# FedMSplit: Correlation-Adaptive Federated Multi-Task Learning across Multimodal Split Networks

Jiayi Chen
University of Virginia
Charlottesville, Virginia, USA
jc4td@virginia.edu

Aidong Zhang
University of Virginia
Charlottesville, Virginia, USA
aidong@virginia.edu

## ABSTRACT

With the advancement of data collection techniques, end users are interested in how different types of data can collaborate to improve our life experiences. Multimodal Federated Learning (MFL) is an emerging area allowing many distributed clients, each of which can collect data from multiple types of sensors, to participate in the training of some multimodal data-related models without sharing their data. In this paper, we address a novel challenging issue in MFL, the modality incongruity, where clients may have heterogeneous setups of sensors and their local data consists of different combinations of modalities. With the modality incongruity, clients may solve different tasks on different parameter spaces, which escalates the difficulties in dealing with the statistical heterogeneity problem of federated learning; also, it would be hard to perform accurate model aggregation across different types of clients. To tackle these challenges, in this work, we propose the FedMSplit framework, which allows federated training over multimodal distributed data without assuming similar active sensors in all clients. The key idea of FedMSplit is to employ a dynamic and multi-view graph structure to adaptively capture the correlations amongst multimodal client models. More specifically, we split client models into smaller shareable blocks and allow each type of blocks to provide a specific view on client relationships. With the graph representation, the underlying correlations between clients can be captured as the edge features in the multi-view graph, and then be utilized to promote local model relations through the neighborhood message passing in the graph. Our experimental results demonstrate the effectiveness of our method under different sensor setups with statistical heterogeneity.

## CCS CONCEPTS

• **Computing methodologies → Learning paradigms**; **Multi-task learning**; *Supervised learning by classification*; Distributed artificial intelligence.

## KEYWORDS

Federated Learning, multitask learning, modality incongruity

**ACM Reference Format:**
Jiayi Chen and Aidong Zhang. 2022. FedMSplit: Correlation-Adaptive Federated Multi-Task Learning across Multimodal Split Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539384

## 1 INTRODUCTION

Federated learning (FL) is currently the dominant framework for distributed training of machine learning models under communication and privacy constraints [6, 9, 13, 14, 17, 18, 23, 28, 31]. An FL paradigm typically involves multiple *clients* collecting data and jointly training ML models or Deep Neural Networks without releasing their local data. Recently, with the advancement of sensory techniques and the dramatically increasing multimodal data, Multimodal Federated Learning (MFL) has attracted much research attention [19, 36, 39]. MFL focuses on how massive distributed clients, each of which can collect multimodal data from multiple types of sensors (e.g., image, video, audio, texts, time-series data, and etc.), can collaborate to train multimodal tasks-related models (e.g., multimodal fusion, cross-modal translation, multimodal knowledge bases, etc.) [25, 29, 37] without sharing their data.

While a majority of previous FL and MFL frameworks focus on the *statistical heterogeneity* problem in the federated training system (i.e., non-IID data over clients) [6, 9, 17, 28], a potential limitation of these methods is that they still assume the *modality congruity* among client models, i.e., all clients have the same setup of input modalities and client models share the same parameter space. However, in real-world MFL scenarios where there are multiple types of sensors, the assumption may not be true. It should be aware that different clients may not have the same type or the same set of working sensors due to the differences of their environments, network connections, and sensor affordability. For example, healthcare centers in remote areas usually lack advanced medical equipment so that their models relies on data collected by other available sensors; in dynamic systems or online learning applications, sensor availability may be not stable over time, thus several modalities can be missing frequently. Therefore, real-world MFL systems usually observe the *modality incongruity* problem–that is, clients may have heterogeneous setups of sensors and their local data consists of different combinations of modalities.

In this paper, we study multimodal FL and focus on not only statistical heterogeneity but also the novel *modality incongruity* problem in MFL. The modality incongruity across clients, which has been neglected by existing FL works, can be a significant issue that will potentially impact the performance and robustness of FL systems. Specifically, it escalates the heterogeneity among clients. Since clients collect data from different sets of sensors, they actually

*KDD '22, August 14–18, 2022, Washington, DC, USA.*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539384

solve different tasks using different model architectures. Then, it could be difficult for clients to collaborate to find a common model for different tasks, and it is hard to perform model aggregation across the clients having different parameter sets.

To approach MFL with both modality incongruity and statistical heterogeneity challenges, we first choose to build our framework based on the federated multi-task learning (FMTL) paradigm, which deals with statistical heterogeneity by learning separate but related models for each clients through a global regularization term that captures client relationships [28]. Inspired by this, regarding the modality incongruity problem, our goal is to build a novel multimodal FMTL framework, which can adaptively explore the relationships between different *types* of clients and finally learn personalized but globally correlated multimodal client models. However, there are two challenges in the multimodal FMTL due to modality incongruity. (1) *First*, it is challenging to measure the relationships between the clients which solve different tasks in different parameter spaces. As a result, how to perform accurate model aggregation across different types of tasks (clients) remains challenging. Although one can simply unify client models by imputing missing modalities and padding the model parameters related to the missing sensors for each client, the extra neurons participating in the federated training will introduce a lot of *noise* during the model aggregation process, as well as *not efficient* as extra modalities and model parameters participate in each communication round. Therefore, we aim to directly learn the individual client models in different parameter spaces. (2) *Second*, due to communication limitation in the real-world FL environments [17], only a subset of clients can participate in the correlated training at each round. Since different clients focus on solving different tasks using different model architectures, it would be hard to guarantee precise model correlations using partial clients. Existing FMTL approaches either perform random client selection or just select nearly all clients (tasks) to participate in each round, which is not efficient with modality incongruity settings. To achieve faster convergence, it is of importance to select different types of clients in a balanced manner at each round.

To tackle the aforementioned challenges, in this work, we formulate a new fundamental structure that facilitates the adaptive correlations amongst different types of multimodal client models. We propose the **FedMSplit** framework, which allows federated training over multimodal client models without assuming the consistency of sensor setups of clients. The key idea of FedMSplit is to employ a *dynamic and multi-view graph structure*, where each vertex corresponds to a client solving the subproblem (local objective) on its local dataset, to automatically capture and utilize the relationships among clients to achieve correlated local model updates. In particular, we propose to split client models into smaller blocks–some blocks are shared by all clients while some are shared amongst a subset of clients, and allow each type of blocks to provide a specific view on client relationships. Then, given the graph representation of multimodal clients, the underlying statistical correlations between clients can be captured as the edge features in the multi-view graph, and then be used to promote local model relations through the neighborhood message passing in graph.

Our contributions are summarized as follows: (1) First, we propose a novel multimodal federated learning framework, which allows federated training over clients that have heterogeneous setups of sensors. To the best of knowledge, this is the first work studying the modality incongruity problem using FMTL. (2) Second, we propose FedMSplit, which employs a dynamic graph structure to adaptively capture the relationships among different types of clients and then achieve correlated model training. We adopt the multi-armed bandit algorithm over the graph, to perform efficient client selection amongst different client architectures. (3) Finally, we evaluate FedMSplit on two multimodal federated datasets with different setups of modality incongruity. The empirical results show the effectiveness of our method.

## 2 RELATED WORKS AND BACKGROUND

**Federated Learning**. A federated learning framework typically involves multiple *clients* collecting data and a central *server* coordinating the learning objective. Given that there are $N$ clients, where client $k$ has a local dataset $\mathcal{D}_k$ containing $n_k = |\mathcal{D}_k|$ data samples, and suppose $f_k(\mathbf{w}, \mathbf{x}_i, y_i)$ is the composite loss function for sample $(\mathbf{x}_i, y_i)$ and parameter vector $\mathbf{w}$ at client $k$, the general FL framework [17, 22] aims to find a single global model $\mathbf{w} \in \mathbb{R}^d$ across the local data:

$$\min_{\mathbf{w}} \left\{ F(\mathbf{w}) = \sum_{k=1}^{N} p_k F_k(\mathbf{w}) = \frac{1}{\sum_k n_k} \sum_{k=1}^{N} \sum_{i=1}^{n_k} f_k(\mathbf{w}, \mathbf{x}_i, y_i) \right\}, \quad (1)$$

where $p_k = \frac{n_k}{\sum_k n_k}$ is the fraction of data at the $k$-th client. The term $F_k(\mathbf{w}) = \frac{1}{n_k} \sum_{i=1}^{n_k} f_k(\mathbf{w}, \mathbf{x}_i, y_i)$ is the local objective function of client $k$, given by the average empirical risks over local samples.

Due to computation and communication limitations, the training of Eq.(1) only runs on a subset of clients $\mathcal{S}(t)$ performing local gradient descent at each round $t$, i.e., $\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \sum_{k \in \mathcal{S}(t)} p'_k \nabla F_k(\mathbf{w}(t))$ until convergence. Previous works have investigated the convergence properties of FL in congruent IID and non-IID scenarios. One of the earliest work of FL is FedAvg [23], which builds the global model based on averaging the local Stochastic Gradient Descent (SGD) [12, 30] updates. It is noticeable that when the statistical heterogeneity between local datasets increases (non-IID), the differences between functions $F_1, F_2, ..., F_N$ increase. Various methods [13, 14, 17, 18] are introduced to improve the robustness of the global model under non-IID settings. For example, FedProx [17] adds a proximal term to the local objective. Personalized FL [6, 9, 31] has been proposed as an alternative to deal with non-IID data, where the global model plays the role of a meta-model to be used as initialization for few-shot adaptation at each client. For example, pFedMe [6] used Moreau envelopes, while PerFedAvg [9] took advances of meta learning approaches: Model-agnostic Meta-learning (MAML) [11]. A common theme of conventional and personalized FL is that they learn a single global model $\mathbf{w}$. However, the limitation is that if local distributions are far from the average distribution [21], a relevant global generalization model does not exist and every client will learn only on its own local data [9]. These methods ignore the correlation amongst clients and thus cannot take the knowledge of other clients having similar local distributions. Moreover, in the sense of multimodal FL where client models are different due to not only local distributions as well as parameter spaces, it is hard to train a single global model that performs well to all types of clients.

**Federated Multi-Task Learning**. Federated Multi-Task Learning (FMTL) is an alternative approach to deal with the non-IID data in FL setting. In comparison to the FL frameworks learning a single global model $\mathbf{w}$, FMTL [28] aims to fit *separate but related* models $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N \in \mathbb{R}^d$ for each clients:

$$\min_{\mathbf{w}_1,...,\mathbf{w}_N,\Omega} \left\{ \sum_{k=1}^{N} \sum_{i=1}^{n_k} f_k(\mathbf{w}_k, \mathbf{x}_i, y_i) + \mathcal{R}(\mathbf{W}, \Omega) \right\} \tag{2}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_N] \in \mathbb{R}^{d \times N}$ is a matrix collecting weight vectors for all clients and $\mathbf{\Omega} \in \mathbb{R}^{N \times N}$ represents the relationships amongst clients. The regularization term $\mathcal{R}(\mathbf{W}, \Omega) = \lambda_1 \mathrm{tr}(\mathbf{W}\Omega\mathbf{W}^\top) + \lambda_2 ||\mathbf{W}||_F^2$ is used to enforce some suitable model similarities among clients that have similar local distributions.

MOCHA [28] first shows that multi-task learning [2] is well-suited to handle the statistical challenges of the federated setting. VIRTUAL [5] treats the FL network as a star-shaped Bayesian network and performs learning using approximated variational inference. OFMTL [16] focused on online applications. Recent works [7, 26] have explored empirical graphs to explicitly leverage the relationships among the clients' model to enforce the clients having similar datasets to have similar predictors. FMTL approaches can directly capture relationships amongst non-IID and unbalanced data, which is well-suited for the statistical challenge of FL. However, existing methods either rely on the complete set of clients participating in each round or a full-sized precomputed relationship measurements to guarantee precise model correlation, which is hard to scale to massive client populations where only a small portion of local models participate at each round. Furthermore, previous works have not discussed the multimodal scenarios that contain heterogeneous model architectures over different types of unimodal/multimodal clients, which can lead to more complex client relationships as well as unbalanced client selection, making the convergence difficult.

**Multimodal Federated Learning**. Multimodal Federated Learning (MFL) is an emerging area in FL focusing on learning multimodal task-related models on the multi-sensory data distribued over clients. At the first glance, we can simply apply existing FL methods on multi-view tasks [17, 21, 23, 28]. However, most of the existing FL methods assume that all clients have each sensor correctly working at all times, thus they are not robust to the situations where most of the clients have unavailable or dropped sensors (e.g., unimodal clients). [19] applies FL on data from two modalities (i.e., images and texts) where representations of local data need to be uploaded to the server. This yet breaks the privacy guarantee of FL because the server could recover the raw data if it has those representations. A recent study [39] learns the correlated alignment information from multiple modalities in the unsupervised manner, which allows the global model to be trained and used on unimodal as well as multimodal data. [36] uses the co-attention mechanism in personalized FL to fuse the complementary information of different modalities. These methods learn a single global model without the consideration of client relationships, whereas we learn separate but correlated models for each clients and take into account the relationships between non-IID and modality-discrepant clients.

# 3 PROBLEM FORMULATION

In this section, we introduce the proposed multimodal FL setup and the problem formulation.

**MFL Setup with Modality Incongruity**. We focus on the MFL problem with *modality incongruity*, where clients have heterogeneous setups of sensors. Formally, given $M$ modalities in total, client-$k$'s local datasets is $\mathcal{D}_k = \{(\mathcal{X}_i, y_i)\}_{i=1}^{n_k}$, where

$$\mathcal{X}_i = \{\mathbf{x}_i^{(j)} | \forall j \in \mathcal{B}_k\} \tag{3}$$

is the input modalities of each sample $i$ and $\mathcal{B}_k \subseteq \mathcal{B} = \{1, 2, ..., M\}$ is the set of active sensors at client $k$. Note that since the number of non-empty subsets of $\mathcal{B}$ is $(2^{|\mathcal{B}|} - 1)$, there can be at most $(2^M - 1)$ types of clients in the network–each type of clients have a certain combination of sensors. In Figure 1(a), we show a trimodal federated dataset, where the cylinders in different colors (yellow, green, and red) illustrate the data collected from different types of sensors and there can be at most seven types of clients.

**Multimodal Split Networks**. In most of existing federated learning frameworks, at each client the model weights are in the same vector space $\mathbf{w}_k \in \mathbb{R}^d$ for $k = 1, ..., N$. In contrast, given the modality incongruity amongst clients, clients having different sensor setups do not share the same model architecture $\mathbf{w}_k \in \mathbb{R}^{d_k}$ where $d_1 \neq d_2 \neq ... \neq d_N$, as shown in the middle of Figure 1(a). This will lead to difficulties in server-client and cross-client communication as different client models cannot be copied or aggregated directly. One may consider a heuristic strategy unifying all client models into the largest one by inserting missing blocks on the input layers, or deleting some blocks for under-predominant modalities. Yet this will introduce bias to model aggregation as well as not efficient. One may also argue that the clients having different sensors should be totally separated from each other during modal aggregation; however, this is not true as the different modalities across clients may still have common knowledge to learn–for example, the sound data and visual appearance of the same object can exists in different but statistically closer clients.

Therefore, instead of totally unifying or separating clients, we aim to directly learn the original client models. Inspired by the idea of split learning [27, 32] we *split* each client models into *blocks* such that there are two types of model blocks among all client models: 1) blocks shared globally by all clients and can be aggregated amongst the entire network; 2) blocks shared locally by the clients having the same corresponding sensors and can be aggregated across partial clients. For example, as for multimodal integration tasks [3, 20, 25, 37], which learn predictive models that integrate the information of given modalities to make decisions, we can split any client model into one or more *modality-specific feature extractors* and a *classifier* that takes the cross-modal aligned features as input. Figure 1 illustrates the multimodal split networks, i.e., the diverse model architectures of different types of integration tasks. Formally, suppose $f_k(\mathbf{w}_k, \mathcal{X}_i, y_i; \mathcal{B}_k)$ is the loss function for the multimodal sample $(\mathcal{X}_i, y_i)$ and parameter vector $\mathbf{w}_k \in \mathbb{R}^{d_k}$ at client $k$ whose sensor set is $\mathcal{B}_k$. We can split by

$$\mathbf{w}_k = \{\mathbf{w}_{kj} | \forall j \in \mathcal{B}_k\} \cup \{\overline{\mathbf{w}}_k\}, \tag{4}$$

where $\mathbf{w}_{kj} \in \mathbb{R}^{d'_j}$ is the weight vector of the feature extractor for sensor-$j$ and $\overline{\mathbf{w}}_k \in \mathbb{R}^{d'}$ is the weight vector of the classifier. That

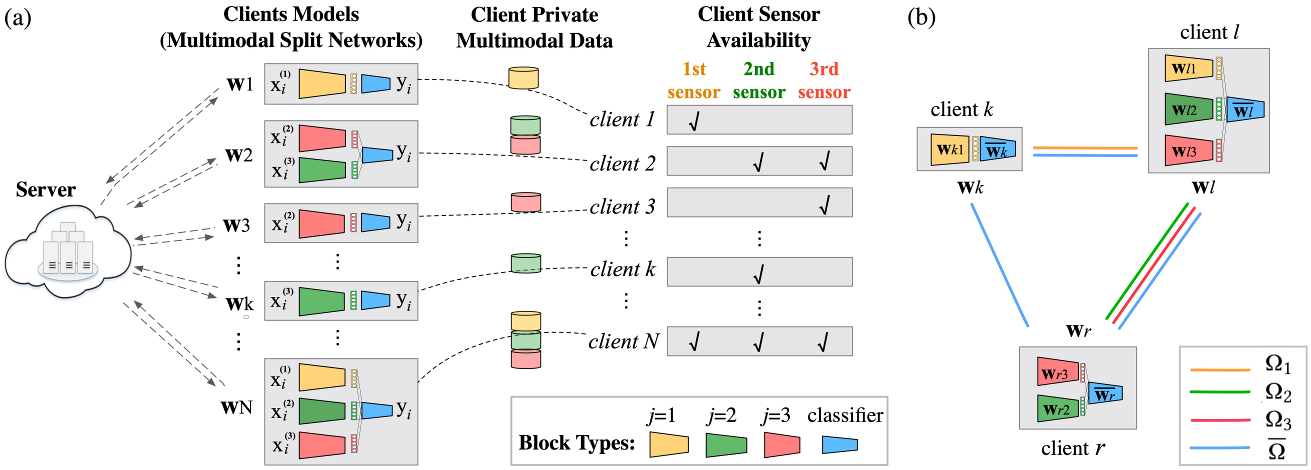

**Figure 1: (a) Multimodal federated learning setup with modality incongruity, where a total of $M = 3$ types of sensors are involved. The models shown here intend for multimodal fusion tasks, yet can be replaced by other models. Any type of client models can be split into at most four different blocks. (b) An illustration of the relationships between different types of clients.**

is, we break the weight vector of dimension as $d_k = d' + \sum_{j\in\mathcal{B}_k} d'_j$. Then, we can rewrite the loss function as

$$f_k(\mathbf{w}_k, \mathcal{X}_i, y_i; \mathcal{B}_k) = g_k\left(\overline{\mathbf{w}}_k, \oplus_{j\in\mathcal{B}_k}\mathbf{h}_i^{(j)}, y_i\right), \tag{5}$$

where the modality-specific hidden feature $\mathbf{h}_i^{(j)} = h_{kj}(\mathbf{w}_{kj}, \mathbf{x}_i^{(j)})$ is obtained through the sensor $j$'s specific feature encoder $h_{kj}$. Then, $\oplus$ denotes the sum of the hidden representations of individual modalities, which combines their complementary information, and $g_k$ is the loss function given the combined representation.

**Federated Multi-task Learning amongst Multimodal Split Networks**. Regarding the modality incongruity between clients together with the statistical and systematic challenges of FL problem, we formulate our problem based on the FMTL framework. The idea is that FMTL naturally explores the client relationships which can also help to find relationships between the split blocks of client models and is suitable for multimodal FL. Formally, we aim to learn a set of implicitly correlated models $\mathbf{w}_1, ..., \mathbf{w}_N$ having different parameter spaces by minimizing the objective:

$$\min_{\mathbf{w}_1,...,\mathbf{w}_N,\Omega_1,...,\Omega_M,\overline{\Omega}} \left\{ \sum_{k=1}^{N}\sum_{i=1}^{n_k} f_k(\mathbf{w}_k, \mathcal{X}_i, y_i; \mathcal{B}_k) + \widetilde{\mathcal{R}}(\mathbf{\Phi}, \mathbf{\Lambda}) \right\} \tag{6}$$

where $\mathbf{\Phi} = \{\mathbf{w}_k \in \mathbb{R}^{d_k}\}_{k=1}^{N}$ represents a collection of the multimodal models and $\mathbf{\Lambda} = [\Omega_1, ..., \Omega_M, \overline{\Omega}] \in \mathbb{R}^{N\times N\times(M+1)}$ is a tensor representing multi-view relationships amongst client models. Each view of the relationships $\Lambda_{\cdot,\cdot,m} \in \mathbb{R}^{N\times N}$ is a matrix corresponding to the relationships amongst certain type of blocks of all client models. If client $k$ and client $l$ do not have the common block-$j$, $\Lambda_{k,l,j} = 0$. Basically, the first term of Eq.(6) allows clients to learn on its own local data, while the second term encourages them to take advantages of related models from other clients'.

It is noticeable that, for any pair of client-$k$ and client-$l$: (1) *Observation 1*: their models $\mathbf{w}_k, \mathbf{w}_l \in \mathbf{\Phi}$ may be not comparable as they may belong to different parameter spaces, i.e., $d_k \neq d_l$; (2) *Observation 2*: the relationship between $w_k$ and $\mathbf{w}_l$ is measured by a non-scalar but multi-dimensional vector $\Lambda_{k,l,\cdot} \in \mathbb{R}^{M+1}$. As a result, directly optimizing $\widetilde{\mathcal{R}}(\mathbf{\Phi}, \mathbf{\Lambda})$ to enforce nuanced model relations can be intractable.

## 4 METHODOLOGY

In this section, we will introduce a federated training algorithm to solve the objective Eq.(6). We will focus on the following issues during the federated training with modality incongruity. (1) ***Adaptive model correlation with local dynamics***. The correlation tensor $\mathbf{\Lambda}$ is a measurement of statistical similarity between local datasets. With the privacy requirements in FL, we cannot calculate it in advance. Further, while we can arbitrarily provide $\mathbf{\Lambda}$ as priori [7, 26], in real-world applications, the relationships of clients may not be fixed all the time. For example, the statistics of local data can change through time such as given time-series data or continual learning tasks. Therefore, adaptively learning $\mathbf{\Lambda}$ with models is necessary. However, it is difficult to *simultaneously optimize* the multi-space parameters $\mathbf{\Phi} = \{\mathbf{w}_k \in \mathbb{R}^{d'+\sum_{j\in\mathcal{B}_k} d'_j}\}_{k=1}^{N}$ and the tensor $\mathbf{\Lambda}$. We adopt the *alternative optimization approach* following [28]. At each round $t$, while fixing the structure $\mathbf{\Lambda}(t)$, we optimize local model parameters $\mathbf{\Phi}(t)$ based on local datasets $\mathcal{D}_k, k = 1, ..., N$ and the penalized term given $\mathbf{\Lambda}(t)$; then, while fixing $\mathbf{\Phi}(t)$, we optimize the model correlation tensor $\mathbf{\Lambda}(t+1)$ based on current local models $\mathbf{\Phi}(t)$. The model correlation is dynamically updated along with the convergence of local models. (2) ***Correlated model update constrained by multi-view relationships***: The individual models of different clients may be not comparable if they belong to different parameter spaces. In this sense, how to measure model relationships, and how to leverage relational local model training according to multiple views of relations, remain unexplored. (3) ***Client selection among multiple types of clients***: Vanilla FTML relies on the complete adjacency for all the clients and update all the client models at each round. Although a complete relationship structure could benefit the correctness of correlated local updates, in practical scenarios where massive clients participate in the training, the

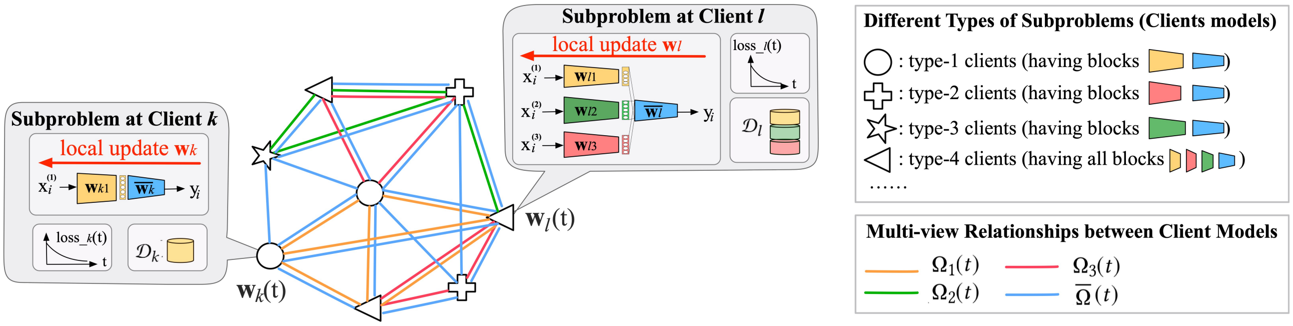

**Figure 2: A graph view for training separate but related clients having different model architectures. Each vertex solves separate but related subproblems. Node features (model parameters) changes over time based on local data as well as related clients.**

computation time and cost of storage at each round would huge. Regarding the systematic challenge and communication limitation [19], instead of computing all clients at each round, we sample a subset of clients to participate into training. However, given the multimodal discrepancy, the difficulty is that how to select clients at each round such that each model blocks are optimized in a balanced manner and we efficiently find the optimal for all clients.

We propose the **FedMSplit** framework to allow federated training over multimodal clients without assuming the congruity of sensor types over clients. Details of the framework are as follows.

## 4.1 Correlation-adaptive Model Update

According to the *alternative optimization* process, at each round $t$, while fixing the structure $\Lambda(t)$, we optimize local model parameters $\Phi(t)$ based on local datasets $\{\mathcal{D}_k\}_{k=1}^{N}$ and the penalized term given $\Lambda(t)$. In other words, the local model $\mathbf{w}_k(t)$ of each client $k$ updates depends on not only the local dataset $\mathcal{D}_k$ but also its related clients' datasets $\mathcal{D}_l$, which is not seen but can be reflected by $\mathbf{w}_l(t)$. Then, any pair of $\Lambda_{k,l,\cdot}(t)$ can update based on $\mathbf{w}_k(t)$ and $\mathbf{w}_l(t)$. Through this process, $\Lambda(t), \Phi(t)$ are dynamically updated until convergence.

Given the heterogeneity of parameter space over clients, "updating $\Phi(t)$ fixing $\Lambda(t)$" would be difficult as the calculation of $\nabla_{\Phi(t)}\widetilde{\mathcal{R}}(\Phi(t),\Lambda(t))$ relies on the $\Phi(t)$ having multiple parameter spaces, and the multi-view relationships $\Lambda(t)$. The idea is that we can allow the two sets of parameters $\Lambda(t), \Phi(t)$ to be embedded in a dynamic graph and then solve them as a node-edge alternative updating problem.

### 4.1.1 *Dynamic Multi-view Graph of Subproblems.*

We define a dynamic multi-view graph structure $\mathcal{G}(t) = (\mathcal{V}, \Phi(t), \mathcal{E}, \Lambda(t), q)$ which consists of the following components and properties:

- $\mathcal{V} = \{v_k\}_{k=1}^{N}$ is the vertex set, where each vertex $v_k$ is associated with a client $k$ containing a local multimodal dataset $\mathcal{D}_k = \{(\mathcal{X}_i, y_i)\}_{i=1}^{n_k}$. Each vertex represents a *subproblem*: the client $k$ aims to fit a model $\mathbf{w}_k \in \mathbb{R}^{d_k}$ to its local data $D_k$.
- $\Phi(t) = \{\mathbf{w}_k(t) \in \mathbb{R}^{d_k}\}_{k=1}^{N}$ is the content of vertices, representing the *model parameters* of each clients at round $t$. In this way, the model parameters of clients can be treated as *client embeddings* in the graph. Client model updating implies the changing of client embeddings through time, so we say the graph is *dynamic*. Recall that the client models has different parameter spaces $\mathbf{w}_k(t) = \{\mathbf{w}_{kj}(t)|\forall j \in \mathcal{B}_k\} \cup \{\overline{\mathbf{w}}_k(t)\}$ so that the vertices are multimodal and message cannot be directly transferred across vertices.
- $\mathcal{E} = \{e_{kl}\}_{k,l=1}^{N}$ is the edge set. Edges are undirected and fully connected, and each edge refers to the similarity between a pair of clients.
- $\Lambda(t) = [\Omega_1(t), ..., \Omega_M(t), \overline{\Omega}(t)] \in \mathbb{R}^{N\times N\times(M+1)}$ represents the edge features. An edge feature $\Lambda(t)_{k,l,\cdot}$ indicates the model weight similarities between two clients and consists of *multiple dimensions* (*multi-view*) of Euclidean distances; each dimension is corresponding to a type of blocks in client models. If client $k$ and client $l$ do not have the common block-$j$, $\Lambda_{k,l,j}(t) = 0$. The edge features of a fully connected graph with massive clients can be huge. Fortunately, in practice, the server does not need to calculate or store them until the end of each round, and only the edges between participated clients at each round will be calculated (see Section 4.2.2.).
- $q : \mathbb{R}^{M+1} \rightarrow \mathbb{R}$ is a function to measure the similarity between any of the two clients $(k, l)$ based on the multi-view correlation vector $\Lambda_{kl\cdot} \in \mathbb{R}^{M+1}$.

Figure 2 illustrates the defined dynamic graph structure, where we use different shapes to indicate individual clients having different types of parameter spaces. The embeddings of clients (model parameters) change over time. For simplicity, we only show eight clients, three modalities, and four types of clients here. Note that in real-world applications we would have more client types in a large-scale graph containing massive number of clients.

Giving the graph of local problems, then the intractable term $\widetilde{\mathcal{R}}(\Phi,\Lambda)$ can be rewritten as the linear combination of multiple views of regularization terms (each view is corresponding to a type of blocks):

$$\widetilde{\mathcal{R}}(\Phi,\Lambda) = \lambda\overline{\mathcal{R}}(\overline{\mathbf{W}},\Lambda,\overline{\Omega}) + \sum_{j=1}^{M}\lambda_j\mathcal{R}_j(\mathbf{W}_j,\Lambda,\Omega_j) \qquad (7)$$

where $\overline{\mathbf{W}} = [\overline{\mathbf{w}}_1, ..., \overline{\mathbf{w}}_N] \in \mathbb{R}^{d'\times N}$ and $\mathbf{W}_j = [\mathbf{w}_{k,j}; \forall k \text{ if } j \in \mathcal{B}_k] \in \mathbb{R}^{d'_j\times N_j}$ are matrices. Each matrix is a collection of a specific block in all clients, over one parameter space.

### 4.1.2 *Correlation-adaptive Model Optimization via Node-Edge Alternative Update.*

Updating correlation $\Lambda(t)$ fixing $\Phi(t)$ is treated as updating edge features based on current node features (client embeddings). Formally, for each pair of clients $(k, l)$ and

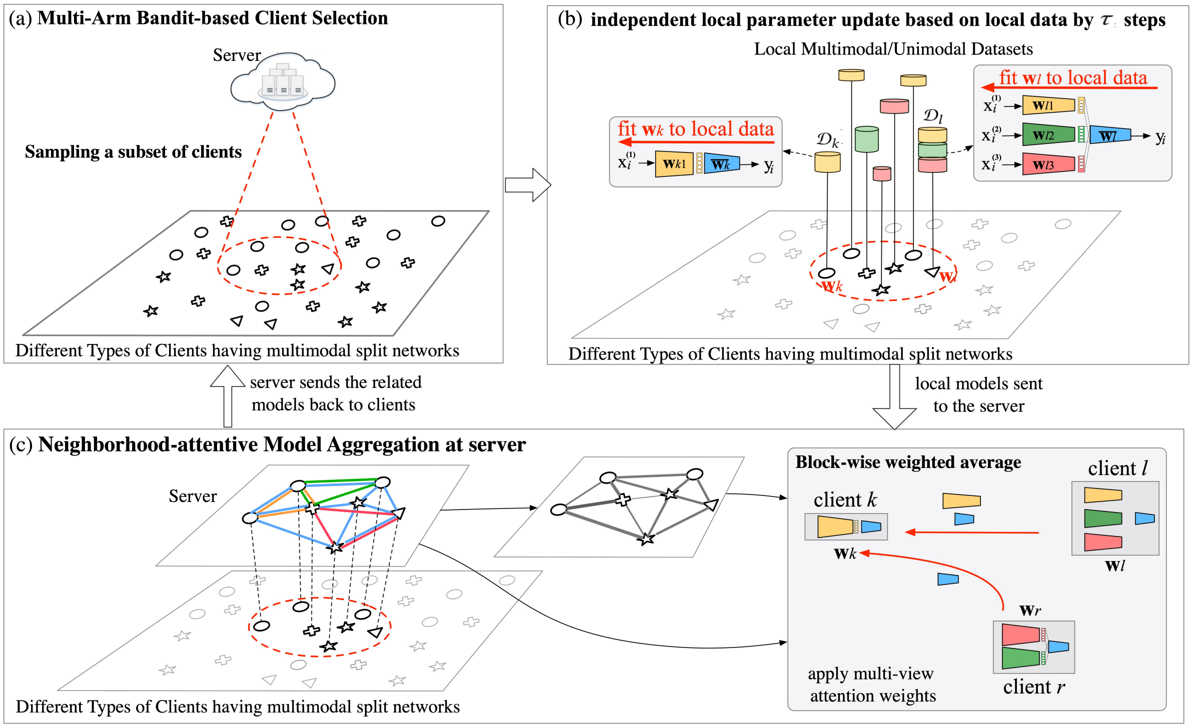
(a) Multi-Arm Bandit-based Client Selection
Server
Sampling a subset of clients
Different Types of Clients having multimodal split networks
(b) independent local parameter update based on local data by τ steps
Local Multimodal/Unimodal Datasets
fit wk to local data
fit wl to local data
$x_i^{(1)}$
$x_i^{(2)}$
$x_i^{(3)}$
wk1
wk
wl1
wl2
wl3
wl
$y_i$
$\mathcal{D}_k$
$\mathcal{D}_l$
wk
wl
Different Types of Clients having multimodal split networks
server sends the related models back to clients
local models sent to the server
(c) Neighborhood-attentive Model Aggregation at server
Server
Block-wise weighted average
client k
client l
wk
wl
wr
client r
apply multi-view attention weights
Different Types of Clients having multimodal split networks

**Figure 3: Overview of FedMSplit training procedure at each server-client communication round.**

each split block $j$, their relationship can be measured as $\Omega_{j,kl} = Att(\mathbf{w}_{kj}, \mathbf{w}_{lj})$ using any metric or attention function $Att(\cdot,\cdot)$, such as additive attention, dot product, multiplicative attention [33]. In the experiments, we use dot product for all simulations.

Then, updating models $\Phi(t)$ fixing the structure $\Lambda(t)$ is viewed as updating node features (client models) based on local datasets as well as current edge features (client-client relationships). The idea is to take into consideration the current overall client-to-client relationships $\Lambda(t)$ to the local training of each client $\mathbf{w}_k$. More specifically, we take two steps to incorporate such heterogeneous and multi-space complex relationships. *First*, for each client, we approximate its complex neighborhood information, by aggregating other client models through a multi-view, attentive, and graph-based message passing process.

$$\begin{cases} \overline{\mathbf{w}}_k^{agg} \leftarrow \sum_{l=1}^N \frac{q(\Lambda_{kl\cdot})\overline{\Omega}_{kl}}{\sum_{p=1}^N q(\Lambda_{kp\cdot})\overline{\Omega}_{kp}} \overline{\mathbf{w}}_l, \\ \mathbf{w}_{kj}^{agg} \leftarrow \sum_{l=1}^N \frac{q(\Lambda_{kl\cdot})\Omega_{j,kl}}{\sum_{p=1}^N q(\Lambda_{kp\cdot})\Omega_{j,kp}} \mathbf{w}_{lj} \text{ for } \forall j \in \mathcal{B}_k \cap \mathcal{B}_l. \end{cases} \tag{8}$$

After that, each client $k$ independently performs local SGD on $\mathcal{D}_k$ by $\tau$ steps, meanwhile, it considers extra relational information:

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \frac{1}{n_k} \sum_{i=1}^{n_k} (\nabla_{\mathbf{w}_k} f_k(\mathbf{w}_k, \mathcal{X}_i, y_i; \mathcal{B}_k) + \nabla_{\mathbf{w}_k} \lambda R_k(\mathbf{w}_k)), \tag{9}$$

where the multi-view relational information is incorporated by minimizing the Mean Squared Error (MSE) loss between the model and the approximate neighborhood information $R_k(\mathbf{w}_k) = ||\overline{\mathbf{w}}_k - \overline{\mathbf{w}}_k^{agg}||_2^2 + \sum_j ||\mathbf{w}_{kj} - \mathbf{w}_{kj}^{agg}||_2^2$. And $\lambda$ is a hyperparameter to balance the local personalization and global correlation.

In this way, the model parameters $\mathbf{w}_k$ is updated based on $\mathcal{D}_k$ (Eq.(9)) as well as other models parameters $\mathbf{w}_l$ which is related to $\mathbf{w}_k$ (Eq.(8)). The dynamic graph becomes stable once all the client models converge and correlated.

## 4.2 Federated Training

In this section, we present the training procedure of FedMSplit. We show that the federated multitask learning over the multimodal split networks of clients can be done in a way like learning the client embeddings in $\mathcal{G}(t) = (\mathcal{V}, \Phi(t), \mathcal{E}, \Lambda(t), q)$ through multiple rounds ($t = 1, 2, ..., T$) of client-server interactions until convergence.

### 4.2.1 *Alternative Optimization on Subgraphs via Client-Server Communication.*

The convergence of $\Phi$ is achieved by multiple rounds of alternative optimization. However, due to communication cost and the systematic challenge [19], it is consuming as well as impossible to calculate the complete correlation tensor $\Lambda(t)$ for all the clients and update all the client models $\Phi(t)$–in each round we would have $O(N^2)$ time and space complexity. Therefore, instead of computing all clients, we sample a subset of clients $\mathcal{S}(t) \subset \mathcal{V}$ to participate at each round $t$. In other words, at each round $t$, we select a subgraph $\mathcal{G}_s(t)$ of $\mathcal{G}(t)$ to perform alternative optimization. In particular, we consider a subgraph of clients and their relationships $\mathcal{G}_s(t) = (\mathcal{S}(t), \Phi_s(t), \mathcal{E}_s, \Lambda_s(t), q)$ where $\Phi_s(t) = \{\mathbf{w}_k(t)\}_{k\in\mathcal{S}(t)}$ and $\Lambda_s(t) \in \mathbb{R}^{|\mathcal{S}(t)|\times|\mathcal{S}(t)|\times(M+1)}$. Instead of optimizing on the entire graph at each round, our method is more practical in real operation–complexity is $O(C^2)$ where $C = |\mathcal{S}(t)| << N$, but note that we somewhat tradeoff the convergence speed because only partial clients and their relationships are considered while neglecting other related clients.

At each round, we perform one-step alternative optimization on subgraph $\mathcal{G}_s(t)$ through the following client-server communication. (1) On the server, we update $\mathbf{\Lambda}_s(t)$ fixing $\mathbf{\Phi}_s(t)$ and then propagate among the separate client models over subgraph $\mathcal{S}(t)$ as in Eq.(8). Note that in practice, the two steps can be replaced by applying a multi-head attention mechanism to node propagation among a subgraph–treating $\mathbf{\Lambda}_s(t)$ as the attention coefficients (see the next section for details). (2) The server then sends the aggregated relational information to each client. (3) On each client, we improve $\mathbf{\Phi}_s(t)$ on local datasets while considering potential client relationship $\mathbf{\Lambda}_s(t)$. Each client $k \in \mathcal{S}(t)$ perform local update by local SGD and multi-relational regularization (Eq.(9)), and, finally, all participants send their new models $\mathbf{\Phi}_s(t+1)$ back to the server. Figure 3 shows the overview of FedMSplit training at each round, and the pseudocode of FedMSplit is summarized in the Algorithm 1 in Appendix A.1.

#### 4.2.2 *Neighborhood-attentive Model Aggregation.*

Once the server receives new client models $\mathbf{\Phi}_s(t+1) = \{\mathbf{w}_k\}_{k \in \mathcal{S}(t)}$ (the models after performing $\tau$ steps of local updates), it adapts current model correlation to be able to promote more relational local models in future rounds. Since we use the normalized relationships among all the neighbors of client $k$ in Eq.(8), this model aggregation can be treated as 1-hop attentive message passing [3, 24, 38] among a subgraph. That is, for $\forall k \in \mathcal{S}(t)$,

$$\begin{cases} \overline{\mathbf{w}}_k^{agg} \leftarrow \sum_{l \in \mathcal{S}(t)} \frac{q(\mathbf{\Lambda}_{kl\cdot}) Att(\overline{\mathbf{w}}_k, \overline{\mathbf{w}}_j)}{\sum_{p \in \mathcal{S}(t)} q(\mathbf{\Lambda}_{kp\cdot}) Att(\overline{\mathbf{w}}_k, \overline{\mathbf{w}}_l)} \overline{\mathbf{w}}_l, \\ \mathbf{w}_{kj}^{agg} \leftarrow \sum_{l \in \mathcal{S}(t)} \frac{q(\mathbf{\Lambda}_{kl\cdot}) Att(\mathbf{w}_{kj}, \mathbf{w}_{lj})}{\sum_{p \in \mathcal{S}(t)} q(\mathbf{\Lambda}_{kp\cdot}) Att(\mathbf{w}_{kj}, \mathbf{w}_{pj})} \mathbf{w}_{lj} \text{ for } \forall j \in \mathcal{B}_k \cap \mathcal{B}_l, \end{cases} \tag{10}$$

where clients having similar statistics will become more related through the weighted model aggregation.

#### 4.2.3 *Client Sampling via Multi-armed Bandit.*

To reduce communication cost of FMTL, we operate on a subset of client at each round. Yet the random sampling strategy (i.e., unbiased client selection) of typical FL frameworks may significantly suffer from non-IID local distributions as well as the multimodal discrepancy of MFL. The problem is that the clients selected at each round (i.e, a subset of vertices in the large-scale graph and perform message passing) may not contain balanced numbers of each type blocks, thus we may not efficiently find the accurate correlations between different types of clients and modalities.

In order to achieve faster convergence, we aim to select clients having larger local loss (i.e., exploitation) [4, 35] as well as having blocks that were less frequently seen before (i.e., exploration). Following [4], to balance the exploration-exploitation trade-off in the multimodal client selection problem. we employ Multi-Armed Bandit (MAB) algorithms [15] for the problem of client selection in Multimodal FL. Regarding the local loss of individual clients are non-stationary during training, we make use of the discounted MAB algorithms as in [4]. The clients are viewed as *arms* in the MAB problem. The discounted cumulative local loss of each client is $L_k(t) = \sum_{t'=1}^{t} \gamma^{t-t'} F_k(t')$; the discounted number of times each client has been selected over the previous rounds is $I_k(t) = \sum_{t'=1}^{t} \gamma^{t-t'} 1_{k \in \mathcal{S}(t')}$; and, the discounted number of times each type of block $j$ has been sampled over previous rounds is, $P_j(t) = \sum_{t'=1}^{t} \gamma^{t-t'} 1_{j \in \mathcal{B}_k \ \forall k \in \mathcal{S}(t')}$. Here, $0 \leq \gamma \leq 1$ is the discount rate.

Then, we define the estimated UCB reward of client $k$ up to round $t$ as

$$A_k(t) = L_k(t)/I_k(t) + U_k(t) \tag{11}$$

where $U_k(t) = \sqrt{\sum_{r=1}^{t} \gamma^{t-r} / (I_k(t) + \sum_{j \in \mathcal{B}_k} P_j(t))}$ is the exploration term for client $k$. At communication round $t$, we select the top $C$ clients with largest discounted UCB rewards. The first term of Eq.(11) enforces selecting clients with estimated larger local loss (exploitation) [4]. However, if certain client has not been selected recently, or any type of model block of the client has not been selected recently, $U_k(t)$ will get larger. This forces the server to select them regardless of their local loss values (exploration).

# 5 EXPERIMENTS

We perform empirical study on multimodal federated datasets with the aim of answering two research questions: 1) How does FedMSplit perform for multimodal clients compared with baselines under different settings of statistical and modality incongruity? 2) How does each component of FedMSplit impact the performance?

## 5.1 Multimodal Federated Datasets

We choose three multimodal integration datasets to create our simulation environments. (1) ***Vehicle Sensor*** [8] for classifying vehicles driving by a segment of road. It contains 23 instances. Each instance is a separate client described by 50 acoustic and 50 seismic features and we predict between AAV-type and DW-type vehicles. (2) ***ModelNet40*** [34] dataset for *multi-view 3D object recognition* tasks. It contains 12,311 3D shapes covering 40 common categories, including airplane, bathtub, bed, bookshelf, chair, cone, cup, and so on. Each 3D CAD object has $M = 2$ modalities as two views of its shapes [10]. (3) ***IEMOCAP*** [1] for *emotion recognition tasks.* It consists of a collection of 4,453 video segments of recorded dialogues. Each segment is annotated for the presence of 9 emotions (happy, angry, excited, fear, etc.), from which we use only the "happy" tag for binary classification. We adopted the same feature extraction scheme [37] for language, visual and acoustic modalities. The feature sizes of the modalities are summarized in Table 1.

#### 5.1.1 *Simulation of Statistical Heterogeneity among Clients.*

We study the effectiveness of FedMSplit on non-IID data. We simulate the non-IID scenarios following [28]. The size of training samples at each clien $k$ is sampled from a Gaussian distribution whose mean and standard deviation is pre-defined as in Table 1.

#### 5.1.2 *Simulation of Modality Incongruity among Clients.*

We impose no restrictions on the modality or combinations of modalities used in the local clients. We simulate this real-world scenario as follows. First, we assume the availability of each sensor $j$ follows a Bernoulli distribution Bernoulli$(\rho_j)$ and different sensors are independent. Here, we use a missing rate $\rho_j$ to indicate the probability a client does not have the modality-$j$. We set equal missing rates for each modalities $\rho_1 = \ldots = \rho_M = \rho$ in all experiments. After that, we shuffle the clients and for each possible sensor set $\mathcal{B}' \subset \{1 \ldots M\}$ we separately pick $N(\mathcal{B}', \rho)$ clients and assign the sensor set $\mathcal{B}'$ to each of them. For example, for IEMOCAP dataset ($M = 3$), there will be 7 types of clients focusing on different tasks: audio-only, text-only, video-only, audio-text, audio-visual, text-video, and audio-text-visual tasks.

**Table 1: Statistics of Multimodal Federated Datasets for Simulation.**

| Dataset | Clients ($N$) | Sensors ($M$) | Feature Sizes ({modality $j : D_j$}) | $\lvert\mathcal{B}_k\rvert$ (range) | $n_k$ (mean, std) | Classes |
|---|---|---|---|---|---|---|
| Vehicle Sensor | 23 | 2 | {Acoustic: (50) , Seismic: (50)} | $[1, 2]$ | $\mathcal{N}(255, 50)$ | 2 |
| ModelNet40 | 12 | 2 | {View 1: (4096), View 2: (2048)} | $[1, 2]$ | $\mathcal{N}(1026, 200)$ | 40 |
| IEMOCAP | 15 | 3 | {Acoustic: (74) , Text: (300) , Visual: (35)} | $[1, 3]$ | $\mathcal{N}(297, 80)$ | 2 |

**Table 2: Average Testing Accuracy (%) on Client Local Testing Data at global round $T$=10 (Non-IID, C=0.3N).**

| | **Vehicle Sensor (1 or 2 sensors)** | | | **ModelNet40 (1 or 2 sensors)** | | | **IEMOCAP (1, 2, or 3 sensors)** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | $\rho$ =0.5 | $\rho$ =0.7 | $\rho$ =0.8 | $\rho$ =0.3 | $\rho$ =0.5 | $\rho$ =0.7 | $\rho$ =0.5 | $\rho$ =0.7 | $\rho$ =0.9 |
| FedAvg [23] | 75.26 | 74.42 | 68.48 | 87.48 | 86.79 | 83.48 | 80.48 | 79.71 | - |
| Multi-FedAvg [23] | 76.98 | 73.61 | 75.59 | 85.75 | 75.46 | 74.60 | 70.86 | - | 60.48 |
| Multi-FedProx [17] | 76.92 | 74.69 | 72.52 | 93.48 | 74.57 | - | 79.62 | 79.33 | 79.14 |
| Local | 74.84 | 73.56 | 69.29 | 91.56 | 90.20 | 83.18 | 73.24 | 73.71 | 54.48 |
| MOCHA [28] | 80.28 | 76.65 | 73.28 | 90.03 | 95.88 | 88.54 | 82.38 | 82.10 | 82.86 |
| Multi-MOCHA [28] | 78.35 | 76.61 | 75.73 | **98.25** | 96.06 | 90.70 | 80.95 | 80.19 | 80.86 |
| **FedMSplit** | **81.92** | **78.85** | **77.68** | **98.34** | **98.54** | **98.38** | **85.24** | **84.16** | **84.95** |

## 5.2 Baselines

We compare FedMSplit with three categories of baselines: (1) *Fully global* and *multimodal* FL frameworks: **FedAvg**, **Multi-FedAvg**, and **Multi-FedProx**, where we apply the vanilla FedAvg [23] and FedProx [17] to our multimodal federated datasets. (2) *Fully local* training on *multimodal* federated datasets: namely **Local**. We separately train local models that have different building blocks, without considering their potential relationships. (3) *Local but globally related multimodal* FL methods: **MOCHA** and **Multi-MOCHA**. The details of baseline implementations can be found in Appendix A.2.

## 5.3 Empirical Results

### 5.3.1 *Impact of Modality Incongruity.*

Table 2 reports the average local testing accuracy of FedMSplit compared with baselines, under different levels of modality incongruity and non-IID scenarios. We report the performance of the global model (FedAvg, Multi-FedAvg, and Multi-FedProx) or the globally stored separate models (Local, MOCHA, Multi-MOCHA, and FedMSplit) on all the clients' local testing data. From Table 2, we can observe that, in general, the increasing modality incongruity between clients results in performance drops of all methods. It is because as $\rho$ increases, models receive less information used for fitting parameters. Overall, FedMSplit gained more advantages over baselines as more clients have missing modalities and more local models have inactive neurons. On ModelNet40, FedMSplit maintains its performance as $\rho$ increases. It is because FedMSplit does not train inactive neurons as well as did not aggregate parameters as FedAvgs, FedProxs and FedMTLs. In FedMSpit, inactive neurons or blocks are not uploaded to the server and do not influence future models of other clients. Moreover, FedMSplit outperforms Local as well, even though Local train the same local architectures as ours. It is because in comparison to Local, the client models in FedMSplit can obtain knowledge about the task from other clients' data.

### 5.3.2 *Ablation Study.*

In Table 3, we evaluate the influence of each component in our model. (1) **Adaptive Correlation v.s. Non-adaptive Correlation**. First, we test a variant of FedMSplit, namely FedMSplit-nAC, such that we do not learn an adaptive correlation tensor between clients; instead, we assume that the relationships between clients is given (i.e., identity matrix). That is, at each round, all the participants having the same block contribute equally to each other. We can observe that FedMSplit-nAC still outperform baselines since we split each model into blocks and avoid transferring inactive neurons that are corresponding to missing sensors. In addition, arbitrarily fixing the relationship (FedMSplit-nAC) leads to slightly performance drop rather than adapting the relationships. (2) **Impact of Multi-view Relationship Measurement**. Second, since in our model the clients relationships are measured as the linear combination of each block's relationships, we are interested in whether each block in the model contributes equally to such measurement or not. In general, we use the measurement function $q(\Lambda_{k,l,\cdot}) = ||\Lambda_{k,l,\cdot}||_1/(1 + |\mathcal{B}_k \cap \mathcal{B}_l|)$, where the classifier and modality-specific feature extractors have similar importance. We then tested other types of measurements: FedMSplit-rightAC, which measures two models based only on their classifier weights $q(\Lambda_{k,l,\cdot}) = \Lambda_{k,l,(M+1)}$, and FedMSplit-leftAC, which measures two models based only on their common feature extractors $q(\Lambda_{k,l,\cdot}) = \sum_{j\in\mathcal{B}_k\cap\mathcal{B}_l} \Omega_{j,kl}/|\mathcal{B}_k \cap \mathcal{B}_l|$. It can be observed that the equally weighed measurement achieved the best performance, as the local data may have not only different classes but also low-level appearance nuances. (3) **Impact of Client Selection Strategy**. We finally tested our bandit-based client sampling strategy that encourages the server to explore clients who have blocks that are

**Table 3: Ablation study of FedMSplit at global round T=10.**

| | **Vehicle Sensor** | | **ModelNet** | |
|---|---|---|---|---|
| **Method** | $\rho$ =0.5 | $\rho$ =0.7 | $\rho$ =0.5 | $\rho$ =0.7 |
| **FedMSplit-nAC** | 80.12 | 75.32 | 98.43 | 97.63 |
| **FedMSplit-rightAC** | 79.88 | 76.82 | 96.20 | 97.67 |
| **FedMSplit-leftAC** | 78.32 | 77.48 | 97.79 | 96.37 |
| **FedMSplit-$\pi_{\text{rand}}$** (C=0.3N) | 80.47 | 77.36 | 96.62 | 94.33 |
| **FedMSplit-$\pi_{\text{UCB}}$** (C=0.3N) | **81.92** | **78.85** | **98.54** | **98.38** |

less selected. We propose a variant FedMSplit-$\pi_{\text{rand}}$, which replaces MAB with the random client selection strategy, and observed that random selection converged slower than the bandit-based counterpart, especially with a high level of modality incongruity.

## 6 CONCLUSION

In this paper, we addressed a novel multimodal FL problem with modality incongruity among clients. We introduced the FedMSplit framework, which allows federated training over multimodal distributed data without assuming similar active sensors in all clients. We employed a dynamic graph structure to capture the adaptive correlations amongst multimodal client models that have been split into smaller shareable blocks. The underlying statistical correlations between the different types of clients are captured as multi-view features and then are used to promote model relations. Our empirical results demonstrated the effectiveness of our method.

## ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is supported in part by the US National Science Foundation under grants 2106913, 2008208, 1955151, 1934600, and 1938167. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

[1] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation* 42, 4 (2008), 335.

[2] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[3] Jiayi Chen and Aidong Zhang. 2020. HGMF: Heterogeneous Graph-based Fusion for Multimodal Data with Incompleteness. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1295–1305.

[4] Yae Jee Cho, Samarth Gupta, Gauri Joshi, and Osman Yağan. 2020. Bandit-based communication-efficient client selection strategies for federated learning. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1066–1069.

[5] Luca Corinzia, Ami Beuret, and Joachim M Buhmann. 2019. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268* (2019).

[6] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. 2020. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848* (2020).

[7] Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. 2021. A New Look and Convergence Rate of Federated Multi-Task Learning with Laplacian Regularization. *arXiv preprint arXiv:2102.07148* (2021).

[8] Marco F Duarte and Yu Hen Hu. 2004. Vehicle classification in distributed sensor networks. *J. Parallel and Distrib. Comput.* 64, 7 (2004), 826–838.

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).

[10] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3558–3565.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1126–1135.

[12] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. 2021. Local sgd: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3556–3564.

[13] Farzin Haddadpour and Mehrdad Mahdavi. 2019. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425* (2019).

[14] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. 2020. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4519–4529.

[15] Volodymyr Kuleshov and Doina Precup. 2014. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028* (2014).

[16] Rui Li, Fenglong Ma, Wenjun Jiang, and Jing Gao. 2019. Online federated multitask learning. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 215–220.

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[18] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189* (2019).

[19] Fenglin Liu, Xian Wu, Shen Ge, Wei Fan, and Yuexian Zou. 2020. Federated learning for vision-and-language grounding problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11572–11579.

[20] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. 2018. Efficient Low-rank Multimodal Fusion With Modality-Specific Factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2247–2256.

[21] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. 2021. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems* 34 (2021).

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[23] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*.

[24] Giannis Nikolentzos, Antoine Tixier, and Michalis Vazirgiannis. 2020. Message passing attention networks for document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8544–8551.

[25] Verónica Pérez-Rosas, Rada Mihalcea, and Louis-Philippe Morency. 2013. Utterance-level multimodal sentiment analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 973–982.

[26] Yasmin SarcheshmehPour, Yu Tian, Linli Zhang, and Alexander Jung. 2021. Networked Federated Multi-Task Learning. *arXiv preprint arXiv:2105.12769* (2021).

[27] Abhishek Singh, Praneeth Vepakomma, Otkrist Gupta, and Ramesh Raskar. 2019. Detailed comparison of communication efficiency of split learning and federated learning. *arXiv preprint arXiv:1909.09145* (2019).

[28] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated multi-task learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 4427–4437.

[29] Yale Song, Louis-Philippe Morency, and Randall Davis. 2012. Multimodal human behavior analysis: learning correlation and interaction across modalities. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. 27–30.

[30] Sebastian Urban Stich. 2019. Local SGD Converges Fast and Communicates Little. In *ICLR 2019-International Conference on Learning Representations*.

[31] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2021. Towards personalized federated learning. *arXiv preprint arXiv:2103.00710* (2021).

[32] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, Seyit Camtepe, and Lichao Sun. 2020. Splitfed: When federated learning meets split learning. *arXiv preprint arXiv:2004.12088* (2020).

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[34] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.

[35] Wenchao Xia, Tony QS Quek, Kun Guo, Wanli Wen, Howard H Yang, and Hongbo Zhu. 2020. Multi-armed bandit-based client scheduling for federated learning. *IEEE Transactions on Wireless Communications* 19, 11 (2020), 7108–7123.

[36] Baochen Xiong, Xiaoshan Yang, Fan Qi, and Changsheng Xu. 2022. A Unified Framework for Multi-modal Federated Learning. *Neurocomputing* (2022).

[37] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1103–1114.

[38] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. 2020. Dynamic graph message passing networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3726–3735.

[39] Yuchen Zhao, Payam Barnaghi, and Hamed Haddadi. 2021. Multimodal federated learning. *arXiv preprint arXiv:2109.04833* (2021).

# A APPENDIX

## A.1 Pseudocode of FedMSplit

**Algorithm 1** Federated Training Algorithm of FedMSplit

```
Input: clients $k \in [N]$, modalities $j \in [M]$, multimodal datasets $\mathcal{D}_1, ..., \mathcal{D}_N$ and active sensor sets $\mathcal{B}_1, ..., \mathcal{B}_N$ of clients
Hyper-parameters: $\eta$,$T$,$\tau$, $C$, $\gamma$
For each client $k$ in parallel, initialize $\mathbf{w}_k$ and split the model into $|\mathcal{B}_k| + 1$ blocks as Eq.(4).
Initialize $I_k(0) = P_k(0) = L_k(0) = 0$
Initialize $\mathbf{w}_k^{agg} = \mathbf{w}_k$
for each round $t = 1, ..., T$ do
    Sampling a subset of $C$ clients $\mathcal{S}(t) \subset [N]$ using Eq.(11)
    // local SGD independently
    for each participant client $k \in \mathcal{S}(t)$ do
        for each step $r = 1, ..., \tau$ do
            update $\mathbf{w}_k$ as Eq.(9)
        end for
    end for
    Send $\{\mathbf{w}_k\}_{k \in \mathcal{S}(t)}$ to the server
    // adapt model correlation via attentive aggregation
    for each participant client $k \in \mathcal{S}(t)$ do
        Split $\mathbf{w}_k$ into blocks
        for each related client $l \in \mathcal{S}(t)$ do
            Calculate attention weight for each pairs of blocks between $\mathbf{w}_k$ and $\mathbf{w}_l$.
        end for
        Obtain aggregated model $\mathbf{w}_k^{agg}$ using Eq.(10)
    end for
    Server sends $\{\mathbf{w}_k^{agg}\}_{k \in \mathcal{S}(t)}$ to clients.
    Update $I_k(t), P_k(t), L_k(t)$ using the selected clients in $\mathcal{S}(t)$ and counting each type of blocks in $\mathcal{S}(t)$.
end for
return: Each client will store its final model $\mathbf{w}_k$.
```

## A.2 Reproducibility

The local objective of all models we used in the experiments is the cross entropy loss, i.e., $\min F_k(\mathbf{w}_k) = \sum_i y_i \log f_k(\mathcal{X}_i; \mathbf{w}_k)$.

In all experiments, we fix $\gamma = 0.9$, $\tau = 4$, and $\eta = 0.005$ for Vehicle Sensor and ModelNet40; and fix $\tau = 1$, $\eta = 0.00002$ for IEMOCAP.

Model configurations are as follows. In the following, the encoded modality's hidden dimension is $P = 32$ for vehicle sensor dataset, $P = 128$ for ModelNet , $P = 64$ for IEMOCAP.

### A.2.1 *Fully global multimodal FL baselines.*

We train a global model having all blocks and the missing modalities on local sites are imputed as zero.

- **FedAvg** [23]: The global model consists of 2 layers: $\{\sum_{j \in M}$in_dim$_j$ × P, P × num_class}. For local model training, we replace the missing sensor data by zeros and then directly concatenate the input modalities into one feature. For example, for IEMOCAP dataset and a client that only has audio features, the input features would be:
  $((0.34, 0.43, ..., 0.98), NaN, NaN) \rightarrow$
  $((0.34, 0.43, ..., 0.98), (0, 0, ...., 0), (0, 0, ..., 0)) \rightarrow$
  $(0.34, 0.43, ..., 0.98, 0, 0, ...., 0, 0, 0, ..., 0)$
- **Multi-FedAvg** [23]: we use individual feature extractors for each modality (i.e., in_dim$_j \times P$) followed by a classifier (a fully connected layer of size $P \times$ num_class followed by Softmax). The output of feature extractors (modality-specific hidden representations) are combined using sum operation. Different from FedAvg, the input features of IEMOCAP is formed as:
  $((0.34, 0.43, ..., 0.98), NaN, NaN) \rightarrow$
  $((0.34, 0.43, ..., 0.98), (0, 0, ...., 0), (0, 0, ..., 0))$
- **Multi-FedProx** [17]: Model architecture is the same as Multi-FedAvg. The difference is that, for local training, we add a regularization term $||\mathbf{w}_k - \mathbf{w}_k(t)||_2^2$.

### A.2.2 *Fully local training on multimodal federated datasets.*

We separately train local models that have different building blocks, without considering their potential relationships.

- **Local**: The model is the partial architecture of Multi-FedAvg or Multi-FedProx, including the feature extractors for only available modalities (i.e., in_dim$_j$ × P) followed by a classifier (a fully connected layer of size $P \times$ num_class followed by Softmax). Note that there is no feature extractor block in the clients having missing modalities. For IEMOCAP dataset and a client that only has audio features, in comparison to the above example of FedAvg, the input feature of the model would be formed as:
  $((0.34, 0.43, ..., 0.98), NaN, NaN) \rightarrow$
  $(0.34, 0.43, ..., 0.98)$.

### A.2.3 *Local but globally related multimodal FL framework.*

- **MOCHA** [28]: although we learn models for each client, in this baseline, the local model architectures are still the same. We define the architecture similar to FedAvg, and the input features look like:
  $((0.34, 0.43, ..., 0.98), NaN, NaN) \rightarrow$
  $((0.34, 0.43, ..., 0.98), (0, 0, ...., 0), (0, 0, ..., 0)) \rightarrow$
  $(0.34, 0.43, ..., 0.98, 0, 0, ...., 0, 0, 0, ..., 0)$
- **Multi-MOCHA** [28]: model architecture is similar to Multi-FedAvg. We use individual feature extractors but let the inactive weights of local models be transferred among clients. The input features of IEMOCAP is formed as:
  $((0.34, 0.43, ..., 0.98), NaN, NaN) \rightarrow$
  $((0.34, 0.43, ..., 0.98), (0, 0, ...., 0), (0, 0, ..., 0))$

### A.2.4 *FedMSplit Reproducibility.*

The model architectures are the same as the Local, which are partial architectures of the complete model architecture of other methods. For IEMOCAP dataset and a client that only has audio features, similar to Local, the input feature of the model would be look like:

$((0.34, 0.43, ..., 0.98), NaN, NaN) \rightarrow$
$(0.34, 0.43, ..., 0.98)$.