Hybrid Federated Learning for Multimodal IoT Systems

Yuanzhe Peng, Yusen Wu, *Member, IEEE*, Jieming Bian, and Jie Xu, *Senior Member, IEEE*

***Abstract*—Multimodal federated learning (FL) targets the intersection of two promising research directions in Internet of Things (IoT) scenarios: 1) leveraging complementary multimodal information to enhance downstream inference performance and 2) conducting distributed training with privacy protection. However, the majority of existing works primarily focus on applying different FL methods in a straightforward manner after the multimodal feature fusion stage without fundamentally disentangling the multimodal FL across both the feature space and the sample space. There still exists an important tradeoff between the computationally demanding nature of multimodal information and the limited computing resources in IoT systems. To tackle this challenge, we propose a hybrid FL algorithm tailored for multimodal IoT systems (HFM). HFM utilizes vertical FL (VFL) to distribute computing resources across the feature space and horizontal FL (HFL) to distribute computing resources across the sample space. This innovative algorithm necessitates consideration of both stale information from the VFL component and perturbed gradients from the HFL component, which is not fully understood from a theoretical point. In this article, we theoretically prove that the convergence of HFM depends on the frequency of VFL communication and HFL communication, as well as the number of vertical partitions and horizontal partitions. Furthermore, we empirically demonstrate that HFM outperforms three types of baselines based on two public multimodal data sets, thereby making it practical for multimodal IoT systems that require rapid and accurate downstream inference tasks, such as classification, prediction, etc.**

***Index Terms*—Edge computing, federated learning (FL), multimodal Internet of Things (IoT), nonconvex optimization.**

I. Introduction

THE Internet of Things (IoT) has undergone a remarkable transformation by intricately weaving together a vast array of devices and sensors, thereby facilitating seamless data exchange and automation across various fields. Within the realm of multimodal IoT, these IoT devices possess the capability to capture a diverse range of data types corresponding to the same sample [1]. Within each silo (e.g., a household or a factory), each IoT device may contain one or multiple sensors tasked with collecting different data modalities. For example, as illustrated in Fig. 1, each IoT device possesses a single type of sensor, such as a camera for capturing images or a microphone for recording audio. Subsequently, these multimodal data are uploaded to the edge server for feature extraction and downstream inference tasks, such as classification, prediction, etc. Compared to single-modal data, multimodal data often offer a more comprehensive array of complementary features, thereby enhancing downstream inference performance, which is the predominant research focus in IoT scenarios [2].

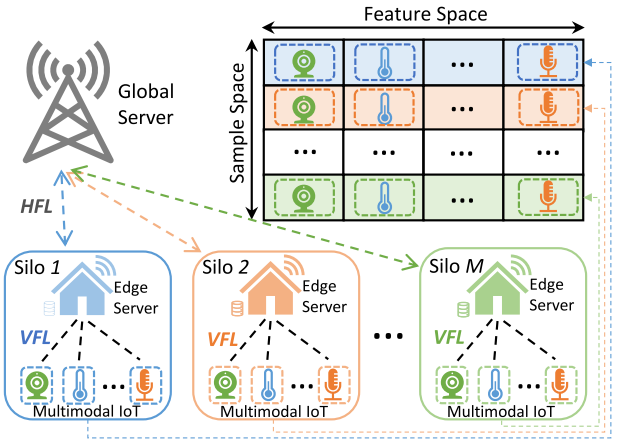


Fig. 1. Multimodal IoT system with problem decomposition across the feature space and sample space. Each silo contains an edge server with limited computing resources (such as memory or storage) and multiple multimodal IoT devices. Each silo encompasses a portion of the horizontally distributed sample space, while each IoT device covers a portion of the vertically distributed feature space. Our research aims to optimize the distribution of computing resources in multimodal IoT systems by conducting distributed training across both the feature space and the sample space while ensuring privacy protection.

Federated learning (FL) has emerged as another significant area of interest within the IoT landscape, owing to its ability to broaden the sample space for model training while safeguarding data privacy [3]. This decentralized learning paradigm empowers devices to collaboratively train models without the need to aggregate sensitive data in a central repository. In the realm of multimodal IoT, characterized by data heterogeneity and privacy concerns, the adoption of FL holds particular promise, offering the potential for developing robust and generalized models across diverse data sources [4].

However, current research on FL in multimodal IoT scenarios faces constraints, primarily stemming from the limited computing resources (e.g., memory or storage) of the

Manuscript received 9 July 2024; accepted 29 July 2024. Date of publication 14 August 2024; date of current version 24 October 2024. This work was supported in part by NSF under Grant 2033681, Grant 2006630, Grant 2044991, and Grant 2319780. *(Corresponding author: Yuanzhe Peng.)*

Yuanzhe Peng, Jieming Bian, and Jie Xu are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: pengy1@ufl.edu; jieming.bian@ufl.edu; jie.xu@ufl.edu).

Yusen Wu is with the Frost Institute for Data Science and Computing, University of Miami, Coral Gables, FL 33146 USA (e-mail: yxw1259@miami.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2024.3443267, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2024.3443267

edge server and the incomplete development of computing capabilities in distributed IoT devices. Although 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 (e.g., feature extraction) and model training at the edge server [5]. Consequently, they overlook the potential development of computing resources within distributed IoT devices. Essentially, most existing methods for multimodal IoT systems, which treat the multimodal input as a "single-modal" input with richer features and higher dimensions, fail to address the fundamental challenge of multimodal FL [6]. Instead of applying FL in a straightforward manner after the multimodal feature fusion stage, our objective is to empower distributed IoT devices not only as sensors for collecting information but also as edge computing devices for feature extraction and downstream inference, thereby alleviating the computing burden on the edge server. Specifically referring to Fig. 1, our research aims to optimize the distribution of computing resources in multimodal IoT systems by conducting distributed training across both the feature space and the sample space.

We have identified two primary bottlenecks impeding the efficacy of FL in multimodal IoT scenarios. First, the edge server within individual silos often struggles with limited computing resources. Memory or storage constraints pose significant challenges, particularly in enabling real-time parallel processing of multimodal data [4]. Second, the limited sample size within each silo, coupled with the diverse distribution of multimodal data types, exacerbates the nonindependent and identically distributed (non-IID) problem in multimodal IoT scenarios, compromising model performance and hindering generalization across the entire data set. These bottlenecks present formidable challenges and require a delicate tradeoff between the computationally demanding nature of multimodal information and the limited computing resources, especially in scenarios that require rapid and accurate downstream inference within multimodal IoT systems. In light of the above discussion, two questions arise.

1) *How* to design an algorithm to optimize the distribution of computing resources across both feature space and sample space while ensuring privacy protection?
2) *How and why* does the convergence performance of the algorithm change when adjusting parameters under various constraints in real-world multimodal IoT systems?

To answer the first question, we propose a hybrid FL algorithm tailored for multimodal IoT systems (HFM), where computing resources, such as memory and storage, are often limited. HFM utilizes vertical FL (VFL) to distribute computing resources across the feature space and horizontal FL (HFL) to distribute computing resources across the sample space. This innovative algorithm necessitates consideration of both stale information from the VFL component and perturbed gradients from the HFL component, which is not fully understood from a theoretical point.

To answer the second question, we theoretically prove that the convergence of HFM depends on the frequency of VFL communication and HFL communication, as well as the number of vertical partitions and horizontal partitions. Besides, we empirically demonstrate that HFM outperforms three types of baselines in terms of convergence rate and convergence error based on two public multimodal data sets, thereby making it practical for multimodal IoT systems that require rapid and accurate downstream inference tasks, such as classification, prediction, etc.

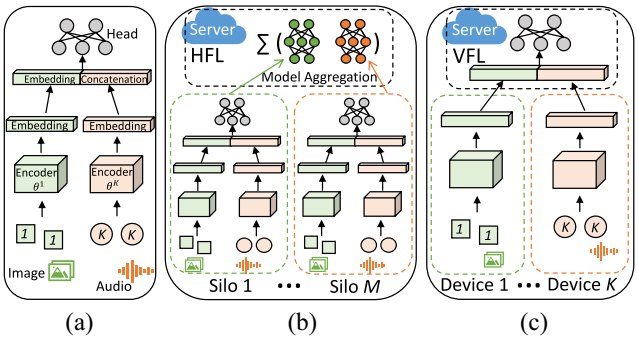


Fig. 2. (a) Multimodal Learning leverages complementary multimodal information but does not incorporate FL to safeguard privacy. (b) Multimodal HFL integrates the FL paradigm in a straightforward manner after the stage of feature fusion but does not fundamentally disentangle multimodalities. (c) Multimodal VFL faces challenges due to limited sample size in IoT scenarios.

Our key contributions can be summarized as follows.

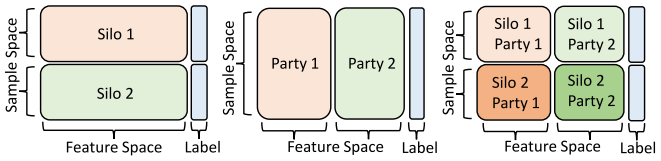
1) We formulate the multimodal FL problem and disentangle it across both the feature space and sample space for IoT systems demanding rapid and accurate inference.
2) We propose a hybrid FL algorithm, named HFM, specifically designed for multimodal IoT systems. HFM utilizes VFL to distribute computing resources across the feature space and HFL to distribute computing resources across the sample space.
3) We analyze the convergence of the HFM algorithm for nonconvex objectives and demonstrate its dependency on the frequency of VFL communication and HFL communication, as well as the number of vertical partitions and horizontal partitions.
4) We validate the HFM algorithm and theoretical analysis through extensive experiments involving various objectives based on two public multimodal data sets.

The remainder of this article is structured as follows. Section II discusses related work on FL for multimodal IoT systems. Section III mathematically formulates the problem. Section IV introduces the HFM algorithm, while its theoretical analysis is provided in Section V. Section VI presents extensive experiments to validate the HFM algorithm and the theoretical analysis. Finally, Section VII concludes this article.

II. Related Work

Multimodal FL intersects two promising research directions in IoT scenarios: first, harnessing complementary multimodal information to improve downstream inference performance, as shown in Fig. 2(a); second, conducting HFL or VFL with privacy protection, as illustrated in Fig. 2(b) and (c). In this section, we delve into related work on FL for IoT systems, followed by a discussion of recent advancements in FL tailored for multimodal scenarios.

FL has emerged as a significant area of research within the IoT domain, presenting a decentralized approach to model



(a) (b) (c)

Fig. 3. (a) HFL addresses horizontally partitioned sample spaces with consistent feature spaces. (b) VFL addresses vertically partitioned feature spaces with consistent sample spaces. (c) Our hybrid FL arises from partitioning both the sample space and the feature space in multimodal IoT systems.

training across distributed data sets [7]. The literature on FL for IoT systems can be broadly classified into two main types, each addressing distinct data distribution patterns [3].

First, HFL focuses on scenarios where data samples are horizontally partitioned with consistent vertical feature spaces, as depicted in Fig. 3(a). For instance, various horizontal silos, such as homes or factories possess limited data sample sizes while prioritizing privacy preservation, necessitating the use of the HFL paradigm to improve downstream inference performance [4]. While some works explore HFL in multimodal contexts, they integrate the HFL paradigm after the multimodal feature fusion stage in a straightforward manner [5], [6], [8]. Critically, the multimodal input in these works can be treated as a single-modal input with richer features and higher dimensions, failing to disentangle the essence of the multimodal FL issue across the feature space.

Second, VFL addresses scenarios where the feature space is vertically partitioned with consistent horizontal sample spaces, as shown in Fig. 3(b). This aligns with another common application in IoT scenarios, where different IoT providers may have data of different features (modalities) of the same batch of customers [9]. For example, Liu et al. [10] proposed the FedBCD algorithm, enabling clients to independently conduct stochastic gradient algorithms while leveraging vertically partitioned data features. Furthermore, some VFL works in multimodal scenarios focus on optimizing communication latency [11]. For example, Wang et al. [12] introduced the TVFL algorithm to address communication cost challenges in VFL settings. However, it is noteworthy that the most general scenario for multimodal systems is that both the sample space and the feature space are partitioned, as depicted in Fig. 3(c). Hence, neither HFL nor VFL alone is adequate to address the challenges posed by multimodal IoT systems due to their single-dimensional partitioning paradigm.

While some current multimodal FL works involve simultaneous horizontal and vertical partitioning, such as the work by Yu et al. [13], who proposed a multilayer SGD algorithm for FL in E-health, their discussion remains limited to two modalities and fails to provide insights into the impact of vertical partitions, while also neglecting the incorporation of an arbitrary feature fusion model (head) at the server. Su and Lau [14] introduced a hierarchical FL framework for hybrid data partitioning; similarly, Das et al. [15] proposed a tiered decentralized coordinate descent algorithm for two-tiered networks. However, both of their utilization of HFL at the bottom of VFL is a natural extension of typical VFL and does not address the key issue of optimizing the distribution of computing resources across feature space in multimodal IoT scenarios. Additionally, several studies have explored modality heterogeneity and the issue of missing modalities in multimodal FL [16], [17], but they do not provide any theoretical analysis, and their personalized FL objective differs from our global FL objective.

TABLE I
ESSENTIAL NOTATIONS

Notation	Definition
M	Number of horizontal silos.
N_m	Number of samples within the m-th silo, $m \in [M]$.
N	Number of total samples across M silos, $N = \sum_{m=1}^{M} N_m$.
J	Number of IoT sensors (modalities), $J \geq K$.
K	Number of IoT devices (vertical parties). When $K = J$, i.e., each IoT device has one sensor, as shown in Fig. 1.
$\mathbf{x}, \mathbf{y}$	Full dataset across M silos, $\mathbf{x}, \mathbf{y} = \{\mathbf{x}_m, \mathbf{y}_m\}_{m=1}^{M}$.
$\mathbf{x}_m, \mathbf{y}_m$	m-th silo-level dataset, $\mathbf{x}_m, \mathbf{y}_m = \{\mathbf{x}_m^k, \mathbf{y}_m\}_{k=1}^{K}$.
$\mathbf{x}_m^k$	k-th vertical party's data within the m-th silo.
$\mathcal{B}_m$	Randomly sampled mini-batch of size B_m.
η	Learning rate.
Q	VFL communication frequency (every Q iterations).
RQ	HFL communication frequency (every RQ iterations).
P	The number of total global rounds.
T	The number of total iterations, $T = RQ \times P$.
Θ	Global model, $\Theta = [\theta^0, \theta^1, \ldots, \theta^k, \ldots, \theta^K]$.
θ^k	k-th vertical party model, $\theta^k = \frac{1}{N}\sum_{m=1}^{M} N_m \theta_m^k$.
Θ_m	m-th silo model, $\Theta_m = [\theta_m^0, \theta_m^1, \ldots, \theta_m^k, \ldots, \theta_m^K]$.
θ_m^0	m-th silo edge server model (i.e., head), $k = 0$.
θ_m^k	m-th silo k-th vertical party model, $k \in [K]$.
h_m^k	m-th silo k-th vertical party embedding function.
$\Phi_m^{t_0}$	The set of embeddings that each party would receive at iteration t_0 within m-th silo.
Φ_m^{-k,t_0}	The set of embeddings from other vertical parties $k' \neq k$ within m-th silo (stale information at iteration t).
$\Phi_m^{k,t}$	The set of embeddings used by k-th party within m-th silo, $\Phi_m^{k,t} = \{\Phi_m^{-k,t_0}; h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m^{t_0}})\}$.
$f(\Theta)$	Global objective, $\frac{1}{N}\sum_{m=1}^{M} N_m f_m(\Theta)$.
$f_m(\Theta_m)$	m-th silo objective, $\frac{1}{N_m}\sum_{i=1}^{N_m} \mathcal{L}[\Theta_m; \mathbf{x}_m^i; y_m^i]$.

III. PROBLEM FORMULATION

To clarify our problem, we begin by presenting essential notations, as outlined in Table I. A comprehensive table of notations for proofs is available in the supplementary material.

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

Within the mth silo, the local data set $\mathbf{x}_m \in \mathbb{R}^{N_m \times J}$ is vertically partitioned across K parties (IoT devices) along the modality axis. It is noteworthy that each party k may contain a varying number of modalities, where $k \in [K]$. For simplicity, we assume that each vertical party k contains the same number of modalities, specifically (J/K) modalities per party. The ith row of $\mathbf{x}_m$ corresponds to a data sample x_m^i. For each sample

x_m^i, a party k holds a disjoint subset of features, denoted as $x_m^{k,i}$, such that $x_m^i = [x_m^{1,i}, \ldots, x_m^{K,i}]$. Each x_m^i is associated with a corresponding label y_m^i. Let $\mathbf{y}_m$ be the vector of all sample labels within the mth silo. Additionally, let $\mathbf{x}_m^k$ represent a local (partial) data set of the kth party within the mth silo, where the ith row corresponds to data features $x_m^{k,i}$.

The objective at the mth silo level is to minimize

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

where $\Theta_m = [\theta_m^0, \theta_m^1, \ldots, \theta_m^k, \ldots, \theta_m^K]$ represents the mth silo model, and $\mathcal{L}(\cdot)$ denotes a loss function that combines the embeddings $h_m^k(\theta_m^k; x_m^{k,i})$ from all vertical parties. For simplicity, we designate $k = 0$ as the edge server and define $h_m^0(\theta_m^0; x_m^i) \coloneqq \theta_m^0$ for all x^i, where $h_m^0(\cdot)$ is equivalent to the identity function. Additionally, let $h_m^k(\theta_m^k; x_m^{k,i})$ denote the embedding for the kth party. The partial derivative associated with the coordinate partition θ_m^k can be expressed as follows:

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

The stochastic partial derivative of the coordinate partition θ_m^k can be expressed as follows:

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

Here, $\mathcal{B}_m$ 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 $h_m^k(\theta_m^k; \mathbf{x}_m^{k,\mathcal{B}_m}) \coloneqq \{h_m^k(\theta_m^k; x_m^{k,\mathcal{B}_m^1}), \ldots, h_m^k(\theta_m^k; x_m^{k,\mathcal{B}_m^{B_m}})\}$ as the set of all embeddings associated with the mini-batch $\mathcal{B}_m$ on party k, where $\mathcal{B}_m^i$ denotes the ith sample in the mini-batch $\mathcal{B}_m$. We consider $\nabla_k f_{\mathcal{B}_m}(\Theta_m)$ and $\nabla_k f_{\mathcal{B}_m}[\theta_m^0, h_m^1(\theta_m^1; \mathbf{x}_m^{1,\mathcal{B}_m}), \ldots, h_m^K(\theta_m^K; \mathbf{x}_m^{K,\mathcal{B}_m})]$ equivalent and use them interchangeably. Besides, assuming that the batch size $B_m = B$ for $m \in [M]$, and the same mini-batch $\mathcal{B}_m$ is used in every Q iterations within each silo, we consider $\mathcal{B}_m^{t_0}$ and $\mathcal{B}_m$ as equivalent and use them interchangeably.

Thus, the global objective is to minimize the following:

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

where $\Theta = [\theta^0, \theta^1, \ldots, \theta^k, \ldots, \theta^K]$ denotes the global full model, and $\theta^k = (1/N) \sum_{m=1}^{M} N_m \theta_m^k$ denotes the partial model on the kth vertical party. This global objective evaluates how well the model fits the whole multimodal data set across K vertical partitions and M horizontal partitions, setting it apart from any existing HFL-type [18] or VFL-type [9] problem.

IV. HFM Algorithm

In this section, we propose a hybrid FL algorithm, named HFM, tailored for multimodal IoT systems with limited computational resources, as outlined in Algorithm 1. To elaborate, we partition the entire procedure into two components: 1) conducting VFL among K vertical parties within each silo and 2) performing HFL across M horizontal silos, as illustrated in the training timeline depicted in Fig. 4.

Algorithm 1: HFM

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**
    for $m = 1, 2, \cdots, M$ *in parallel* **do**
      for $k = 1, 2, \cdots, K$ *in parallel* **do**
        IoT device sends embedding $h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m^{t_0}})$ to the edge server;
      $\Phi_m^{t_0} \leftarrow \{\theta_m^{0,t}, h_m^1(\theta_m^{1,t}), \cdots, h_m^K(\theta_m^{K,t})\}$;
      Edge server sends $\Phi_m^{t_0}$ to all K IoT devices;
  if t (mod) $RQ = 0$ **then**
    for $k = 0, 1, \cdots, K$ *in parallel* **do**
      Global server computes $\theta^{k,t} = (1/N) \sum_{m=1}^{M} N_m \theta_m^{k,t}$;
      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**
      $\Phi_m^{k,t} \leftarrow \{\Phi_m^{-k,t_0}; h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m^{t_0}})\}$;
      $\theta_m^{k,t+1} \leftarrow \theta_m^{k,t} - \eta \nabla_k f_{\mathcal{B}_m^{t_0}}(\Phi_m^{k,t}; \mathbf{y}_m^{\mathcal{B}_m^{t_0}})$;

We additionally provide a schematic diagram of HFM in Fig. 5, which offers a clear comparison with the three methods discussed in Section II, as illustrated in Fig. 2.

A. VFL Across K Vertical Parties

At the beginning of each VFL round ($t \bmod Q = 0$) within the mth silo, designated as t_0, a mini-batch $\mathcal{B}_m^{t_0}$ is randomly sampled from $\mathbf{x}_m$. Each vertical party k, in parallel, performs block coordinate stochastic gradient descent on its local model parameters θ_m^k for Q local iterations. Specifically, for kth vertical party to compute the stochastic partial gradient with respect to its features across partial modalities, it requires the embeddings computed by all other parties $k'(k' \neq k)$, as well as its own kth party embeddings $h_m^k(\theta_m^{k,t})$. Within each silo m, these embeddings owned by IoT devices are shared with the edge server and subsequently distributed to all K parties. We define $\Phi_m^{-k,t_0} = \{h_m^{k'}(\theta_m^{k',t_0})\}_{k'=0}^{K-1}$ as the set of embeddings from other vertical parties k'; thus, the set of embeddings used by the kth party is $\Phi_m^{k,t} = \{\Phi_m^{-k,t_0}; h_m^k(\theta_m^{k,t}; \mathbf{x}_m^{k,\mathcal{B}_m^{t_0}})\}$, which inevitably contains stale information Φ_m^{-k,t_0} during $t > t_0$ in this round. For each iteration t, each party k updates θ_m^k by computing the stochastic partial derivatives $\nabla_k f_{\mathcal{B}_m^{t_0}}(\Phi_m^{k,t}; \mathbf{y}_m^{\mathcal{B}_m^{t_0}})$ and applying a gradient step with step size η. It is noteworthy that each party utilizes a stale view of the silo-level model to compute its gradient during multiple local iterations, as it

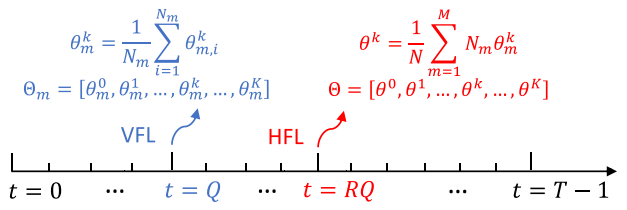


Fig. 4. Our HFM algorithm comprises: 1) Silo-level VFL across K parties in parallel for $m \in [M]$, occurring when $t \mod Q = 0$ and 2) HFL across M silos, occurring when $t \mod RQ = 0$. If we run HFM for P global rounds, i.e., $T = RQ \times P$ iterations.

reuses the embeddings received at the start of each round t_0. In Section V, we theoretically prove that HFM converges despite all IoT devices employing stale information during multiple local iterations, which may potentially hinder convergence, but it is unavoidable at this stage.

Remark 1: In addition to the inevitable use of stale information in multiple local iterations due to the limited communication overhead of HFM, another significant deviation of HFM from previous VFL algorithms, e.g., FedBCD [10], lies in its adoption of the edge server model (referred to as the head [19], denoted as θ_m^0) with trainable parameters, thereby facilitating the integration of arbitrary multimodal fusion networks. To update such a model through multiple local iterations, the parameters of the head are distributed among all involved IoT devices (vertical parties).

Remark 2: During the silo-level VFL process within HFM, kth IoT device maintains a block of partial model parameters on the kth vertical party. To compose the full silo-level model, the partial model parameters of each IoT device can be copied with the help of the edge server. This process can be performed periodically with model checkpointing.

Remark 3: Although we assume that both the edge server and all involved IoT devices (vertical parties) within each silo have a copy of the labels $\mathbf{y}_m$, we also consider solutions for scenarios where such ideal conditions may not be met in real-world IoT systems. In cases where labels are private and only available to a single party (e.g., only the edge server has the labels within each silo), the label holder can provide sufficient information for the parties to compute gradients for certain classes of model architectures [9].

B. HFL Across M Horizontal Silos

In IoT scenarios, the partial data set held by each silo (e.g., a household or a factory) is influenced by the modality type and the number of samples, resulting in significant variations in sample size and multimodal data distribution among different silos. Non-IID data sets may cause a silo-level model to fit well to its local silo data set but not necessarily to the whole data set across all M silos. To develop a global model that achieves superior performance when applied to entire multimodal IoT systems, we employ global HFL across all M silos while ensuring that it does not interfere with the VFL performed in parallel in each silo. Specifically, when $t \mod RQ = 0$ (where R is a positive integer like Q), the global server performs global model aggregation on all K parties across all M silos. For the kth party, this aggregation is represented as

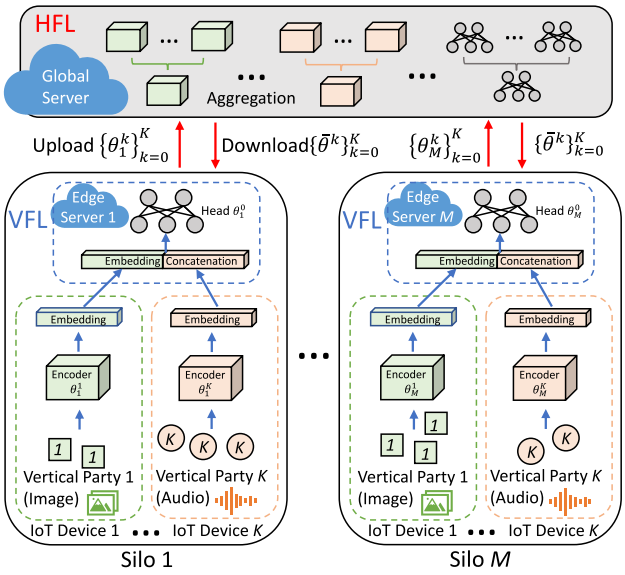


Fig. 5. Our HFM presents a novel approach compared to existing methods in Fig. 2. It thoroughly disentangles multimodal FL problem and optimizes the distribution of computing resources by employing VFL across K vertical parties (feature space) and HFL among M horizontal silos (sample space).

$\theta^{k,t} = (1/N)\sum_{m=1}^{M} N_m\theta_m^{k,t}$, and then the global server sends the updated models $\theta^{k,t}$ to all M silos. Notably, the aggregation here is based on different vertical parties, encompassing different modalities (features). In other words, a crucial distinction between our HFM and previous multimodal FL methods is that we fundamentally disentangle multimodal input rather than incrementally treating the multimodal input as a single-modal input with richer features and higher dimensions.

Remark 4: Each Q iteration is referred to as a VFL communication round, symbolizing the communication between K IoT devices and the edge server within each silo. Each RQ iteration is termed an HFL communication round, indicating the communication between M edge servers (silos) and the global server. If we execute HFM for P global rounds, i.e., $T = RQ \times P$ local iterations.

Remark 5: Although FL already provides privacy benefits by avoiding the sharing of raw data, we offer additional and more stringent solutions. Within each silo, IoT devices (vertical parties) share only embeddings and compute partial derivatives related to their local models, thereby avoiding privacy concerns caused by transmitting raw data. Moreover, we can enhance security against sophisticated attacks using methods like secure multiparty computation [20] or homomorphic encryption [21].

V. Theoretical Analysis

In this section, we discuss the theoretical analysis of the convergence of our proposed Algorithm 1.

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 1 bounds how fast the gradient and partial derivatives can change. While Assumption 1 does not directly bound the smoothness of the global objective function $f(\Theta)$, we can easily deduce this in the supplementary material.

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 2 requires that the stochastic partial derivatives computed by each vertical party k and the edge server are unbiased estimates of the full-batch partial derivatives. Assumption 2 can be satisfied in practice by ensuring that sample IDs for a mini-batch are chosen at random.

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

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

Assumption 3 bounds the variance between the stochastic partial derivatives and full-batch partial derivatives.

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 vertical parties $\mathcal{K}$ of size K

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

We note that Assumptions 2 and 3 resemble the IID assumptions in the convergence analysis of HFL. However, in the silo-level VFL within HFM, all vertical parties store identical sample IDs but different modalities (features). Thus, there is no equivalent notion of a non-IID distribution in silo-level VFL within HFM.

Now, we present the main theoretical results of the HFM convergence analysis as follows.

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

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

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

Proof: The proof is given in the supplementary material. ■

Remark 6: The convergence error in Theorem 1 arises from parallel updates on coordinate blocks in Algorithm 1, dependent on the VFL communication frequency (Q), the HFL communication frequency (RQ), the number of vertical parties (K), and the number of horizontal 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 indicate errors stemming from multiple local iterations with stale information in the VFL component and from the variance of stochastic gradients in the HFL component. We explore this further in the Theory versus Practice section, as illustrated in Figs. 9–11.

Corollary 1: When $K = 1$ (i.e., there is only one vertical party) and $Q = 1$, our proposed HFM reduces to HFL with R local iterations for M horizontal silos. Then, the average squared gradient over P global rounds is bounded

$$\begin{aligned}\frac{1}{P}\sum_{t=0}^{P-1}\mathbb{E}\left[\left\|\nabla f\left(\Theta^t\right)\right\|^2\right] &\leq \frac{2[f(\Theta^0)-f(\Theta^*)]}{\eta P} + 2\eta^2L^2\left(R^2\frac{2}{M}\frac{\sigma^2}{B}\right.\\ &\left.+\left((R-1)^2+\left(\frac{2}{M}+1\right)R^2\right)\delta^2\right)+\eta L\frac{1}{M}\left(\frac{\sigma^2}{B}+\delta^2\right)\end{aligned} \tag{11}$$

where Assumptions 3 and 4 are extended to encompass the full derivatives rather than the partial derivatives.

Remark 7: If $\eta \propto (\sqrt{M}/\sqrt{P})$ and considering a fixed R, the convergence rate in Corollary 1 degenerates to $\mathcal{O}((1/\sqrt{MP}) + (M/P))$. If P is sufficiently large to satisfy $P > M^3$, then the term (M/P) is dominated by the term $(1/\sqrt{MP})$, resulting in the convergence rate degenerating to $\mathcal{O}(1/\sqrt{MP})$, consistent with the convergence rate provided in [22] for HFL. Put differently, when HFM is reduced to HFL, it achieves linear speed-up with respect to the number of horizontal silos M.

Corollary 2: When $M = 1$ (i.e., there is only one horizontal silo) and $R = 1$, our proposed HFM reduces to VFL with Q local iterations for K vertical parties. Then, the average squared gradient over P global rounds is bounded

$$\begin{aligned}\frac{1}{P}\sum_{t=0}^{P-1}\mathbb{E}\left[\left\|\nabla f\left(\Theta^t\right)\right\|^2\right] &\leq \frac{2[f(\Theta^0)-f(\Theta^*)]}{\eta P} + 2\eta^2\sum_{k=0}^{K}L_k^2\left(Q^2(K+1)\frac{\sigma_k^2}{B}\right.\\ &\left.+Q^2(K+2)\frac{\delta^2}{K}\right)+\eta L\sum_{k=0}^{K}\left(\frac{\sigma_k^2}{B}+\frac{\delta^2}{K}\right).\end{aligned} \tag{12}$$

Remark 8: If $\eta \propto (1/\sqrt{P})$, the convergence rate in Corollary 2 degenerates to $\mathcal{O}(1/\sqrt{P})$, consistent with the convergence rate obtained in [10] for VFL. This further proves that the proposed HFM can flexibly coordinate HFL and VFL.

VI. Experiments

In this section, we conduct experiments on two publicly available multimodal data sets (Table II) to validate our HFM algorithm against three baseline methods. Notably, the data sets we selected are not confined to IoT scenarios but cover more complex, general multimodal scenarios.

A. Data Sets

MIMIC-III: Medical information mart for intensive care (MIMIC-III) data set [23] contains anonymized information of patients admitted to critical care units in a hospital. We follow the data processing steps outlined in [24] to obtain 14 681 training samples and 3236 test samples. Each sample comprises 48 time steps corresponding to 48 h, with each time step having 76 features, such as demographic information, vital signs, medications, etc. The objective is to predict in-hospital mortality (ihm task) as a binary classification task.

ModelNet40: ModelNet40 comprises images of computer-aided design (CAD) models depicting various objects [25].

TABLE II
STATISTICS OF TWO PUBLIC MULTIMODAL DATA SETS

Dataset	Train / Test	Modality	Class
MIMIC-III [23]	14,681 / 3,236	76 (features)	2 (ihm task [24])
ModelNet40 [25]	9,843 / 2,468	12 (views)	40 (objects)

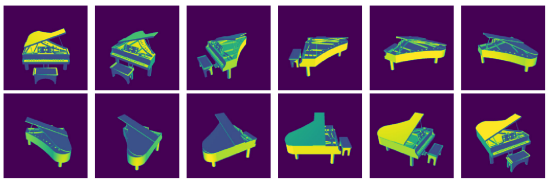


Fig. 6. ModelNet40 data set demo: Images from 12 different views corresponding to the same ID with the piano label.

Each CAD model is represented by 12 images captured from different camera views, as shown in Fig. 6. It is noteworthy that these images are not generated through data augmentation techniques, such as flipping or adding noise. Therefore, ModelNet40 is widely utilized as a multimodal data set. The data set includes 9843 CAD models in the training set and 2468 CAD models in the test set. The objective is multiclass classification with 40 classes of objects.

B. Implementation and Reproducibility

We employed an internal cluster of 48 compute nodes running CentOS 7, each with 4× 12-core 2.6 GHz Intel Xeon Gold 6126 CPUs, 1× NVIDIA Tesla V100 GPU with 32 GB HBM and 128 GB RAM, and 3× NVIDIA Tesla P100 GPUs. 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 data sets, which far exceed that of typical IoT scenarios. However, our algorithm can be deployed in multimodal IoT systems by scaling down to simpler data sets and smaller multimodal systems simultaneously.

For the MIMIC-III data set, our preprocessing procedure partitions the MIMIC-III data set into various prediction cases, with our experiments specifically targeting the prediction of in-hospital mortality (ihm). 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 (i.e., head [19]) at the edge server, which utilizes cross-entropy loss for class prediction. We utilize 5-fold cross validation for hyperparameter selection, such as performing grid search for the learning rate within the range $[0.001, 0.02]$. Due to the imbalanced nature of the MIMIC-III data set, consisting of only 16% positive samples, we assess the generalization performance on the test data set 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 data set. For the ModelNet40 data set, during the training process within each silo, we vertically partition the local data along the 12-views axis into K vertical parties (e.g., when $K = 4$, each partition contains 3 of the 12 views). Then, each device trains a ResNet18 model with a penultimate layer. The concatenated embeddings (features) are then fed into the classifier layer (i.e., head) at the edge server, which utilizes 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 $[0.0001, 0.002]$. We use top-5 accuracy as the metric to evaluate performance on the test data set. In the context of top-5 accuracy, a prediction is considered correct if any of the five highest probabilities in the model's output corresponds to the correct class label.

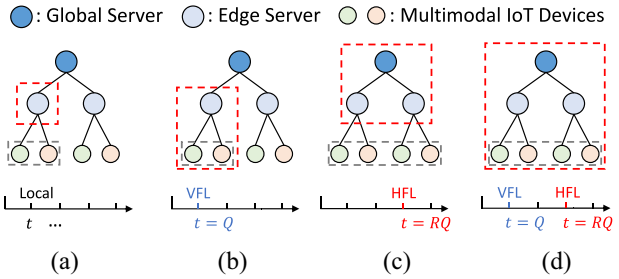


Fig. 7. Comparison of computing resources (in red), sample sizes (in gray), and training methods (along the timeline) between HFM and three baselines. (a) Local. (b) VFL. (c) HFL. (d) Our HFM.

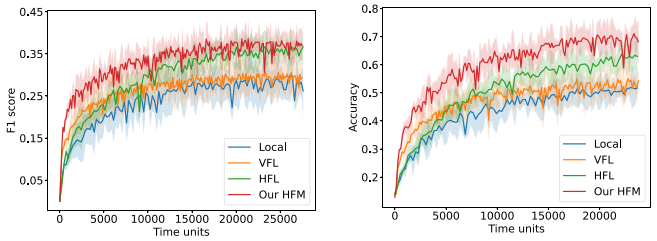
C. Comparison With Baselines

Our baseline experiments cover three categories, each potentially associated with several established multimodal FL methods. To vividly demonstrate the efficacy of our proposed HFM, we utilize a red dotted box to emphasize the computational resources allocated for each type of baseline, and a gray dotted box to indicate the sample size, along with highlighting their training differences on the timeline, as depicted in Fig. 7.

Local Training With Multimodal Data: This baseline corresponds to the traditional multimodal learning approach for IoT systems [2], [26]. 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. This bottleneck may significantly impact downstream inference performance, necessitating efficient resource allocation strategies to mitigate such limited memory or storage issues. Additionally, this baseline fails to expand the training sample space while ensuring privacy.

VFL With Multimodal Data: This baseline corresponds to a form of multimodal FL methods that explores VFL for distributed training of multimodal data [10], [27]. However, these methods do not effectively address the challenge of limited samples (within each silo) in IoT scenarios.

HFL With Multimodal Data: This baseline corresponds to a form of multimodal FL that does not involve disentangling the training of multimodal data across feature space [5], [6]. Here, the multimodal input can be perceived as a single modal input with richer information. Besides, the computing resources of



(a) (b)

Fig. 8. (a) Comparison of convergence performance on MIMIC-III. (b) Comparison of convergence performance on ModelNet40.

the edge server are limited, thus all multimodal inputs cannot be processed in parallel.

We denote t_{Vcomm} as the VFL communication latency between the edge server and IoT devices, and t_{Hcomm} as the HFL communication latency between the global server and edge servers. The local computation latency of each training step is denoted by t_{comp}. During training, we consider VFL communication between IoT devices and edge servers occurs over mobile Internet, with respective download and upload speeds of 122.74 Mbps and 10.02 Mbps (the median country speed in February 2024 [28]). HFL communication between edge servers and the global server takes place via fixed broadband, with respective download and upload speeds of 242.38 Mbps and 30.68 Mbps (the median country speed in February 2024 [28]). To simplify time accumulation to a reasonable scale, we initially set $t_{\text{Hcomm}} = 1$ time unit. Then, considering the parameter size and communication latency in our experiments, we scale $t_{\text{Vcomm}} = r$ time units and $t_{\text{comp}} = s$ time units, respectively. Under our hyperparameter setting, for the MIMIC-III data set, we reasonably scale $r = 2$ and $s = 2$ to simplify time accumulation. For the ModelNet40 data set, we reasonably scale $r = 2$ and $s = 3$ to simplify time accumulation. It is noteworthy that r and s may be influenced by various factors, including: 1) communication latency under different distances and regions (e.g., long distance may result in high latency) [29] [30] and 2) computation time on different modalities [31], and (3) fine-tuning hyperparameters, such as η. Essentially, distributed computing tends to yield benefits when communication latency has a relatively minor impact [32].

In order to fairly compare the convergence performance between HFM and three baselines, we fixed VFL communication frequency ($Q = 5$), HFL communication frequency ($RQ = 10$), vertical parties ($K = 2$), horizontal silos ($M = 5$), and repeated each experiment 10 times. As shown in the results presented in Fig. 8(a) and Table III based on the MIMIC-III data set, and in Fig. 8(b) and Table IV based on the ModelNet40 data set, it is evident that our proposed HFM and VFL baseline effectively leverage distributed computing resources from K IoT devices to accelerate convergence compared to the HFL baseline and the Local baseline, respectively. Additionally, the convergence errors of our proposed HFM and HFL baseline are smaller than those of the VFL baseline and Local baseline, respectively, attributed to the use of global HFL across M horizontal silos.

TABLE III
TIME UNITS TO ACHIEVE TARGET F_1-SCORE FOR THREE BASELINES AND OUR HFM ON MIMIC-III DATA SET

Methods	Time Units to Achieve Target F_1-Score		
	$F_1 = 0.2$	$F_1 = 0.27$	$F_1 = 0.35$
Local	5195 ± 858	13365 ± 1979	N/A
VFL	2102 ± 387	7145 ± 896	N/A
HFL	3018 ± 595	8023 ± 1048	17276 ± 1934
Our HFM	1132 ± 238	3287 ± 779	14605 ± 1213

TABLE IV
TIME UNITS TO ACHIEVE TARGET ACCURACY FOR THREE BASELINES AND OUR HFM ON MODELNET40 DATA SET

Methods	Time Units to Achieve Target Accuracy		
	$Acc = 0.4$	$Acc = 0.5$	$Acc = 0.6$
Local	6502 ± 1867	18427 ± 2479	N/A
VFL	2663 ± 845	8906 ± 2055	N/A
HFL	4647 ± 1032	7175 ± 1659	18214 ± 3058
Our HFM	1705 ± 452	3122 ± 1016	7918 ± 1565

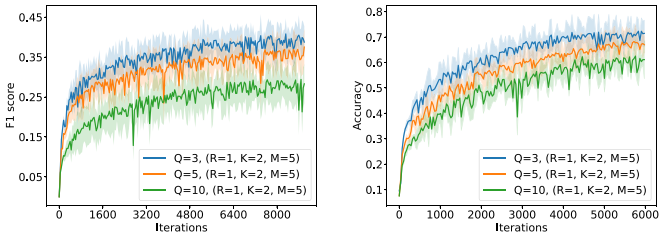
We also observed that as the accumulated time increases sufficiently, the convergence error of the HFL baseline tends to approach that of our proposed HFM. Similarly, the convergence error of the Local baseline tends to approach that of the VFL baseline. This observation is intuitive because their respective sample spaces are consistent, as illustrated in the gray dotted box in Fig. 7. The key distinction lies in our HFM approach, which optimizes the distribution of computing resources across all IoT devices and addresses the challenges posed by limited memory and storage in multimodal IoT systems through additional distributed training on these devices.

In sum, on two public multimodal data sets, our proposed HFM demonstrates improvements over the three types of baselines in terms of both convergence rate and convergence error, thereby making it practical for IoT scenarios that require rapid and accurate downstream inference tasks, such as classification, prediction, etc.

D. Theory Versus Practice

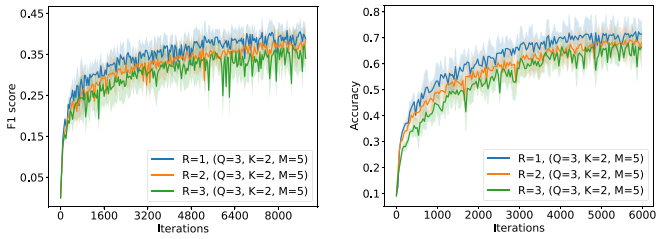
In this part, we conduct extensive experiments (ablation study) to verify our theoretical analysis in Section V. Specifically, we explored variations in the communication frequency of VFL and HFL, as well as the number of vertical parties in HFM, within the constraints of real-world multimodal IoT systems. For example, this may involve conducting multiple local iterations with limited communication costs or performing VFL among predetermined IoT devices (vertical parties). It is noteworthy that we have not included extra experimental results regarding the number of horizontal silos (M). This is because when we fix the total number of samples N, changing M will also alter the Non-IID degree, thus making it difficult to fairly observe the impact on convergence performance.

The Impact of VFL Communication Frequency: From Fig. 9(a) and (b), it is observed that as the value of Q increases (indicating less frequent VFL communication), the convergence error increases when the number of iterations is fixed, as demonstrated in Theorem 1. However, the convergence rate



(a) (b)

Fig. 9. (a) Impact of Q on MIMIC-III. (b) Impact of Q on ModelNet40.



(a) (b)

Fig. 10. (a) Impact of R on MIMIC-III. (b) Impact of R on ModelNet40.

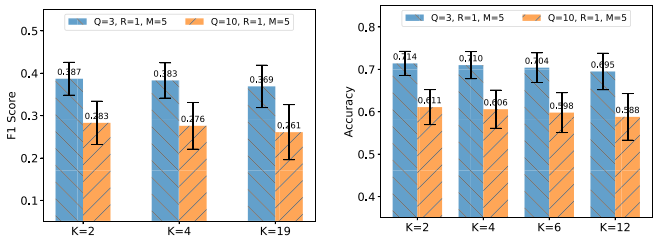
improves when we fix the number of VFL communication rounds, which is intuitive, as devices (within each silo) can train more with a larger value of Q between VFL communications. Thus, by appropriately increasing Q with a suitable learning rate η, we can improve communication efficiency by reducing the total number of VFL communication rounds required for a given level of performance.

The Impact of HFL Communication Frequency: From Fig. 10(a) and (b), it is observed that as the value of R increases (indicating less frequent HFL communication when Q is fixed), the convergence error increases when the number of iterations is fixed, as demonstrated in Theorem 1. However, the convergence rate improves if we fix the number of HFL communication rounds, which is intuitive, as devices (across all silos) can train more with a larger value of R between HFL communications. Thus, by appropriately increasing R with a suitable learning rate η, we can enhance communication efficiency by reducing the total number of HFL communication rounds required for a given level of performance.

The Impact of the Number of Vertical Parties: From Fig. 11(a) and (b), it is observed that as the value of K decreases (indicating fewer vertical parties with a fixed number of modalities), both the convergence error and variance tend to decrease slightly. This observation aligns with intuition and the theoretical analysis 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 total number of vertical parties is typically not very large [33].

VII. Conclusion

In conclusion, to address the first question posed in Section I, we propose a hybrid FL algorithm, named HFM, specifically designed for multimodal IoT systems with constrained computational resources. HFM uniquely combines



(a) (b)

Fig. 11. (a) Impact of K on MIMIC-III. (b) Impact of K on ModelNet40.

VFL and HFL paradigms to distribute computing resources across feature and sample spaces simultaneously. To tackle the second question raised in Section I, we theoretically prove that the convergence of HFM depends on the communication frequency of VFL and HFL, as well as the number of vertical partitions and horizontal partitions. Furthermore, we empirically demonstrate that HFM outperforms three types of baselines in terms of both convergence rate and convergence error based on two public multimodal data sets, thereby making it practical for multimodal IoT systems that require rapid and accurate downstream inference, such as classification, prediction, etc. In future work, we aim to explore the potential of asynchronous settings due to issues with heterogeneous modalities or heterogeneous IoT devices in multimodal IoT systems.

References

[1] A. K. Singh, D. Kundur, and M. Conti, "Introduction to the special issue on integrity of multimedia and multimodal data in Internet of Things," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 20, no. 6, pp. 1–4, 2024.

[2] Y. Huang, C. Du, Z. Xue, X. Chen, H. Zhao, and L. Huang, "What makes multi-modal learning better than single (provably)," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 10944–10956.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[4] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 1–24, Jan. 2022.

[5] Y. Zhao, P. Barnaghi, and H. Haddadi, "Multimodal federated learning on IoT data," in *Proc. IEEE/ACM 7th Int. Conf. Internet Things Design Implement.*, 2022, pp. 43–54.

[6] B. Xiong, X. Yang, F. Qi, and C. Xu, "A unified framework for multimodal federated learning," *Neurocomputing*, vol. 480, pp. 110–118, Apr. 2022.

[7] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[8] T. Feng et al., "FedMultimodal: A benchmark for multimodal federated learning," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2023, pp. 4035–4045.

[9] Y. Liu et al., "Vertical federated learning: Concepts, advances, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, Jul. 2024.

[10] Y. Liu et al., "FedBCD: A communication-efficient collaborative learning framework for distributed features," *IEEE Trans. Signal Process.*, vol. 70, pp. 4277–4290, Aug. 2022.

[11] P. Liu, G. Zhu, W. Jiang, W. Luo, J. Xu, and S. Cui, "Vertical federated edge learning with distributed integrated sensing and communication," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2091–2095, Sep. 2022.

[12] J. Wang et al., "TVFL: Tunable vertical federated learning towards communication-efficient model serving," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
[13] C. Yu, S. Shen, S. Wang, K. Zhang, and H. Zhao, "Efficient multi-layer stochastic gradient descent algorithm for federated learning in E-health," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 1263–1268.
[14] L. Su and V. K. Lau, "Hierarchical federated learning for hybrid data partitioning across multitype sensors," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10922–10939, Jul. 2021.
[15] A. Das, T. Castiglia, S. Wang, and S. Patterson, "Cross-silo federated learning for multi-tier networks with vertical and horizontal data partitioning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 6, pp. 1–27, 2022.
[16] X. Ouyang et al., "Harmony: Heterogeneous multi-modal federated learning through disentangled model training," in *Proc. 21st Annu. Int. Conf. Mobile Syst., Appl. Services*, 2023, pp. 530–543.
[17] J. Chen and A. Zhang, "FedMSplit: Correlation-adaptive federated multi-task learning across multimodal split networks," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2022, pp. 87–96.
[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[19] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2089–2099.
[20] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6103–6115, Nov. 2022.
[21] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, 2018.
[22] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5693–5700.
[23] A. E. Johnson et al., "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, pp. 1–9, 2016.
[24] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Sci. Data*, vol. 6, no. 1, p. 96, 2019.
[25] Z. Wu et al., "3D Shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
[26] Z. Huang, X. Xu, J. Ni, H. Zhu, and C. Wang, "Multimodal representation learning for recommendation in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10675–10685, Dec. 2019.
[27] M. Gong et al., "A multi-modal vertical federated learning framework based on homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1826–1839, 2024.
[28] "United states median country speeds." Feb. 2024. [Online]. Available: https://www.speedtest.net/global-index/united-states
[29] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.
[30] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2021.
[31] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
[32] Z. Ning et al., "Distributed and dynamic service placement in pervasive edge computing networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 6, pp. 1277–1292, Jun. 2021.
[33] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.