JHV and three baselines, we fixed the communication frequency ($Q = 5$), the number of silos ($M = 2$), and the number of IoT devices ($K = 2$), i.e., each device contains only half of the sample space and half of the feature space as shown in Fig. 2 (a). Each experiment was repeated 10 times.

As shown in the results in Fig. 4, based on the ModelNet10 dataset implemented on the testbed, JHV and Baseline 2 effectively utilize distributed computing resources from two IoT devices (NX and TX2) to accelerate convergence compared to Baseline 3 and Baseline 1, respectively. Additionally, the convergence errors of JHV and Baseline 3 are smaller than those of Baseline 2 and Baseline 1, respectively, due to the utilization of HFL across two silos. Notably, in Section 4.3.3, implemented on a powerful computing cluster which can flexibly vary K and M, the superiority of our JHV is more apparent with larger values of K and M, where multimodal data becomes more distributed in the feature space and sample space, which is more common in real-world IoT scenarios.

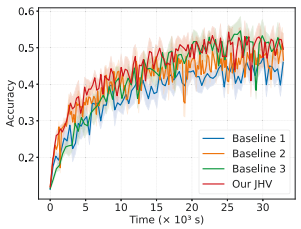

Figure 4: Comparison of convergence performance on the ModelNet10 dataset.

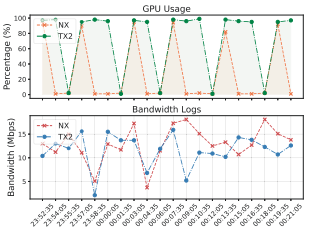

Figure 5: Dynamic resources (GPU usage, bandwidth logs) in Silo 2, shown in Fig. 2 (c).

4.3.2 Dynamic Resource Usage: Another key challenge addressed in deploying JHV on the testbed is the resource dynamics in real-world systems. We utilized the package `jetson-stats`[3] from JetPack 4.6.1 to record the GPU usage of the hardware system and `speedtest-cli`[4] to measure the bandwidth. As shown in Fig. 5, we recorded the dynamic resource changes of one of the silos during half an hour of continuous operation. We observed that the NX experienced more "waiting" states compared to the TX2 due to varying computing capabilities during synchronous communication. Moreover, the bandwidth of different devices within the same time slot fluctuated within a dynamic range. These findings also inspire our future work on asynchronous communication.

4.3.3 Scalability and Generalization: To further assess the scalability and generalizability of our proposed JHV, we set up more silos ($M = 10$) and more devices ($K = 2, 4, 19$) on a powerful computing cluster with 4× 12-core Intel Xeon Gold 6126 CPUs, 1× Tesla V100, and 3× Tesla P100 GPUs, and evaluated JHV using another public multimodal dataset, MIMIC-III. Notably, while the computing resources employed here may differ from those typically found in IoT systems, this is due to the complexity of our multimodal datasets, which far exceed that of typical IoT scenarios. The selected dataset is also not confined to IoT scenarios but encompasses more complex, general multimodal scenarios, as it includes up to 76 features corresponding to the same data sample. Considering that communication latency within the internal cluster cannot be measured as accurately as in the testbed using wall clock time, we introduce a normalized time unit to scale time accumulation appropriately. This time scale is based on the average message size of each transmission in VFL and HFL, along with the median network speed in the US in April 2024 [9]. To fairly compare the convergence performance between JHV and three baselines, we first set the communication frequency ($Q = 5$), number of silos ($M = 10$), and number of devices ($K = 2$). Each experiment was repeated 10 times.

As illustrated in Fig. 6 using the MIMIC-III dataset, JHV and Baseline 2 effectively utilize distributed computing resources from

[3]https://github.com/rbonghi/jetson_stats
[4]https://pypi.org/project/speedtest-cli/

K devices to expedite convergence compared to Baseline 3 and Baseline 1, respectively. Furthermore, the convergence errors of JHV and Baseline 3 are smaller than those of Baseline 2 and Baseline 1, respectively, due to the utilization of HFL across M silos, which expands the sample space while protecting privacy. We also observed that as the accumulated time increases sufficiently, the convergence error of Baseline 3 tends to approach that of our proposed JHV. Similarly, the convergence error of Baseline 1 tends to approach that of Baseline 2. This observation is intuitive because their respective sample spaces are consistent, as illustrated in the gray dotted box in Fig. 3. The key distinction lies in our JHV approach, which optimizes the distribution of computing resources comprehensively across both feature and sample spaces on all available devices.

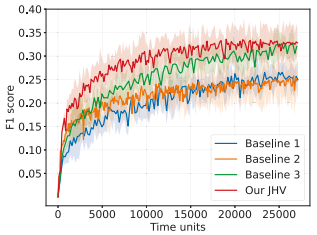

Figure 6: Comparison of convergence performance on the MIMIC-III dataset.

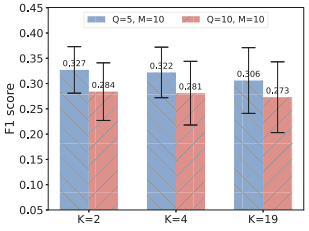

Figure 7: The impact of Q and K on the MIMIC-III dataset (Q=5/10, K=2/4/19).

4.3.4 Ablation Study: We further varied Q and K to evaluate their impact on convergence performance, considering real-world factors such as communication cost constraints or the number of available IoT devices. This exploration aims to validate if the experimental results align with our theoretical analysis in Section 3.3. We do not vary the number of silos (M) here, as fixing the total number of samples N while varying M also changes the Non-IID degree, complicating a fair assessment of its impact on convergence performance. As depicted in Fig. 7, we maintained a fixed total number of iterations; thus, a larger Q implies a lower communication frequency, resulting in poorer convergence performance, which aligns with our theoretical results in Theorem 1. Compared to the impact of Q, the effect of K is relatively minor. As K decreases, both the convergence error and variance tend to decrease slightly. This observation also aligns with intuition and the theoretical results in Theorem 1, as a smaller K suggests that data are more "pooled" together in the feature space. In practical scenarios, the influence of the K factor is generally moderate, assuming that the number of IoT devices within each home is typically not very large [11].

5 CONCLUSION

In conclusion, we propose the JHV algorithm, which uniquely leverages VFL to distribute computing resources across multimodal IoT devices and HFL to distribute computing resources across multiple silos. We not only theoretically analyze the convergence of JHV but also empirically deploy and verify it on a real-world testbed. Experimental results on two public multimodal datasets demonstrate that JHV outperforms three baselines, thereby making it practical for multimodal IoT systems that require rapid and accurate downstream inference such as classification and prediction.

REFERENCES

[1] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *Comput. Surveys* 51, 4 (2018), 1–35.

[2] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting Shared Representations for Personalized Federated Learning. In *International conference on machine learning*. PMLR, Virtual Event, 2089–2099.

[3] Maoguo Gong, Yuanqiao Zhang, Yuan Gao, AK Qin, Yue Wu, Shanfeng Wang, and Yihong Zhang. 2023. A Multi-Modal Vertical Federated Learning Framework Based on Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security* 19 (2023), 1826–1839.

[4] Bin Gu, An Xu, Zhouyuan Huo, Cheng Deng, and Heng Huang. 2021. Privacy-preserving Asynchronous Vertical Federated Learning Algorithms for Multiparty Collaborative Learning. *IEEE Transactions on Neural Networks and Learning Systems* 33, 11 (2021), 6103–6115.

[5] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask Learning and Benchmarking with Clinical Time Series Data. *Scientific Data* 6, 1 (2019), 96.

[6] Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. 2021. What Makes Multi-modal Learning Better Than Single (Provably). *Advances in Neural Information Processing Systems* 34 (2021), 10944–10956.

[7] Zhenhua Huang, Xin Xu, Juan Ni, Honghao Zhu, and Cheng Wang. 2019. Multimodal Representation Learning for Recommendation in Internet of Things. *IEEE Internet of Things Journal* 6, 6 (2019), 10675–10685.

[8] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M Hadi Amini. 2021. A Survey on Federated Learning for Resource-constrained IoT Devices. *IEEE Internet of Things Journal* 9, 1 (2021), 1–24.

[9] Speedtest Global Index. 2024. *United States Median Country Speeds, February 2024.* Speedtest. https://www.speedtest.net/global-index/united-states

[10] 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. 2016. MIMIC-III, A Freely Accessible Critical Care Database. *Scientific Data* 3, 1 (2016), 1–9.

[11] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning* 14, 1–2 (2021), 1–210.

[12] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2019. Point2sequence: Learning the Shape Representation of 3D Point Clouds with an Attention-based Sequence to Sequence Network. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (2019), 8778–8785.

[13] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. 2024. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Transactions on Knowledge and Data Engineering* 36 (2024), 3615–3634.

[14] Yang Liu, Xinwei Zhang, Yan Kang, Liping Li, Tianjian Chen, Mingyi Hong, and Qiang Yang. 2022. FedBCD: A Communication-Efficient Collaborative Learning Framework for Distributed Features. *IEEE Transactions on Signal Processing* 70 (2022), 4277–4290.

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics* 54 (2017), 1273–1282.

[16] Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhan Zhang, and Yufei Guo. 2023. Spiking PointNet: Spiking Neural Networks for Point Clouds. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., New Orleans, 41797–41808.

[17] Amit Kumar Singh, Deepa Kundur, and Mauro Conti. 2024. Introduction to the Special Issue on Integrity of Multimedia and Multimodal Data in Internet of Things. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 6 (2024), 1–4.

[18] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. CVPR, Boston, 1912–1920.

[19] Baochen Xiong, Xiaoshan Yang, Fan Qi, and Changsheng Xu. 2022. A Unified Framework for Multimodal Federated Learning. *Neurocomputing* 480 (2022), 110–118.

[20] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 1–19.

[21] Yuchen Zhao, Payam Barnaghi, and Hamed Haddadi. 2022. Multimodal Federated Learning on IoT Data. In *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, Milan, 43–54.