Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

FedHyperM: Modality-Sensitive Federated Hypergraph Learning for Heterogeneous Multimodal Edge Intelligence

Qibin Zhou

qibinzhou99@163.com

East China Normal University

Jianqi Shi

East China Normal University

Boon Xian Chai

Swinburne University of Technology - Hawthorn Campus: Swinburne University of Technology

Rifai Chai

Swinburne University of Technology - Hawthorn Campus: Swinburne University of Technology

Xin Li

YSTC Energy Pty Ltd

Di Guo

Hangzhou Dianzi University Information Engineering

Research Article

Keywords: Edge intelligence, Federated multimodal learning, Modality heterogeneity, Hypergraph learning

Posted Date: July 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10102389/v1

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

FedHyperM: Modality-Sensitive Federated Hypergraph Learning for Heterogeneous Multimodal Edge Intelligence

Qibin Zhou[1], Jianqi Shi[1*], Boon Xian Chai[2], Rifai Chai[2], Xin Li[3], Di Guo[4]

[1]Software Engineering Institute, East China Normal University, Shanghai, China.
[2]School of Engineering, Swinburne University of Technology, Melbourne, Victoria, Australia.
[3]YSTC Energy Pty Ltd, Melbourne, Victoria, Australia.
[4]Department of Computer Science and Technology, Hangzhou Dianzi University Information Engineering College, Hangzhou, Zhejiang, China.

*Corresponding author(s). E-mail(s): jqshi@sei.ecnu.edu.cn;

Abstract

Federated learning enables edge devices to collaboratively train models in edge–cloud environments without sharing raw sensor data. In practical edge deployments, however, devices often have access to different subsets of modalities due to hardware heterogeneity, varying sensing conditions, or temporary sensor failures. Such modality heterogeneity makes conventional federated aggregation unreliable, as parameters learned from different modality evidence may not be directly comparable. Meanwhile, this setting also raises privacy concerns, since image, depth, and signal data may reveal sensitive user or environmental information. To address these challenges, we propose FedHyperM, a privacy-preserving federated multimodal learning framework for edge devices. FedHyperM performs hypergraph-based multimodal prediction by treating modality-specific features as nodes and adaptively constructing hyperedges to capture high-order correlations among multimodal features. On the cloud server, FedHyperM further introduces modality-sensitive aggregation (MSA), which uses the minimum edit distance (MinED) between available modality names to weight uploaded hypergraph learning parameters during aggregation. Experiments on the UR-FALL dataset under different modality existence rate (MER) settings demonstrate that

FedHyperM consistently outperforms representative federated multimodal baselines, including FedAvg, FedCor, FedProx, AutoFed, and FedInMM. Compared with the five baselines, FedHyperM improves average accuracy by 3.64% to 22.33% and average F1-score by 2.91% to 26.82%.

Keywords: Edge intelligence, Federated multimodal learning, Modality heterogeneity, Hypergraph learning

1 Introduction

Over the past decade, rapid advances in Internet of Things (IoT) technologies have led to the proliferation of edge devices, including smart sensors and mobile phones, which continuously generate massive volumes of data across diverse application domains [1, 2]. At the same time, deep learning (DL) models have become essential for many edge-deployed applications, such as face recognition [3–5], mobile health monitoring [6], smart homes [7] and industrial fault diagnosis [8, 9], where large-scale training data are often required to achieve satisfactory performance. Edge computing (EC) has therefore emerged as a promising paradigm for integrating IoT with artificial intelligence by moving computational resources closer to data sources, thereby substantially reducing communication overhead compared with conventional cloud-centric architectures [10]. Nevertheless, effectively exploiting the abundant data produced in edge environments to improve DL model performance, while simultaneously preserving data privacy, remains a fundamental challenge for EC systems [11, 12].

Federated learning (FL) provides an effective paradigm for enabling collaborative model training in EC systems without requiring raw data to be centralized [13, 14]. In a typical FL workflow, each edge device trains a local model on its private dataset and transmits only model updates to a cloud server, where the received updates are aggregated to produce a shared global model. By retaining raw data locally, FL helps mitigate privacy risks while leveraging distributed data resources to optimize DL models. Existing studies have investigated various aspects of FL, including communication-efficient aggregation [13], optimization under non-IID data distributions [14], and resource-aware device selection [15].

However, these methods remain insufficient for practical edge scenarios, where edge devices are often equipped with heterogeneous sensors and thus observe different combinations of modalities. For example, emergency response scenarios, such as earthquakes or floods, may require the joint analysis of on-site images or videos, social media reports, and audio distress signals [16]. In contrast, hospital edge environments typically rely on distinct modality combinations, such as text-based medical records and CT images [17]. Such modality heterogeneity leads local models to learn from different sources of evidence, making their uploaded parameters difficult to compare directly, even when the models are optimized for the same task. Consequently, conventional aggregation strategies, which typically assume a shared input space or compatible model updates, may introduce aggregation noise by indiscriminately combining parameters learned from inconsistent modality information [18, 19].

To address these challenges, we propose FedHyperM, a collaborative federated multimodal learning framework for edge devices under modality heterogeneity. FedHyperM first initializes a multimodal feature extractor locally on each edge device and freezes the extractor during collaborative learning. It then performs hypergraph-based multimodal prediction by treating modality-specific features as nodes and adaptively constructing hyperedges to capture high-order correlations among multimodal features. After local training, each edge device uploads its hypergraph learning parameters and available modality names to the cloud server. The server then applies a modality-sensitive aggregation (MSA) mechanism, where minimum edit distance (MinED) is used as an aggregation weight measure.

The main contributions of this work are:

- We propose FedHyperM, a novel collaborative framework to facilitate federated multimodal learning across edge devices under modality heterogeneity.
- We design a hypergraph-based multimodal prediction module and an MSA mechanism to capture high-order correlations among multimodal features and aggregate hypergraph learning parameters according to MinED-based aggregation weights.
- We conduct extensive experiments on the UR-FALL dataset under different modality existing rate (MER) settings and compare FedHyperM with five baselines. Experimental results show that FedHyperM consistently improves accuracy and F1-score, and ablation studies verify the effectiveness of hypergraph and MSA.

2 Related Work

Federated Learning for Edge Devices. FL is widely used to train models across distributed edge devices without collecting raw data on a cloud server. FedAvg [13] defines the standard procedure: each edge device trains a local model and the server aggregates uploaded model parameters. Many later methods improve this procedure under heterogeneous edge conditions. FedProx [14] adds a proximal term to reduce the drift of local models under non-IID data. FedCor [15] selects informative edge devices to accelerate convergence. Other studies improve aggregation or topology design in edge environments [20, 21]. These studies improve FL under data, optimization, or system heterogeneity, but they usually assume that uploaded parameters are learned from comparable input spaces.

Multimodal FL extends this setting by allowing each edge device to learn from multiple sensing modalities. FedAvg-2 [22] study multimodal FL on IoT data, and FedMultimodal [23] provides a benchmark for missing modalities, missing labels, and noisy labels. AutoFed [24] addresses multimodal heterogeneity through modality imputation. These methods improve multimodal FL through data completion, device selection, or feature weighting. However, they rarely consider the aggregation of hypergraph learning parameters uploaded by edge devices with different available modality sets.

Modality Heterogeneity and Incomplete Modalities. Modality heterogeneity means that different edge devices may own different modality sets. This problem is related to incomplete-modality learning, where some modalities are absent

during training or inference. In federated settings, FedMSplit [25] uses correlation-adaptive multi-task learning across multimodal split networks. FedMBridge [26] learns bridgeable representations for edge devices with different modality types, while FedInMM [27] uses dynamic modality weighting for incomplete modalities. These studies show that modality absence affects both local training and global aggregation. However, many of them treat missing modalities as information to be reconstructed or aligned. Therefore, it is important to study federated multimodal learning methods that can directly operate with different available modality sets on edge devices.

Hypergraph Modeling for Multimodal Collaboration. Hypergraph learning has been widely studied for representing data relations that go beyond pairwise dependencies. By using hyperedges to associate groups of nodes, hypergraphs can encode group-level relations among features or samples [28, 29]. In multimodal prediction, such group-level relations describe cross-modal correlations among modality-specific features, such as image, depth, and signal features. Representative methods have explored this idea in different settings. VCHM [30] modeling constructs sparse hypergraph relations for visual classification. FedRel [20] learns spatial-temporal graph relevance in federated environments. STHL [31] exploits spatial-temporal data for physiological signal recognition. LFH [32] further learns hypergraph structures from heterogeneous data. These studies support adaptive hypergraph construction over modality-specific features, but they do not address the aggregation of hypergraph learning parameters across edge devices with heterogeneous modality availability.

3 Preliminaries

3.1 Hypergraph Learning

A hypergraph is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$, where $\mathcal{V} = \{v_1, ..., v_n\}$ denotes the node set and $\mathcal{E} = \{e_1, ..., e_m\}$ denotes the hyperedge set, with n and m being the numbers of nodes and hyperedges, respectively. Each hyperedge $e \in \mathcal{E}$ connects a subset of nodes in $\mathcal{V}$, i.e, $e = \{v_i, ..., v_j\} \subseteq \mathcal{V}$. The diagonal matrix $\mathcal{W} \in \mathcal{R}^{m \times m}$ specifies the weights of the hyperedges, with $diag(\mathcal{W}) = [w(e_1), ..., w(e_M)]$. The relationship between nodes and hyperedges is usually represented by incidence matrix $H \in \mathcal{R}^{n \times m}$, with entries are continuous value in $[0, 1]$. If node v_j is incident to hyperedge e_i, then $H(v_j, e_i) \geq 0$; otherwise, $H(v_i, e_j) = 0$. The value $H(v_i, e_j)$ can be interpreted as the importance of node v_j to hyperedge e_i. Based on this formulation, the degree of each node $v_j \in \mathcal{V}$ and the degree of each hyperedge $e_i \in \mathcal{E}$ are defined as follows:

$$d(v_j) = \sum_{e \in \mathcal{E}} w(e) \times H(v_j, e) \tag{1}$$

$$d(e_i) = \sum_{v \in \mathcal{V}} H(v, e_i) \tag{2}$$

3.2 Problem Definition

As illustrated in Figure 1, we consider an edge-cloud system consisting of a central server and a set of edge devices, denoted by $\mathcal{C} = \{c_i\}_{i=1}^{N}$, where N is the the total

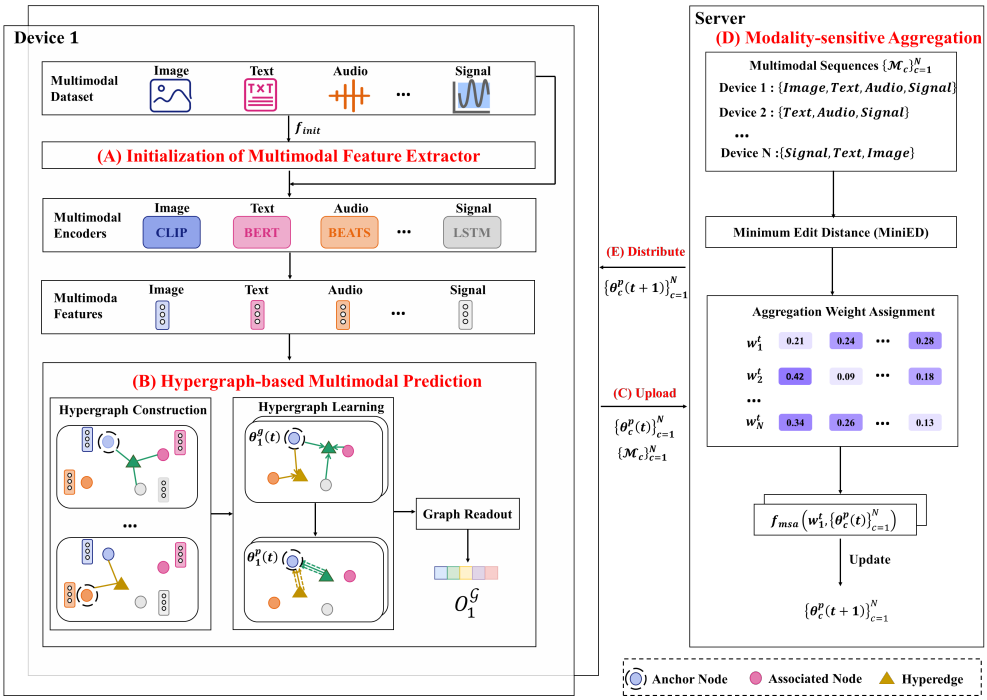


Fig. 1: The overview of our proposed framework.

number of devices. Each device owns a local dataset with heterogeneous modalities, denoted by $\mathcal{X} = \{X_c^m\}_{m=1}^{M_c}$, where $X_c^m \in \mathcal{R}^{D \times d_m}$ represents the raw data of modality m, and D denotes the size of local dataset. Each edge device host a local model $\theta_c \in \Theta$, which incorporates a feature extractor and a predictor, defined as $\theta_c = \{\theta_c^e, \theta_c^p\}$.

The edge devices collaboratively aim to enhance the performance of the on-device models $\{\theta_c\}_{c \in \mathcal{C}}$ through collaborative learning, while while preserving the privacy of local dataset. The overall objective function to be minimized over the collaborative learning process is defined as:

$$\min_{\{\theta_c\}_{c \in \mathcal{C}}} \mathcal{L}(\{\theta_1, ..., \theta_N\}; \{X_1, ..., X_N\}) \tag{3}$$

Here, we consider multimodal heterogeneity across edge devices, denoted by $\exists i, j \in [1, ..., N], M_i \neq M_j$, where M_i represents the number of modalities available on edge device i.

4 Methodology

Figure 1 illustrates an overview of the FedHyperM architecture. We elaborate on the proposed framework in this section by formulating the three key processes of FedHyperM: (1) extracting high-quality modality-specific features from multimodal raw data; (2) adaptively constructing hypergraphs over multimodal features on the

edge side to capture high-order correlations through hypergraph learning; and (3) applying a modality-sensitive aggregation (MSA) mechanism on the server side to mitigate performance degradation caused by multimodal heterogeneity.

4.1 Initialization of Multimodal Feature Extractor

As illustrated in the part (A) of Fig 1, each edge device first pre-trains the feature extractor on its local dataset, defined as $f_{init}(\theta_c^e; \mathcal{X})$. The multimodal feature extractor can be implemented as a set of modality-specific encoders, denoted by $\theta_c^e = \{E^m\}_{m \in \mathcal{M}}$, where $\mathcal{M}$ represents the set of modalities. Specifically, BERT [33] is used to the text modality, CLIP [34] for the image modality, LSTM [35] for the signal modality, and BEATS [36] for the audio modality. The multimodal feature extraction process can be defined as:

$$\tilde{\mathcal{X}}_c = f_e(\mathcal{X}_c; \theta_c^e) \tag{4}$$

$\tilde{\mathcal{X}}_c = \{\tilde{X}_c^m\}_{m \in \mathcal{M}} \in \mathcal{R}^{|\mathcal{M}_c| \times d_m}$ denotes the extracted high-quality multimodal features, where $\mathcal{M}_c$ represents the set of modalities available on edge device c, and d_m denotes the unified feature embedding dimension. Notably, the feature extractor is executed locally on each edge device. It is first pre-trained on the local multimodal dataset before collaborative learning and then frozen during the collaborative learning process.

4.2 Hypergraph-based Mulitimodal Prediction

Adaptive Hypergraph Construction over Multimodal Features The special structure of hyperedges enables hypergraphs to capture high-order data relationships [31]. Since multimodal features $\tilde{\mathcal{X}}$ do not naturally contain explicit nodes, we treat each modality-specific feature as a node on each edge device. By grouping nodes with shared attributes or strong relationships into a single hyperedge, hypergraphs can effectively capture local group-level information among nodes [20]. Therefore, identifying related nodes for different hyperedges is essential for hyperedge construction. To facilitate this process, we define two types of nodes: anchor nodes and associated nodes. A anchor node $\dot{v}_c \in \mathcal{V}$ serves as an anchor for hyperedge construction and dynamically connects to a set of associated nodes $S\{\bar{v}_c\}$ to form a hyperedge. The set of associated nodes $S(\bar{v})$ is united with the anchor node $\dot{v}_c$ to represent a complete data relation within a hyperedge.

To effectively uncover potential high-order relationships across modalities, we propose a dynamic learning process that constructs a hypergraph over multimodal features. Specifically, for each anchor node $\dot{v}_c \in \mathcal{V}$, which represents a modality, the hyperedge $e(\dot{v}_c) \in \mathcal{E}$ is generated by reconstructing the anchor node $\dot{v}_c$ and connecting it with candidate associated nodes $\bar{S}(\dot{v}_c) = \{\bar{v}_c | \bar{v}_c \in \mathcal{V}, \bar{v}_c \neq \dot{v}_c\}$ that representing other modalities. This process can be mathematically formulated as follows:

$$l(\dot{v}_c) = \|\tilde{\mathcal{X}}_c(\dot{v}_c) \cdot \theta_p(\dot{v}_c) - \theta_{\dot{v}_c} \cdot \tilde{\mathcal{X}}_c(\bar{S}(\dot{v}_c))\|_2 \tag{5}$$

where $\|\cdot\|_2$ represents the $l2$ norm of the vector. $\tilde{\mathcal{X}}_c(\dot{v}_c)$ and $\tilde{\mathcal{X}}_c(\bar{S}(\dot{v}_c))$ denote the node feature matrices of the anchor node $\dot{v}_c$ and the candidate associated node set, respectively. $\theta_p(\dot{v}_c) \in \mathcal{R}^{d_m \times d}$ denotes a learnable projection matrix specific to the

generation of the hyperedge $e(\dot{v}_c) \in \mathcal{E}$. $\theta_{\dot{v}_c} \in \mathcal{R}^{|\mathcal{M}_c|-1}$ represents the trainable reconstruction coefficient vector for a specific spatial hyperedge, where each element $\theta_{\dot{v}_c}(\bar{v}_c)$ denotes the learned reconstruction coefficient of the node $\bar{v}_c \in \bar{S}(\dot{v}_c)$. Based on $\theta_{\dot{v}_c}$, the hyperedge corresponding to the anchor node $\dot{v}_c$ is constructed by selecting candidate associated nodes $S(\dot{v}_c) = \{\bar{v}_c | \bar{v}_c \in \bar{S}(\dot{v}_c), \theta_{\dot{v}_c}(\bar{v}_c) > 0\}$ whose reconstruction coefficient are positive. In contrast, unselected nodes with a reconstruction coefficients below zero are connected by dotted lines.

The reconstruction error $l(\dot{v}_c)$ measure the deviation of the reconstruction of the anchor node. In practice, we impose both $l1$ norm and $l2$ norm on $\theta_{\dot{v}_c}$ simultaneously to balance the number of selected associated nodes and control the impact of outliers that may lead to non-smooth results [30, 37]. Overall, the loss of the hypergraph generation process is defined as:

$$\mathcal{L}_r = \sum_{\dot{v} \in \mathcal{V}} l(\dot{v}_c) + \alpha \|\theta_{\dot{v}_c}\|_1 + \beta \|\theta_{\dot{v}_c}\|_2 \tag{6}$$

$\|\cdot\|_2$ denotes the $l1$ norm of a vector. α and β are hyperparameters that trade off $l1$ norm and $l2$ norm regularization terms of the coefficient vector $\theta_{\dot{v}_c}$. The overall hypergraph construction process is defined as $f_{hgc}(\tilde{\mathcal{X}})$.

Hypergraph Learning for Prediction Let H represents incidence relations between hyperedges and nodes, with entries defined as follows:

$$H\left(v_c, e^k\left(\dot{v}_c\right)\right) = \begin{cases} 1 & v_c = \dot{v}_c \\ \theta_{\dot{v}_c}(v_c) & v_c \in e\left(\dot{v}_c\right), v_c \neq e^k\left(\dot{v}_c\right) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The embedding of hyperedges is aggregated by node features as follows:

$$E(e\left(\dot{v}_c\right)) = \frac{\sum_{v_c \in \mathcal{V}} H(v_c, e\left(\dot{v}_c\right)) \times \tilde{\mathcal{X}}(v_c)}{\sum_{v_c \in \mathcal{V}} H(v_c, e\left(\dot{v}_c\right))} \tag{8}$$

After that, we employ a multi-head attention mechanism to better capture correlations across nodes in the generated hypergraph, where hyperedges associated with each anchor node are used to update the anchor node embedding. The hyperedge associated with node $\dot{v}_c$ is represented as $\mathcal{E}(\dot{v}_c)$. Fig 1 part (B) illustrates the update process for a anchor node $\dot{v}_c$ with two associated hyperedges $e_1(\dot{v}_c)$ and $e_2(\dot{v}_c)$: we first compute the multi-head attention between $\dot{v}_c$ and $e_1(\dot{v}_c)$, and then use the normalized attention score as the weight of $e_1(\dot{v}_c)$. The calculation involves Eq 9 to Eq 13.

$$Q^h(\dot{v}_c) = \Theta_c^q(h) \cdot \tilde{\mathcal{X}}(\dot{v}_c) \tag{9}$$

where the dimension of $\tilde{\mathcal{X}}(\dot{v}_c)$ is $\mathcal{R}^d$, and $\Theta_c^q(h) \in \mathcal{R}^{\frac{d}{P} \times d}$ denotes the trainable matrix on the h-th attention head, $h \in [1, P]$. P is the number of attention heads. The anchor node is projected to generate the h-th query vector $Q^h(\dot{v}_c)$ by $\Theta_c^q(h)$. Similarly, we

also generate key vector $K^h(\dot{v}_c)$ based on hyerpedge $e_1(\dot{v}_c)$ as:

$$K^h(e_1(\dot{v}_c) = \Theta_c^k(h) \cdot E(e_1(\dot{v}_c) \tag{10}$$

The dimension of $\Theta_c^k(h)$ is $\mathcal{R}^{\frac{d}{P} \times d}$. The attention value of h-th head between $\dot{v}_c$ and $e_1(\dot{v}_c)$ are derived as follows:

$$A^h(\dot{v}_c, e_1(\dot{v}_c)) = (Q^h(\dot{v}_c) \cdot \Theta_c^a(h) \cdot K^h(e_1(\dot{v}_c))) \cdot \frac{1}{\sqrt{d}} \tag{11}$$

The attention matrix $\Theta_c^a(h) \in \mathcal{R}^{\frac{d}{P} \times \frac{d}{P}}$ represent the learnable correlation information across multimodalities. Moreover, since since large magnitudes of K and Q can substantially increase the attention scores and eventually cause gradient explosion [38], we divide the attention scores by $\sqrt{d}$ to stabilize the training process.

Following this, the calculated attention score $A^h(\dot{v}_c, e_1(\dot{v}_c))$ dynamically quantifies the importance of hyperedge $e_1(\dot{v}_c$ to the anchor node $\dot{v}_c$ under the h-h attention head. The weight of hyperedge $e_1(\dot{v}_c$ is obtained as follows:

$$\bar{w}_g(\dot{v}_c, e_1(\dot{v}_c)) = \Big\Vert_{h \in [1,P]} A^h(\dot{v}_c, e_1(\dot{v}_c)) \tag{12}$$

$\Big\Vert$ represents concatenation, and the dimension of weight vector $\bar{w}_g(\dot{v}_c, e_1(\dot{v}_c))$ is $\mathcal{R}^P$, stacked by attention score on each head. To obtain the final weight vector for each hyperedge, we aggregate the weight values from all related hyperedges and apply the softmax function as follows:

$$w_g\left(\dot{v}_c, e\right) = \underset{\forall e \in \mathcal{E}(\dot{v}_c)}{Softmax}\left(\bar{w}_g\left(\dot{v}_c, e\right)\right) \tag{13}$$

where $w_g\left(\dot{v}_c, e\right) \in \mathcal{R}^{P \times |\mathcal{E}(\dot{v}_c)|}$, with elements are normalized among all hyeredges per head. The attentive aggregation of head h for node $\dot{v}_c$ is calculated as follows:

$$o^h(\dot{v}_c) = \sum_{e \in \mathcal{E}(\dot{v}_c)} \left(w_g^h\left(\dot{v}_c, e\right) \times K^h(e)\right) \tag{14}$$

$w_g^h\left(\dot{v}_c, e\right)$ denotes the scalar entry of h-head in the weight vector $w_g\left(\dot{v}_c, e\right)$ of hyperedge e, and the dimension of $o^h(\dot{v}_c)O^h(\dot{v}_c)$ is $\mathcal{R}^{\frac{d}{P}}$. After concatenating all P heads, the updated node embedding of $\dot{v}_c$ is obtained through a shallow multi-layer perceptron (MLP), denoted by:

$$O_c^{\mathcal{V}}(\dot{v}_c) = MLP(\Big\Vert_{h \in [1,P]}) o^h(\dot{v}_c) \tag{15}$$

We average the node embedding of all nodes $O_c^{\mathcal{V}}$ to read-out the graph representation $O_c^{\mathcal{G}} \in \mathcal{R}^d$. The loss function used by each edge device during local training is defined as follows:

$$\mathcal{L} = \gamma \mathcal{L}_r + (1 - \gamma) CE(MLP(O_c^{\mathcal{G}}, lb)) \tag{16}$$

The MLP maps the graph representation to the label space, while the cross-entropy loss (CE) measures the discrepancy between the predicted and ground-truth labels lb. γ is a weight hyperparameter to trade off the effects of reconstruction loss and the loss of supervised local training. The overall hypergraph learning process is defined as $f_{hgl}(\mathcal{G}_c; \theta_c^p)$.

4.3 Modality-sensitive Aggregation

upon completion of on-device local training, each edge device uploads its hypergraph learning parameters, $\theta_c^p = \{\Theta_c^q, \Theta_c^k\}$, together with the sequence of available modality names, $\mathcal{M}_c$, to the server. To mitigate performance degradation caused by divergence across on-device modalities [39], we propose an MSA process based on minimum edit distance (MinED) [40], which serves as an aggregation weight measure to identify informative model parameters from neighboring edge devices. A larger sum of pairwise MinED scores among neighboring edge devices indicates greater modality diversity. The normalized model parameter aggregation for edge device c is then computed as follows:

$$w_{\dot{c}}(\bar{c}) = \frac{MinED(\mathcal{M}_{\dot{c}}, \mathcal{M}_{\bar{c}})}{\sum_{c \in \mathcal{C}} MinED(\mathcal{M}_{\dot{c}}, \mathcal{M}_c)} \tag{17}$$

We assign aggregation weight matrix w_c to each edge device , $c \in \mathcal{C}$ and aggregate the uploaded model parameters accordingly. This process is defined as $f_{msa}(\{\theta_c^p\}_{c \in \mathcal{C}})$

4.4 Overall Training Process

The overall collaborative training process of FedHyperM is described in Algorithm 1. The workflow of FedHyperM proceeds as follows:

1. At the beginning of the collaborative learning process, the multimodal feature extractors are initialized on each edge device.
2. In the t-th communication round, each device encodes multimodal features, performs hypergraph-based multimodal prediction, and then uploads the parameters learned during hypergraph training to the server.
3. The server aggregates the uploaded parameters using a modality-sensitive aggregation mechanism and then distributes the updated parameters to each device.
4. Each device receives the aggregated parameters from the server and updates its local hypergraph learning model accordingly.

5 Experimental Designs

5.1 Research Questions

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed framework from different perspectives. We aim to answer the following research questions:

- **RQ1:** How does our proposed FedHyperM perform in terms of model performance compared to the five baselines?

Algorithm 1 FedHyperM

Input: The number of device N, the number of communication round T, the set of local dataset across devices $\mathcal{X}$, and the set of local models across edge devices $\{\theta_c\}$

The multimodal feature extractor at each edge device is initialized as $f_{init}(\theta_c^e; \mathcal{X}_c)$.
for each round $t \in [1, 2, ..., T]$ **do**
  // Edge device side.
  for each device $c \in \mathcal{C}$ (in parallel) **do**
    $\tilde{\mathcal{X}}(t) \leftarrow f_e(\mathcal{X}_c; \theta_c^e(t))$ ▷ Multimodal feature extraction
    The hypergraph is dynamically constructed from the multimodal features using $\mathcal{G}_c(t) \leftarrow f_{hgc}(\tilde{\mathcal{X}}(t))$
    Hypergraph learning for prediction is performed using $f_{hgl}(\mathcal{G}_c(t); \theta_c^p(t))$
    upload $\theta_c^p(t)$ to the server
  end for
  // Server side.
  Assign aggregation weights by Eq.17 ▷ Modality-sensitive model aggregation
  Update $\tilde{\theta}_c^p(t)$ by $f_{msa}(\{\{\theta_c^p\}_{c\in\mathcal{C}}\})$
  Distribute $\tilde{\theta}_c^p(t)$ to each edge device c
  // Edge device side.
  for each device $c \in \mathcal{C}$ (in parallel) **do**
    Update $\theta_c^p(t+1))$ by $\tilde{\theta}_c^p(t)$
  end for
end for

Name	Setup
Loss function	Cross entropy loss
Optimiser	Adam [41]
Learning rate	1.5e-3
Dropout rate	0.3
Weight initialiser	Xavier [42]
Communication round	150
Graph readout func	2 linear layers, 1 output layer

Table 1: Shared parameter setup in FedHyperM.

- **RQ2:** What is the impact of key hyperparameters β and α on the FedHyperM performance?
- **RQ3:** How do different components, such as and affect the performance of FedHyperM?

5.2 Experimental Setup

We assume an edge-cloud environment comprising one cloud server and $E = 3$ edge devices.

1) Dataset: To evaluate the proposed framework, we conduct experiments on the UR-FALL dataset [5], which comprises 11,544 samples extracted from 70 RGB-depth

video sequences. Each sample includes three modalities and is annotated for three-class fall detection: not lying, lying on the ground, and transitional poses.

2) Data Partition: In our work, we simulate modality heterogeneity by assuming that the availability of each modality $n \in \mathcal{N}$ follows an independent Bernoulli distribution $Bernoulli(\rho_n)$ [39]. Here, we use the modality existing rate (MER), ρ_n, to denote the probability that an edge device possesses modality n. Specifically, as $\rho \to 0$, the variance in modality availability across edge devices increases. The local datasets for both the cloud server and edge devices are uniformly sampled from the global dataset. Each local dataset contains 2000 samples, of which 80% are used for training on both the devices and the server [20].

3) Training Settings: The shared parameter settings during the training process are summarized in Table 1. We adopt Adam [41] as the optimizer with a learning rate of 1.5×10^{-3} for all trainable components, and set the dropout rate to 0.3 [20]. An LSTM is used to extract features from signal data, while ResNet50 is applied to image data [22]. The models are trained collaboratively for 150 communication rounds with a batch size of 16 for both training and testing [21]. All parameters are initialized using Xavier initialization [42].

To emulate the edge-cloud environment, we leverage Python's multiprocessing package[1], assigning each edge device, including the server, to an individual process. This allows edge devices to conduct local training and transmit updated model parameters independently across communication rounds, while the server coordinates and aggregates these updates at each round.

4) Baselines: We compare our framework with five state-of-the-art baselines.

- **FedAvg** [13] is the first and one of the most widely adopted FL methods. In each training round, edge devices send their locally updated model parameters to a central server, which aggregates them by averaging to form a global model that is then redistributed for the next round of local training.
- **FedCor** [15] adopts a correlation-based device selection strategy for heterogeneous FL. It formulates the objective of accelerating convergence as an optimization problem that maximizes the expected posterior loss reduction among edge devices using a Gaussian process.
- **FedProx** [14] incorporates a proximal term into the local training loss among edge devices to constrain the divergence between local and global models, thereby aim to mitigate performance degradation caused by system and data heterogeneity.
- **AutoFed** [24] employs an autoencoder-based data imputation method to handle missing modalities. The autoencoder is pre-trained and remains fixed during the training process of AutoFed.
- **FedInMM** [27] utilize a dynamic weight map to rescale modality features and formulate different contributions of these features among edge devices.

5) Performance Metrics:

[1][Online]. Available: https://docs.python.org/3.8/library/multiprocessing.html

MER	Method	UR-FALL					
		Avg. Acc	B. Acc	W. Acc	Avg. F1	B. F1	W. F1
$\rho = 0.5$	FedAvg	0.608	0.694	0.555	0.614	0.669	0.527
	FedCor	0.655	0.716	0.604	0.634	0.671	0.587
	FedProx	0.623	0.689	0.584	0.578	0.673	0.528
	AutoFed	0.652	0.712	0.608	0.658	0.693	0.625
	FedInMM	0.715	0.729	0.674	0.702	0.721	0.677
	Ours	**0.741**	**0.753**	**0.732**	**0.733**	**0.747**	**0.710**
$\rho = 0.7$	FedAvg	0.627	0.692	0.575	0.629	0.674	0.583
	FedCor	0.664	0.685	0.613	0.657	0.699	0.594
	FedProx	0.640	0.677	0.598	0.631	0.685	0.596
	AutoFed	0.688	0.742	0.654	0.694	0.737	0.651
	FedInMM	0.729	0.737	0.701	0.724	0.742	0.705
	Ours	**0.767**	**0.776**	**0.754**	**0.758**	**0.771**	**0.739**
$\rho = 0.8$	FedAvg	0.706	0.741	0.678	0.710	0.737	0.687
	FedCor	0.714	0.749	0.692	0.697	0.731	0.666
	FedProx	0.726	0.741	0.699	0.730	0.748	0.712
	AutoFed	0.742	0.759	0.728	0.751	0.759	0.734
	FedInMM	0.758	0.767	0.741	0.756	0.768	0.748
	Ours	**0.787**	**0.794**	**0.782**	**0.778**	**0.785**	**0.771**

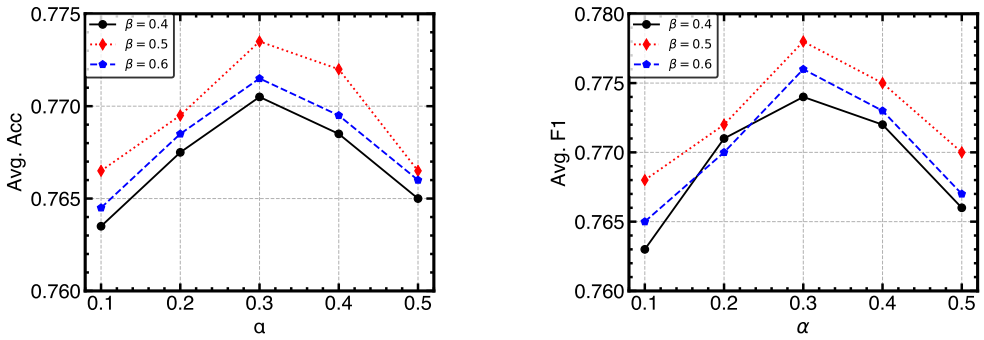
Table 2: Results on UR-FALL for performance evaluation.

- **Average Accuracy** (Avg. Acc) measures the mean test accuracy across all models on the edge devices.
- **Best Accuracy** (B. Acc) reports the hightest test accuracy achieved by any model across the edge devices.
- **Worst Accuracy** (W. Acc) records the lowest test accuracy among all models on the edge devices.
- **Average F1** (Avg. F1) denotes the average test F1 score across all edge models.
- **Best Accuracy** (B. F1) represents the highest F1 achieved by any edge models.
- **Worst Accuracy** (W. F1) records the minimum F1 across all edge models.

6 Results and Disscussion

6.1 Overall Performance Analysis and Discussion (RQ1)

Under modality-heterogeneous settings, we simulate different modality missing rates by setting the modality existing rate ρ to 0.5, 0.7, and 0.8. We evaluate the proposed method on the UR-FALL dataset and compare it with FedAvg, FedCor, FedProx, AutoFed, and FedInMM. The results are reported in Table 2. As shown in the table, FedHyperM consistently achieves the best performance across all modality availability settings and evaluation metrics, demonstrating its robustness under modality-missing conditions. Specifically, when $\rho = 0.5$, our method achieves 0.741 Avg. Acc, 0.753 B. Acc, 0.732 W. Acc, 0.733 Avg. F1, 0.747 B. F1, and 0.710 W. F1, outperforming the strongest baseline FedInMM by 3.64%, 3.29%, 8.61%, 4.42%, 3.61%, and 4.87%, respectively. When the modality availability increases to $\rho = 0.7$, our method further



(a) Average accuracy on UR-FALL.

(b) Average F1 on UR-FALL.

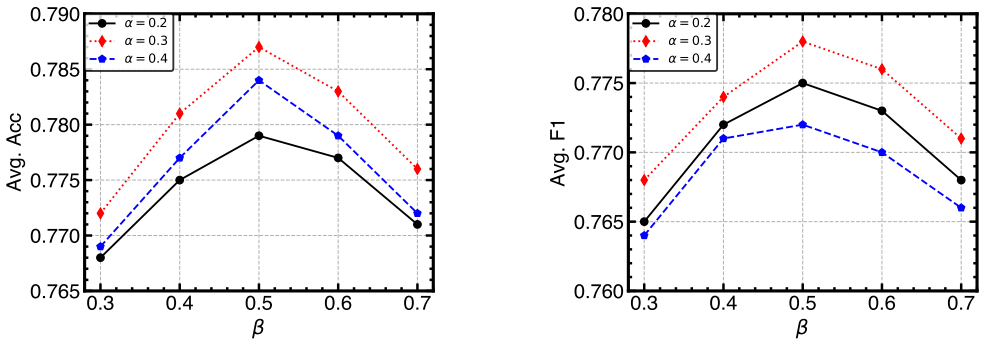
Fig. 2: The impact of α on FedHyperM for various β.

improves the performance to 0.767 Avg. Acc and 0.739 W. F1, with relative gains of 5.21% and 4.82% over FedInMM. Under $\rho = 0.8$, our method still maintains clear advantages, achieving 0.787 Avg. Acc, 0.794 B. Acc, and 0.771 W. F1.

Across all modality heterogeneity levels and all five baselines, FedHyperM achieves average relative improvements of 12.37% in Avg. Acc, 7.36% in B. Acc, 17.58% in W. Acc, 12.12% in Avg. F1, 7.71% in B. F1, and 16.43% in W. F1. Compared with all baselines across different modality heterogeneity levels, our method brings consistent improvements of 3.64% to 22.33% in Avg. Acc, 3.29% to 14.62% in B. Acc, 5.53% to 31.89% in W. Acc, 2.91% to 26.82% in Avg. F1, 2.21% to 14.39% in B. F1, and 3.07% to 34.72% in W. F1. These results indicate that the proposed method can effectively exploit multimodal information under heterogeneous modality availability and achieve stable performance improvements over existing multimodal federated learning baselines.

6.2 Sensitivity Analysis (RQ2)

In this section, we investigate the sensitivity of FedHyperM to two key hyperparameters, i.e., α and β, on the UR-FALL dataset, as shown in Fig 2 and Fig 3. Specifically, Fig 2 reports the impact of α under different values of β, while Fig 3 further analyzes the influence of β when α is fixed to different settings. We evaluate the performance using average accuracy and average F1-score. As illustrated in Fig 2, when β is fixed at 0.4, 0.5, and 0.6, both average accuracy and average F1-score first increase as α grows from 0.1 to 0.3, and then gradually decrease when α further increases to 0.5. This indicates that a small α may underemphasize the l1 norm regularization of the reconstruction vector limiting the effectiveness of FedHyperM in capturing useful heterogeneous information. In contrast, an overly large α may introduce excessive bias toward this factor, which weakens the balance among different components and slightly degrades the final performance. The best results are generally achieved when $\alpha = 0.3$, where FedHyperM obtains the highest average accuracy and average F1-score under different β settings. In addition, $\beta = 0.5$ consistently yields better performance than



(a) Average accuracy on UR-FALL.

(b) Average F1 on UR-FALL.

Fig. 3: The impact of β on FedHyperM for various α.

Method	Avg. Acc	B. Acc	W. Acc	Avg. F1	B. F1	W. F1
Ours	0.787	0.794	0.782	0.778	0.785	0.771
-w/o HyperG	0.738	0.754	0.717	0.744	0.761	0.735
-w/o MSA	0.721	0.739	0.706	0.718	0.738	0.695

Table 3: Ablation Study Results of FedHyperM on the UR-FALL Dataset.

$\beta = 0.4$ and $\beta = 0.6$, suggesting that a moderate β value is more suitable for the UR-FALL dataset.

Fig 3 further presents the effect of β when α is set to 0.2, 0.3, and 0.4. A similar trend can be observed: the performance improves as β increases from 0.3 to 0.5, but declines when β continues to increase to 0.7. When β is too small, FedHyperM may insufficiently consider the impact of l2 norm regularization of the reconstruction vector, leading to suboptimal model aggregation and representation learning. However, when β becomes too large, the model may overemphasize this factor and become less robust to modality and data heterogeneity. Among all tested settings, $\beta = 0.5$ achieves the best overall performance. Moreover, $\alpha = 0.3$ consistently outperforms $\alpha = 0.2$ and $\alpha = 0.4$ across most β values.

Overall, the results in Fig 2 and Fig 3 demonstrate that FedHyperM is relatively stable within a reasonable range of hyperparameter settings, while the best performance is obtained around $\alpha = 0.3$ and $\beta = 0.5$. Therefore, we adopt $\alpha = 0.3$ and $\beta = 0.5$ as the default configuration in our experiments.

6.3 Ablation Study (RQ3)

To evaluate the contribution of each core component in FedHyperM, we conduct an ablation study on the UR-FALL dataset, and the results are reported in Table 3. Specifically, we design two variants of FedHyperM. The first variant, FedHyperM w/o HyperG, removes the hypergraph learning module and directly concatenate the

multimodal features into an MLP for prediction. The second variant, FedHyperM w/o MSA, replaces the proposed modality-sensitive aggregation strategy with simple average aggregation.

As shown in Table 3, the complete FedHyperM consistently achieves the best performance across all evaluation metrics. Compared with FedHyperM w/o HyperG, FedHyperM improves Avg. Acc, B. Acc, W. Acc, Avg. F1, B. F1, and W. F1 from 0.738, 0.754, 0.717, 0.744, 0.761, and 0.735 to 0.787, 0.794, 0.782, 0.778, 0.785, and 0.771, respectively. This corresponds to relative gains of 6.64%, 5.31%, 9.07%, 4.57%, 3.15%, and 4.90%. These results indicate that the hypergraph learning module can effectively capture high-order correlations among multimodal features. In addition, removing the proposed MSA mechanism leads to more significant performance degradation. Compared with FedHyperM w/o MSA, FedHyperM achieves improvements of 9.15%, 7.44%, 10.76%, 8.36%, 6.37%, and 10.94% on Avg. Acc, B. Acc, W. Acc, Avg. F1, B. F1, and W. F1, respectively. This demonstrates that simple average aggregation is insufficient under modality-heterogeneous settings, since it ignores the modality-specific contribution of different edge models. By contrast, the proposed MSA mechanism assigns more appropriate aggregation weights according to modality-related information, thereby improving model robustness and generalization.

Overall, the ablation results verify the effectiveness of both HyperG and MSA. The performance drop caused by removing either component confirms that high-order multimodal relation modeling and modality-aware aggregation are both essential for FedHyperM under heterogeneous multimodal federated learning scenarios.

7 Conclusion

In this paper, we proposed FedHyperM, a federated multimodal learning framework for edge devices under modality heterogeneity. FedHyperM constructs adaptive hypergraphs on local devices to capture high-order correlations among available modalities and introduces a modality-sensitive aggregation mechanism to reduce aggregation noise caused by different modality combinations. Experiments on the UR-FALL dataset show that FedHyperM consistently outperforms representative baselines, including FedAvg, FedCor, FedProx, AutoFed, and FedInMM, under different modality existing rates. Sensitivity and ablation studies further verify the stability of FedHyperM and the effectiveness of its key components. These results demonstrate that FedHyperM can effectively improve federated multimodal learning performance while preserving local data privacy in heterogeneous edge environments.

8 List of Abbreviations

Declarations

Availability of data and material. The datasets used and analysed during the current study are publicly available, and the relevant sources are cited in the manuscript.

Competing interests. The authors declare that they have no competing interests.

Abbrevations	Full name
FedHyperM	Federated Hypergraph Learning for Multimodal Prediction
MSA	Modality-Sensitive Aggregation
MinED	Minimum Edit Distance
MER	Modality Existence Rate
IoT	Internet of Things
DL	Deep Learning
EC	Edge Computing
FL	Federated Learning
non-IID	Non-Independent and Identically Distributed
Avg. Acc	Average Accuracy
B. Acc	Best Accuracy
W. Acc	Worst Accuracy
Avg. F1	Average F1 Score
B. F1	Best F1 Score
W. F1	Worst F1 Score

Table 4: List of Abbreviations.

Funding. No funding was received for this study.

Authors' contributions. Qibin Zhou developed the proposed method, conducted the experiments, analysed the results, and prepared the first draft of the manuscript. Jianqi Shi supervised the research and revised the manuscript. Boon Xian Chai and Rifai Chai contributed to the discussion of the method and results. Xin Li and Di Guo reviewed the manuscript and provided suggestions. All authors read and approved the final manuscript.

References

[1] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., Zomaya, A.Y.: Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal **7**(8), 7457–7469 (2020)

[2] Alatawi, M.N.: Edge computing and federated learning for privacy-preserving iot analytics. EURASIP Journal on Wireless Communications and Networking **2026**(1), 1–6 (2026)

[3] Li, Y.-H., Chang, C.-C., Su, G.-D., Yang, K.-L., Aslam, M.S., Liu, Y.: Coverless image steganography using morphed face recognition based on convolutional neural network. EURASIP Journal on Wireless Communications and Networking **2022**(1), 1–28 (2022)

[4] Kwolek, B., Kepski, M.: Improving fall detection by the use of depth sensor and accelerometer. Neurocomputing **168**(1), 637–645 (2015)

[5] Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Computer Methods and Programs in Biomedicine **117**(3), 489–501 (2014)

[6] Gupta, P., Chouhan, A.V., Wajeed, M.A., Tiwari, S., Bist, A.S., Puri, S.C.: Prediction of health monitoring with deep learning using edge computing. Measurement: Sensors **25**(1), 100604 (2023)

[7] Yar, H., Imran, A.S., Khan, Z.A., Sajjad, M., Kastrati, Z.: Towards smart home automation using iot-enabled edge-computing paradigm. Sensors **21**(14), 4932 (2021)

[8] Wang, J., Jin, J., Zhang, L., Dai, H.-N., Di-Pietro, A., Zhang, T.: Towards spatial-temporal meta-hypergraph learning for multimodal few-shot fault diagnosis. Journal of Industrial Information Integration **1**(1), 100924 (2025)

[9] Wang, J., Jin, J., Zhang, T., Chai, B.X., Di Pietro, A., Georgakopoulos, D.: Leveraging auxiliary task relevance for enhanced bearing fault diagnosis through curriculum meta-learning. IEEE Sensors Journal (2025)

[10] Aldrees, A., Dutta, A.K., Emara, A., Shahab, S., Shaikh, Z.A., Daradkeh, Y.I., Anjum, M.: Enhancing dynamic resource management in decentralized federated learning for collaborative edge internet of things. EURASIP Journal on Wireless Communications and Networking **2025**(1), 1–32 (2025)

[11] Li, Z., Li, Q., Zhou, Y., Zhong, W., Zhang, G., Wu, C.: Edge-cloud collaborative learning with federated and centralized features. In: ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1949–1953 (2023)

[12] Zhang, Y., Luo, Y., Yang, T., Wu, X., Hu, B.: Eecs-fl: energy-efficient client selection for federated learning in aiot. EURASIP Journal on Wireless Communications and Networking **2025**(1), 1–12 (2025)

[13] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017)

[14] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Machine Learning and Systems **2**(1), 429–450 (2020)

[15] Tang, M., Ning, X., Wang, Y., Sun, J., Wang, Y., Li, H., Chen, Y.: Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10102–10111 (2022)

[16] Alam, F., Ofli, F., Imran, M.: Crisismmd: Multimodal twitter datasets from natural disasters. In: International AAAI Conference on Web and Social Media, vol. 12, pp. 1–10 (2018)

[17] Lau, J.J., Gayen, S., Ben Abacha, A., Demner-Fushman, D.: A dataset of clinically generated visual questions and answers about radiology images. Scientific Data **5**(1), 180251–180261 (2018)

[18] Wang, H., Liu, X., Zhong, X., Chen, L., Liu, F., Zhang, W.: Multimodal online federated learning with modality missing in internet of things. IEEE Transactions on Mobile Computing (2025)

[19] Liu, Y., Wang, C., Yuan, X.: Fedmobile: Enabling knowledge contribution-aware multi-modal federated learning with incomplete modalities. In: ACM on Web Conference, pp. 2775–2786 (2025)

[20] Zhang, T., Liu, Y., Shen, Z., Xu, R., Chen, X., Huang, X., Zheng, X.: An adaptive federated relevance framework for spatial–temporal graph learning. IEEE Transactions on Artificial Intelligence **5**(5), 2227–2240 (2023)

[21] Liu, Y., Zhang, T., Shen, Z., Wu, L., Chen, S., Jin, J.: Towards heterogeneity-aware and energy-efficient topology optimization for decentralized federated learning in edge environment. IEEE Transactions on Mobile Computing **25**(6), 7954–7969 (2026)

[22] Zhao, Y., Barnaghi, P., Haddadi, H.: Multimodal federated learning on iot data. In: IEEE/ACM International Conference on Internet-of-Things Design and Implementation, pp. 43–54 (2022)

[23] Feng, T., Bose, D., Zhang, T., Hebbar, R., Ramakrishna, A., Gupta, R., Zhang, M., Avestimehr, S., Narayanan, S.: Fedmultimodal: A benchmark for multimodal federated learning. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4035–4045 (2023)

[24] Zheng, T., Li, A., Chen, Z., Wang, H., Luo, J.: Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving. In: Annual International Conference on Mobile Computing and Networking, pp. 1–15 (2023)

[25] Chen, J., Zhang, A.: Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 87–96 (2022)

[26] Chen, J., Zhang, A.: Fedmbridge: Bridgeable multimodal federated learning. In: International Conference on Machine Learning, pp. 1–12 (2024)

[27] Yu, S., Wang, J., Hussein, W., Hung, P.C.: Robust multimodal federated learning for incomplete modalities. Computer Communications **214**(1), 234–243 (2024)

[28] Wang, J., Zhang, T., Zhang, L., Bai, Y., Li, X., Jin, J.: Hyperman: Hypergraph-enhanced meta-learning adaptive network for next poi recommendation. In: IEEE International Conference on Multimedia and Expo, pp. 1–6 (2025)

[29] Li, J., Zhang, T., Wang, J., Zhang, Y., Li, Y., Zhao, Y., Shen, Z., Wu, L., Liu, J.: Metasth-sleep: Towards effective few-shot sleep stage classification for health management with spatial-temporal hypergraph enhanced meta-learning. arXiv preprint arXiv:2505.17142 (2025)

[30] Wang, M., Liu, X., Wu, X.: Visual classification by l1 hypergraph modeling. IEEE Transactions on Knowledge and Data Engineering **27**(9), 2564–2574 (2015)

[31] Liu, Y., Zhao, Z., Zhang, T., Wang, K., Chen, X., Huang, X., Yin, J., Shen, Z.: Exploiting spatial-temporal data for sleep stage classification via hypergraph learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5430–5434 (2024)

[32] Zhang, T., Liu, Y., Shen, Z., Ma, X., Qi, P., Ding, Z., Jin, J.: Learning from heterogeneity: A dynamic learning framework for hypergraphs. IEEE Transactions on Artificial Intelligence **6**(6), 1513–1528 (2025)

[33] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Aassociation for Computational Linguistics, pp. 4171–4186 (2019)

[34] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021)

[35] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)

[36] Chen, S., Wu, Y., Wang, C., Liu, S., Tompkins, D., Chen, Z., Wei, F.: Beats: Audio pre-training with acoustic tokenizers. arXiv preprint arXiv:2212.09058 (2022)

[37] Zhang, T., Xu, R., Zhang, J., Liu, Y., Chen, X., Yin, J., Zheng, X.: DSHGT: Dual-supervisors heterogeneous graph transformer—a pioneer study of using heterogeneous graph learning for detecting software vulnerabilities. ACM Transactions on Software Engineering and Methodology **33**(8), 1–31 (2024)

[38] Zhang, T., Liu, Y., Chen, X., Huang, X., Zhu, F., Zheng, X.: GPS: A policy-driven sampling approach for graph representation learning. arXiv preprint arXiv:2112.14482 (2021)

[39] Liu, Y., Chu, S., Zhang, T., Zhou, H., Shen, Z., Wang, J., Qi, J., Xia, F.: ML-ECS: A collaborative multimodal learning framework for edge-cloud synergies. arXiv preprint arXiv:2602.14107 (2026)

[40] Fu, Y., Peng, H., Ou, L., Sabharwal, A., Khot, T.: Specializing smaller language models towards multi-step reasoning. In: International Conference on Machine Learning, pp. 10421–10430 (2023)

[41] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[42] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)

Appendix: Figure Title and Legend

Figure 1: The overview of FedHyperM. framework.

FedHyperM consists of three key processes for multimodal edge–server collaborative learning. First, multimodal raw data collected from edge devices are encoded into high-quality modality-specific representations using dedicated feature extractors, enabling each modality to preserve its discriminative semantic information. Second, on the edge side, FedHyperM adaptively constructs hypergraphs over the extracted multimodal features, where hyperedges model high-order correlations among samples and modalities. Hypergraph learning is then performed to enhance feature interaction and improve local model representation under heterogeneous multimodal conditions. Third, on the server side, a modality-sensitive aggregation (MSA) mechanism is applied to dynamically adjust the contribution of each edge model according to its modality characteristics, thereby alleviating performance degradation caused by multimodal heterogeneity and improving global model robustness.

Figure 2: The impact of α on FedHyperM for various β.

(a) The line chart illustrates the effect of α on average accuracy under different values of β on the UR-FALL dataset, showing that the optimal setting of α is 0.3; (b) The results present the influence of α on average F1 under different β values on the UR-FALL dataset, demonstrating that $\alpha = 0.3$ achieves the best performance.

Figure 3: The impact of β on FedHyperM for various α.

(a) The line chart illustrates the effect of β on average accuracy under different values of α on the UR-FALL dataset, showing that the optimal setting of β is 0.5; (b) The results present the influence of β on average F1 under different α values on the UR-FALL dataset, demonstrating that $\beta = 0.5$ achieves the best performance.