MultimodalHD: Federated Learning Over Heterogeneous Sensor Modalities using Hyperdimensional Computing

Quanling Zhao Xiaofan Yu Shengfan Hu Tajana Rosing
Department of Computer Science and Engineering, University of California San Diego
{quzhao,x1yu,shh042,tajana}@ucsd.edu

***Abstract*—Federated Learning (FL) has gained increasing interest as a privacy-preserving distributed learning paradigm in recent years. Although previous works have addressed data and system heterogeneities in FL, there has been less exploration of *modality heterogeneity*, where clients collect data from various sensor types such as accelerometer, gyroscope, etc. As a result, traditional FL methods assuming uni-modal sensors are not applicable in multimodal federated learning (MFL). State-of-the-art MFL methods use modality-specific blocks, usually recurrent neural networks, to process each modality. However, executing these methods on edge devices proves challenging and resource-intensive. A new MFL algorithm is needed to jointly learn from heterogeneous sensor modalities while operating within limited resources and energy. We propose a novel hybrid framework based on Hyperdimensional Computing (HD) and deep learning, named *MultimodalHD*, to learn effectively and efficiently from edge devices with different sensor modalities. MultimodalHD uses a static HD encoder to encode raw sensory data from different modalities into high-dimensional low-precision hypervectors. These multimodal hypervectors are then fed to an attentive fusion module for learning richer representations via inter-modality attention. Moreover, we design a proximity-based aggregation strategy to alleviate modality interference between clients. MultimodalHD is designed to fully utilize the strengths of both worlds: the computing efficiency of HD and the capability of deep learning. We conduct experiments on multimodal human activity recognition datasets. Results show that MultimodalHD delivers comparable (if not better) accuracy compared to state-of-the-art MFL algorithms, while being 2x - 8x more efficient in terms of training time. Our code is available online[1].**

I. Introduction

With recent advancements in machine learning and edge computing platforms, Federated learning (FL) has emerged as a popular approach for distributed training and Internet-of-Things (IoT) deployments. Although previous research has explored addressing challenges such as data heterogeneity (e.g., non-iid data distribution on clients [1]), system heterogeneity (e.g., varied computational and communication delays [2]), and unexpected stragglers (e.g., client drops due to various types of failure [3]) in FL, there has been limited exploration in *Multimodal Federated Learning (MFL)*. In contrast to uni-modal FL, which assumes a single sensor modality and an identical model architecture on all clients, Multimodal FL considers heterogeneous sensor modalities - a more realistic setting because not all edge devices possess the exact same sensors. For example, gyroscope, accelerometer and magnetometer can all be used to monitor human activity, but not all of them may be available on one client (Fig. 1). Traditional FL frameworks such as FedAvg [4] assume uni-modal sensors and uniform model architectures, struggling with the heterogeneous sensor modalities among clients. State-of-the-art MFL works use sophisticated neural networks to address modality heterogeneity, such as deep canonically correlated autoencoders [5] and split neural networks [6], thus incurring high computational costs. These designs are unsuitable for edge devices due to their limited computational capabilities, network connectivity, and reliance on battery power. Therefore, a new and lightweight FL algorithm is needed to learn both *effectively* and *efficiently* under arbitrary modality combinations at the edge.

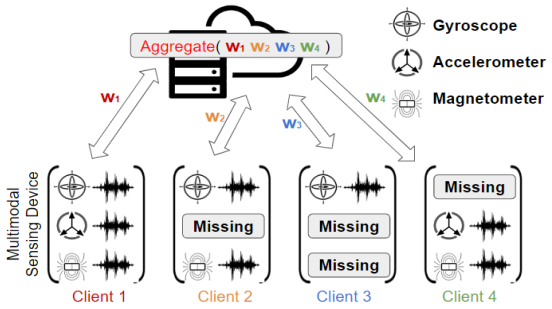

Fig. 1: An example of the **Multimodal FL** scenario with 3 different sensor modalities across 4 clients. The clients only exchange model weights with the cloud, which aggregates these weights.

Hyperdimensional Computing (HD) is a new brain-inspired computing paradigm that encodes data into high-dimensional, often low-precision vectors known as *hypervectors*. Cognitive tasks, including classification, can be executed in the HD space using a set of simple operations. Compared to traditional neural networks (NNs), HD-based designs have demonstrated comparable accuracy in various applications while saving magnitudes of execution time and energy [7]–[11]. In short, the efficiency of HD makes it a suitable candidate for edge applications; simultaneously, HD space provides a new opportunity to manage multiple modalities. Prior studies have investigated HD multimodal fusion by adding the hypervectors

[1]https://github.com/QuanlingZhao/DATE-24-MultimodalHD

of different modalities and applying standard HD training [12], [13]. However, this approach failed to take advantage of inter-modality dynamics, a factor proved to be crucial in deep learning-based designs [6], [14], [15].

In this paper, we propose a novel hybrid framework named *MultimodalHD* that combines the efficiency of HD and the capability of deep learning (DL). MultimodalHD utilizes a static HD encoder to encode multimodal time series data into hypervectors. We then design a novel attention module which fuses hypervectors with inter-modality correlations. Furthermore, we devise a proximity-based aggregation strategy in the cloud to alleviate interference between clients. Although our method is applicable to a variety of MFL applications, in this paper we focus on human activity recognition (HAR) tasks. HAR naturally comes with multimodal sensors (accelerometer, gyroscope, etc.) and is often performed on small mobile devices. Therefore, HAR serves as an exemplary use case that requires effective MFL within stringent resource constraints.

In summary, MultimodalHD is the first work that integrates HD and DL designs for effective and efficient MFL:

- MultimodalHD uses HD encoder to efficiently extract information from multimodal time-series sensor data, bypassing traditional recurrent neural networks (RNNs).
- We include two novel DL components to improve multimodal representation learning and alleviate modality interference: attention-based fusion on local clients and proximity-based aggregation on the cloud.
- Our evaluation of three HAR datasets shows that MultimodalHD is 2x - 8x more efficient in training time compared to state-of-the-art multimodal FL baselines using RNNs, while ensuring comparable or better accuracy.

II. Related Works

Multimodal Federated Learning (MFL). Learning from multimodal data in a federated setting has gained significant interest in recent years. In contrast to traditional FL which only focus on training uni-modal model, MFL adds complexity in model aggregation due to the presence of modality heterogeneity among clients.

Multimodal-FL [5] employs a split autoencoder on each client to learn from multiple modalities without supervision. CreamFL [16] uses inter and intra-modal contrastive loss to complement information about the absent modality. However, neither of the works incorporates personalized models to accommodate client-specific patterns. MMFL [15] enables personalization with a meta learning-based approach; however, its co-attention mechanism can only fuse between two modalities. FedMSplit [6] and Harmony [17] both partition client models into modality-specific blocks to harness modality heterogeneity. Each of these methods employs an individual RNN as a feature extractor for each modality, leading to high training costs and challenges in parallelization due to the sequential nature of RNNs. In contrast, our design MultimodalHD allows multimodal federated personalized learning while excelling in efficiency. A detailed comparison is provided in Table I.

TABLE I: Comparing MultimodalHD and state-of-the-art Multimodal Federated Learning works.

Method	Modality Heterogeneity	Personalization	Hardware Efficiency
[5], [16]	X		
[15]	Limited	X	
[6], [17]	X	X	
MultimodalHD	X	X	X

Hyperdimensional Computing. Although HD has been successfully applied in various scenarios [7]–[11], HD-based multimodal or federated learning are less visited. HDCMER [12] and Schelegel *et al.* [13] bundle the encoded hypervectors from different modalities for fusion, and use the fused hypervectors for emotion recognition and driving style classification respectively. FHDnn [18] and FedHD [14] enable FL by sharing a fixed HD encoder among all clients, learning a HD hypervector for each class on local clients, and aggregating the class hypervectors averagely in the cloud. To the best of the authors' knowledge, MultimodalHD is the first HD-based design for MFL.

III. Primitives

We first introduce the MFL problem definition, HD primitives and our motivation for attention-based multimodal fusion.

A. Multimodal Federated Learning: Problem Definition

We consider a supervised MFL problem with personalization. To model a realistic heterogeneous MFL setting, we pose no restriction on the number of modalities on a certain client. Let C_k denote a client for $k \in \{1, \dots, N\}$. Specifically, $C_k = \{D_k, \ , w_k\}$ where D_k is the labeled multimodal dataset on client k, is an HD encoder shared among all clients, and w_k denotes personalized model weights on client k. Suppose B is the set of all modalities in the system, B_k is the set of locally available modalities on client k with $B_k \subset B$. Assuming client k has n_k local data samples, let $D_k = \{(X_i, y_i)\}_{i=1}^{n_k}$ be the local multimodal dataset where $X_i = \{x_i^{(j)} | \forall j \in B_k\}$ and y_i are the raw multimodal sample and the label respectively. Each $x_i^{(j)}$ in a sample X_i represents time-aligned uni-modal sensor readings within a sliding time window of length T. Following [6], we set the objective of our MFL problem as learning a set of different but correlated model weights $\{w_1, \dots, w_N\}; w_1 \neq w_2 \neq \dots \neq w_N$.

$$\min_{w_1, \dots, w_N} \sum_{k=1}^{N} \sum_{i=1}^{n_k} f(w_k, \ (X_i), y_i)) + R(w_k; w_1, \dots, w_N) \quad (1)$$

where $f(w_k, \ (X_i), y_i)$ is a loss function defined on model weights w_k, encoded hypervector (X_i) and true label y_i. $\mathcal{R}$ is a regularization term that forces a certain level of similarity between w_k and the models from other clients, thus encouraging positive knowledge sharing among clients.

B. HD Primitives

Hyperdimensional computing (HD) is a lightweight computing paradigm that encodes data into hypervectors. HD learning tasks can be performed through a set of simple arithmetic

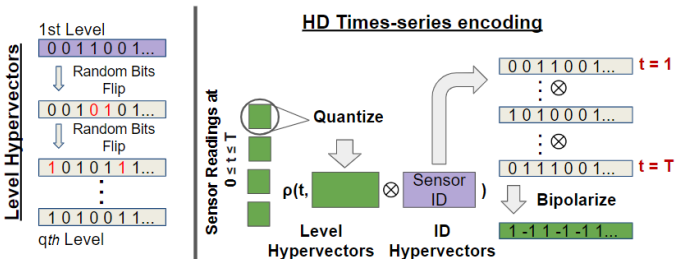

Fig. 2: Left: The generation process of level hypervectors. Right: The complete HD encoding process for time-series data.

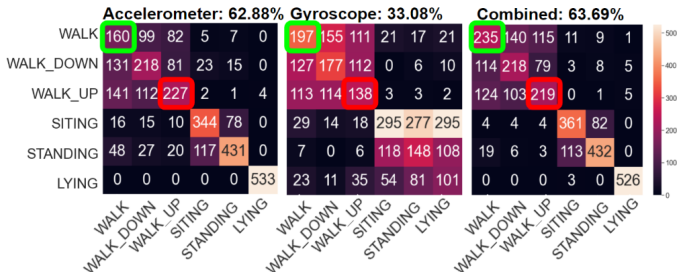

Fig. 3: The confusion matrices of the six activities in the HAR dataset [20] when using bundling as multimodal fusion in HD. The green boxes highlight the case where fusion improves classification, while the red boxes indicate when fusion degrades performance.

operations with excellent efficiency. Suppose the HD dimensionality is D. Associative learning is performed on hypervectors with the following well-defined operations: (1) *Bind*: $\otimes(\{0;1\}^D;\{0;1\}^D) \Rightarrow \{0;1\}^D$. Binding takes two hypervectors and returns a hypervector that is dissimilar to both operands. For binary hypervectors, binding is implemented via element-wise XOR. (2) *Bundle*: $\oplus(\mathbb{Z}^D;\mathbb{Z}^D) \Rightarrow \mathbb{Z}^D$. Bundling induces the notion of set in HD space as it returns a hypervector that is maximally similar to its constituting elements. Bundling is implemented via addition. (3) *Permute*: $(t;\{0;1\}^D) \Rightarrow \{0;1\}^D$. Permutation is implemented using logical shift, where t represents the number of shifts.

Time-series HD Encoding. The first step in HD is to encode raw data samples into hypervectors. The goal is to map high-precision, low-dimensional real-valued sensor readings to low-precision, high-dimensional hypervectors in HD space, while preserving the spatial and temporal patterns. Here, we use general encoding schemes for time series data, i.e. the spatial-temporal encoder [7], [19]. An explanation is provided below. To represent numeric values, we generate *level hypervectors*. We begin by quantizing the support of sensor reading into q bins, and each bin is represented with a level hypervector. Starting with a randomly generated binary hypervector representing the 1st level, each subsequent level can be generated by randomly flipping pD bits (where p denotes the flipping rate) from the previous level. This approach allows for the quantization of sensor readings into hypervectors while preserving the underlying structure. Fig. 2 (left) illustrates the generation process of level hypervectors. Furthermore, another set of base hypervectors, known as the *ID hypervectors*, is randomly generated to represent different modalities.

The complete encoding process is shown in Fig. 2 (right). Consider the encoding of $x_i^{(j)}$, a time series of length T. The process begins by quantizing real-valued sensor readings, with each quantized value assigned to a level hypervector among $\overline{L}_1;:::;\overline{L}_q$. Subsequently, the level hypervectors are bound together with their corresponding ID hypervectors ID_j to encode the information of modality j. To capture temporal information, the bound hypervectors are permuted based on their corresponding temporal order t within the time window. Finally, all hypervectors across the temporal dimension are bound and bipolarized to produce the ultimate hypervector. Formally, the encoding of $x_i^{(j)}$ can be expressed as:

$$(x_i^{(j)}) = BP(\ (1;\overline{L_1}\quad ID_j)\quad :::\quad (T;\overline{L_T}\quad ID_j)) \qquad (2)$$

C. Challenges of using HD in MFL

One main challenge for MFL is learning joint representation by fusing the information from different sensing modalities. Previous HD literature [12], [13] proposed a method of simply bundling hypervectors from different modalities to form class hypervectors and utilizing the cosine similarity metric for classification. However, we argue that such a method does not fully exploit the potential of multimodal data. The simplistic bundling of hypervectors from various modalities assumes that all modalities are equally important in all situations. However, as many studies in the literature on multimodal learning have demonstrated, that is not the case [21]. To validate this, we conduct a simple experiment on the HAR dataset [20] with a standard HD classification pipeline [10] using the bundling operation for multimodal fusion. Fig. 3 depicts the confusion matrices during classification. The results show only marginal performance improvement and, in some cases, even degradation when incorporating new modalities. In order to capture the full inter-modality dynamics and complementary information, a more intelligent multimodal fusion method is needed, especially when modalities are many and diverse. Motivated by this observation, we propose MultimodalHD.

IV. Proposed Framework: MultimodalHD

In this section, we introduce our proposed framework, MultimodalHD, which features an innovative architecture combining HD and deep learning. The design philosophy is to use an HD encoder to map time-series data into feature space, replacing traditional RNNs, and use an attention mechanism for multimodal fusion. As shown in Fig. 4, MultimodalHD first encodes the multimodal time-series data into hypervectors using a shared HD encoder. Subsequently, multimodal information is fused together with a novel attentive fusion design. Finally, the fused multimodal representations are input to a multilayer perceptron (MLP) for classification. At the federated level, we propose a personalized aggregation strategy to alleviate modality interference due to modality heterogeneity in the cloud. Notably, the model architecture of MultimodalHD is designed in a modality-invariant way, meaning that all client models share the same architecture and parameter space, even in the presence of heterogeneous and unavailable modalities. We detail the two key designs in MultimodalHD: attentive multimodal fusion (Sec. IV-A) and aggregation (Sec. IV-B).

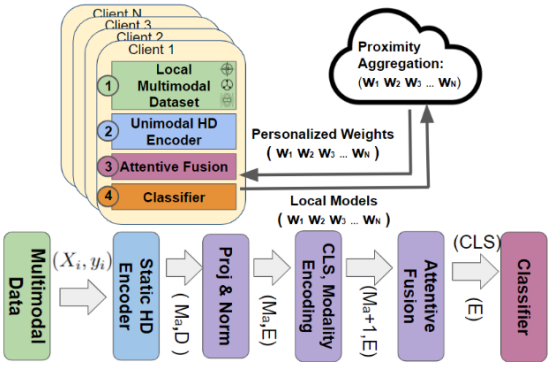

Fig. 4: Top: The overall structure of MultimodalHD. Bottom: The local computation pipeline of MultimodalHD on one client, where parentheses denote tensor size.

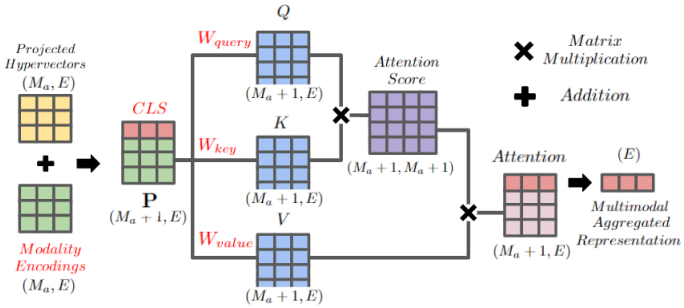

Fig. 5: The attention fusion module in MultimodalHD to fuse hypervectors from different sensing modalities. Parts highlighted in red are aggregated at the cloud.

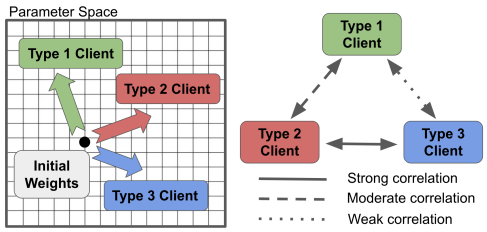

Fig. 6: An intuitive example of modality interference at the cloud that motivates the proximity-based aggregation.

A. Attentive Multimodal Fusion

Inspired by the attention mechanism [22], we adapt inter-modal attention to learn a fused representation from multimodal hypervectors. The self-attention mechanism facilities capturing dynamics in multimodal signals [23]. This empowers the model to intelligently combine information from different modalities, a capability not achievable with previous HD based methods [12], [13].

The lower portion of Fig. 4 presents the overview of local computational pipeline on a single client, while the detailed operations of inter-modality attention are shown in Fig. 5. Given M_a (number of available modality) hypervectors of dimension D, we first apply a trainable projection layer to reduce the dimensionality of the hypervectors to E. Gaining inspiration from the positional encoding in transformers [22], [24], we create modality encodings which are assigned to each sensing modality and added to the corresponding projected hypervectors. Unlike the original positional encodings which encode the position and order of inputs, the purpose here is to encourage the model to learn information associated with each modality itself rather than the data from that modality. Next, a classification (CLS) token, similar to the ViT [24] and BERT [25] architectures, are concatenated with the projected multimodal hypervectors before passing to the attention computation that involves $Q; K; V$ matrices (as shown in Fig. 5). The output of CLS token serves as an attentively aggregated representation of all modalities. Both the modality encodings and CLS token are implemented as trainable parameters (as part of w_k on client k) and are aggregated across clients. After projecting and adding modality/CLS encodings, we have a matrix of size $((M_a + 1) \times E)$ denoted as $\mathbf{P}$. The computation of attention is shown in Fig. 5, formally as:

$$Q = W_{query}\ \mathbf{P};\ K = W_{key}\ \mathbf{P};\ V = W_{value}\ \mathbf{P} \tag{3}$$

$$Attention = softmax(\frac{Q_p K^T}{\sqrt{E}})V \tag{4}$$

Here $W_{query;key;value}$ are trainable attention weight matrices. Note, that the dimension of attention weights only depends on the embedding dimension E, hence our attentive multimodal fusion module and classifier are invariant to the number of modalities on a client. This allows us to use a uniform model architecture across all clients in the presence of modality heterogeneity.

B. Proximity-based Cloud Aggregation

During the aggregation phase, the weights of the attention module ($W^{(m)}_{query;key;value}$, projection layer, CLS, modality encodings) and the MLP classifiers are exchanged. At the cloud level, we propose a new proximity-based cloud aggregation strategy to mitigate interference between clients. Client models are trained on different combination of modalities, thus they are likely to be optimized towards different sub-regions in the parameter space. We refer to this as *modality interference*.

Fig. 6 shows an intuitive example of modality interference between clients. It is more beneficial to encourage the aggregation between strongly correlated clients (type 2 and type 3) for complementary information and information redundancy as they share relatively more homogeneous modality. Weakly correlated clients (type 1 and type 2) is likely to result in degraded performance due to modality mismatch. However, quantifying modality interference between clients solely based on their available modalities is challenging, as we lack information about the modality's physical properties. Hence we propose an adaptive aggregation strategy at the cloud using model weights similarity as an approximate indicator of potential modality interference. Our method mitigates modality interference while allowing for personalization.

Let $\{w_1; :::; w_N\}$ denote the updated local models transmitted from clients to the cloud. Let $S^{cos}_{ij} = cos(w_i; w_j)$ represent the pairwise cosine similarity between client i and j's model parameters. At the cloud, we adaptively adjust the aggregating

weights w_i^{new} for each client i based on the softmax of S_{ij}^{cos}:

$$softmax(S_i^{cos})_j = \frac{\exp(\frac{S_{ij}^{cos}}{\tau})}{\sum_{k=1}^{N} \exp(\frac{S_{ik}^{cos}}{\tau})} \quad (5)$$

$$w_i^{new} = \sum_{j=1}^{N} softmax(S_i^{cos})_j \; w_j \quad (6)$$

Here τ is a temperature hyperparameter. Intuitively, our proximity-based aggregation strategy gives heavier weights to models from clients with similar modalities and suppresses modality interference between client pairs with dissimilar modalities (with a small S_{ij}^{cos}).

V. Evaluation

A. Experimental Setup

Datasets. We use three commonly used multimodal human activity recognition datasets with continuous sensor readings, HAR [20], MHEALTH [26] and OPPORTUNITY (OPP) challenge dataset [27]. The HAR dataset is collected with smartphones, containing time-series accelerometer and gyroscope readings of 30 subjects performing 6 common daily activities. The MHEALTH dataset is collected via wearable sensing devices, containing accelerometer, gyroscope and magenetometer data for 13 common activities. We use accelerometer and gyroscope data from the OPP dataset with 17 mid-level classes after removing the null class, following previous work [5]. The configuration details for different modalities are reported in Table II. We use $T = 128$ and split the datasets into individual multimodal time series samples with a 75% overlap. Federated experiments use a batch size of 64, total training rounds of 20, and local epochs of 2 (per communication round).

Baselines. In the MFL setting, we evaluate MultimodalHD in comparison to two representative state-of-the-art MFL methods. **Split-AE** [5] uses split-autoencoder to learn and extract correlated representations from different modalities. **FedMSplit** [6] uses separate blocks for available modalities on the clients and updates the global model based on a dynamically learned graph. We use 10 hidden units per LSTM block for both baselines. All methods are implemented using PyTorch. The important parameters in MultimodalHD are summarized in Table III. For all methods that require a classifier, we use a two-layer MLP with 25 hidden units.

Metrics. We use the weighed F1 score as the evaluation metric, following previous studies [5]: $F_1 = \frac{2\ TP}{2\ TP+FP+FN} \times 100$. The F1 score for one client is the weighted average of F1 scores for all classes. The overall performance is evaluated by the average F1 score across all clients. In terms of efficiency experiments, we measure and compare the training time per epoch on a Raspberry Pi model 4B.

B. Multimodal Federated Learning

We first experiment in the MFL setting where clients have different available sensory modalities as shown in Table II. The goal of personalized MFL is to achieve the best average F1 by utilizing information from different modalities and clients. Fig. 7 (top) shows the convergence of all methods across all three datasets. MultimodalHD achieves better/comparable final results compared to state-of-the-art MFL baselines in all scenarios. Specifically, MultimodalHD also converges faster with regard to communication rounds on the HAR dataset. Although FedMSplit ends up with slightly better results on the HAR and MHEALTH datasets, we emphasize that the performance of MultimodalHD is close to optimal with an efficiency advantage.

TABLE II: Sensor modality configurations in MFL on various datasets. **Acc., Gyr, Mag.**: Accelerometer, Gyroscope, and Magetometer sensors. #: Number of clients.

HAR [20]			**MHEALTH** [26]				**OPP** [27]		
Acc.	Gyr.	#	Acc.	Gyr.	Mag.	#	Acc.	Gyr.	#
X	X	10	X	X	X	3	X	X	2
×	X	10	X	×	X	3	X	×	2
X	×	10	X	X	×	4	-	-	-

TABLE III: Important parameters in MultimodalHD.

Param.	Description	Value (HAR, MHEALTH, OPP)
D	HD dimension	1000
E	Projected dimension	25
τ	Temperature in aggregation	$2e^{-4}$, $7e^{-4}$, $2e^{-3}$
q	Numner of quantization level	10, 100, 300
p	flipping rate	$1e^{-2}$, $2e^{-2}$, $1e^{-3}$

C. Effects of Different Federated Aggregation method

We fix the local model training pipeline and compare our proximity-based aggregation method with two commonly used aggragation methods: FedAvg [4] and FedPer [28]. FedAvg performs weighted averaging and FedPer allows personalized weights for final MLP layers. As shown in Fig. 7 (bottom), FedAvg produces the least satisfactory results on the HAR and MHEALTH datasets. This aligns with the modality interference issue discussed in Sec. IV-B, as FedAvg equally aggregates models trained on different modalities. FedPer partially fixes this issue by allowing personalized weights, specifically not overwriting the final layer during federated aggregation. Our proximity-based aggregation in MultimodalHD further improves on that by taking into account modality interference during aggregation, while also allowing personalization. The OPP dataset demonstrates similar results under three different aggregation strategies. This is attributed to OPP having less modality heterogeneity compared to HAR and MHEALTH.

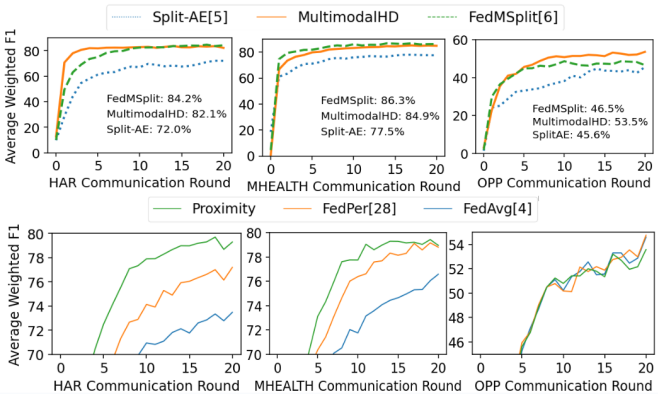

Fig. 7: Top: Average weighted F1 scores of MultimodalHD and all baselines under the MFL setting. Bottom: Effects of different aggregation methods in MultimodalHD.