# Harnessing Asynchrony to Balance Modalities in Multi-Modal Federated Learning

Yiming Ma[1,5], Boyi Liu[1], Zimu Zhou[2,3], Yanfeng Wang[4,5], and Yongxin Tong[1]

[1] State Key Laboratory of Complex & Critical Software Environment,
School of Computer Science, Beihang University, Beijing, China
{ma.yiming,boyliu,yxtong}@buaa.edu.cn
[2] Department of Data Science, City University of Hong Kong, Hong Kong, China
zimuzhou@cityu.edu.hk
[3] City University of Hong Kong Shenzhen Research Institute, Shenzhen, China
[4] School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China
wangyanfeng@sjtu.edu.cn
[5] Shanghai Artificial Intelligence Laboratory, Shanghai, China

**Abstract.** Multi-Modal Federated Learning enables clients to collaboratively train multi-modal models without sharing raw data. In practice, it suffers from modality laziness, where dominant modalities overshadow weaker ones, and asynchronous modality availability, where modalities arrive at clients at different times. Existing modality balancing methods assume synchronous access to all modalities in each round, making them unfit for asynchronous arrivals. We present MBA (Modality Balancing via Asynchrony), a lightweight framework that exploits asynchrony to combat modality laziness under feature-level fusion. First, clients perform opportunistic local balancing, where early-arriving modalities create uni-modal feature anchors to regularize multi-modal local updates without idle waiting. Then the server adopts balance-aware asynchronous aggregation, which estimates and corrects global modality imbalance via staleness-weighted updates. Experiments show that MBA improves both accuracy and efficiency, demonstrating that asynchrony can be harnessed to achieve balanced multi-modal federated learning.

**Keywords:** Federated Learning · Multi-Modal · Modality Laziness.

## 1 Introduction

Multi-Modal Federated Learning (MMFL) enables clients that hold multiple data types (*e.g.*, images, audio, text, sensors) to collaboratively train models without sharing raw data [9, 4]. By jointly leveraging complementary modalities, MMFL can achieve higher accuracy than its uni-modal counterpart and is particularly suitable for privacy-sensitive, sensor-rich edge and IoT deployments [20, 32, 31]. For example, modalities can be fused at the feature level [12, 8], where each client maintains modality-specific encoders and a shared fusion classifier with concatenated embeddings for multi-modal prediction. Such fusion balances flexibility and communication efficiency while avoiding direct exposure of raw features.

Despite its potential, MMFL faces two interacting challenges that degrade its performance in practice.

- **Modality laziness.** Joint optimization in multi-modal models often suffers from *modality laziness*, where stronger modalities dominate updates while weaker modalities remain under-trained [6]. Differences in convergence speed and gradient scale allow faster-learning modalities to steer the fused representation and impair the learning of others [13, 23, 22]. As training progresses, such cross-modal imbalance worsens, leading to biased representations and reduced overall accuracy.
- **Asynchronous modality availability.** In real-world deployments, modalities rarely arrive at the clients simultaneously. Heterogeneous sampling rates, buffering policies, and network fluctuations cause inputs to become available at different times [15, 26, 28, 17]. Waiting for all modalities before training avoids bias but wastes wall-clock time. Training immediately with partial modalities improves efficiency but exacerbates modality laziness by overemphasizing frequently available modalities.

The principle to mitigate modality laziness is to *balance* the training across modalities. In centralized training, this is achieved by gradient modulation [22], held-out reweighting [23], early-stopping [27], or prototype-based calibration [8]. A few pioneer studies [7] have addressed modality laziness in federated environments. However, all these studies assume synchronous access to all modalities per round, limiting their effectiveness under real-world asynchronous conditions.

In this paper, we harness asynchrony rather than treat it as an obstacle in handling modality laziness. We propose MBA (Modality Balancing via Asynchrony), a lightweight scheme that transforms early-arriving modalities into *opportunistic anchors* to estimate and correct cross-modality imbalance without waiting for full availability. MBA introduces a *two-tier* mechanism that integrates seamlessly with asynchronous federated training. *(i) Opportunistic Local Modality Balancing.* When only a subset of modalities is available, the client performs uni-modal local training and caches the corresponding features as *anchors.* Once all modalities arrive, multi-modal training proceeds with an anchor-based *regularizer* that discourages local gradient dominance and steers weaker modalities without adding wall-clock delay. *(ii) Balance-Aware Asynchronous Aggregation.* The server computes staleness-aware global imbalance estimates and adjusts encoder-specific aggregation rates accordingly, while applying standard staleness decay to the shared classifier.

Our main contributions are summarized as follows.

- We study modality laziness in MMFL under *asynchronous modality availability*, a practical setting for edge and IoT deployments.
- We propose a novel two-tier modality balancing strategy tailored to asynchronous modalities. Specifically, we introduce *opportunistic feature anchors*

for local balancing without idle waiting, and design *balance-aware asynchronous aggregation* that estimates and calibrates modality imbalance with asynchronously arriving model updates.
- Extensive evaluations across four multi-modal benchmarks show that MBA improves both accuracy and efficiency, demonstrating that leveraging asynchrony enables robust and balanced multi-modal federated learning in realistic heterogeneous environments.

## 2 Related Work

### 2.1 Multi-Modal Federated Learning

Multi-modal Federated Learning (MMFL) explores federated learning with clients that hold data from multiple modalities (*e.g.*, text, image, audio, sensor streams). Compared to the uni-modal counterpart, MMFL requires modality fusion, which falls into *early* and *late* fusion [12, 13]. Finer-grained taxonomies further split early fusion into *input-level* and *feature-level* fusion [8]. We focus on *feature-level* fusion in this work.

### 2.2 Modality Incompleteness

A practical challenge in MMFL is *modality incompleteness*, where clients hold a subset of modalities. Harmony [21] addresses this by decoupling modality-specific training from multimodal learning, *i.e.*, uni-modal encoders are trained locally and later coordinated on multi-modal clients. Similarly, ModalityMirror [10] aggregates decoupled models from uni- and multi-modal clients and applies knowledge distillation to boost performance.

We study an orthogonal yet equally practical scenario where clients *possess all modalities* but their data arrive *asynchronously* over time, *i.e.*, asynchronous modality availability. Such asynchrony is common in deployments due to heterogeneous sampling rates, buffering, and network fluctuations [28]. Methods for incomplete modalities [21, 10] are *synchronous* and do not handle asynchronously arriving modalities.

### 2.3 Modality Laziness

Modality laziness [6, 13] arises when one modality overwhelms optimization, leaving weaker modalities under-trained and degrading overall performance. It has been studied mainly in centralized training and more recently in MMFL.

In centralized training, OGM-GE [22] balances optimization by modulating gradient magnitudes. Gradient Blending [23] uses a held-out set to estimate imbalance and re-weight losses. Wu *et al.* [27] adopt early stopping of strong modalities once an imbalance threshold is met. PMR [8] computes imbalance via feature-level prototypes to generalize across fusion stages. In federated learning, BMSFed [7] adopts the PMR-style measure and improves prototype quality by selectively sampling uni-modal clients each round.

Despite their effectiveness, these approaches assume *synchronous* training and *simultaneous* modality availability to compute imbalance metrics and apply corrections. In contrast, we target MMFL under *asynchronous modality availability* and leverage asynchrony to estimate and mitigate modality imbalance.

## 3 Problem Statement

### 3.1 Preliminaries

We study modality laziness in multi-modal federated learning, a unique obstacle to effective modality fusion.

**Multi-Modal Federated Learning (MMFL).** Let $\{c_1, \dots, c_N\}$ be the clients and $\{1, \dots, M\}$ the modalities. Client $c_n$ holds a private multi-modal dataset $D_n = (\{\mathbf{x}_{n,m}\}_{m=1}^{M},\ \mathbf{y}_n)$, where $\mathbf{x}_{n,m}$ stacks modality-$m$ inputs at client $n$ and $\mathbf{y}_n$ is the corresponding label vector. Under server coordination, MMFL [19, 4] collaboratively learns a global model $\theta$ by minimizing

$$\min_{\theta}\ F(\theta) = \sum_{n=1}^{N} \frac{|D_n|}{\sum_{k=1}^{N} |D_k|}\, F_n(\theta; D_n), \tag{1}$$

where $F_n$ is typically the average cross-entropy on $D_n$.

**Feature-Level Fusion in MMFL.** We consider *client-side feature-level* modality fusion [21]. The model decomposes as $\theta = (\{\phi_m\}_{m=1}^{M},\ \psi)$, with modality-specific encoders $\phi_m$ and a shared fusion classifier $\psi$. At round $t$, the server broadcasts $\theta_g^{(t)} = (\{\phi_{g,m}^{(t)}\}_{m=1}^{M},\ \psi_g^{(t)})$ to clients to initialize their local copies $\theta_n^{(t)} = (\{\phi_{n,m}^{(t)}\}_{m=1}^{M},\ \psi_n^{(t)})$. Client $c_n$ performs local training by computing modality-specific embeddings

$$h_{n,m} = \phi_{n,m}^{(t)}(\mathbf{x}_{n,m}), \tag{2}$$

and forming fused predictions

$$\psi_n^{(t)}(\mathrm{Concat}(h_{n,1}, \dots, h_{n,M})) \tag{3}$$

to update $\{\phi_{n,m}\}_{m=1}^{M}$ and $\psi_n$. After local training, $\{\phi_{n,m}^{(t+1)}\}_{m=1}^{M}$ and $\psi_n^{(t+1)}$ are uploaded for server aggregation (*e.g.*, weighted averaging).

**Modality Laziness.** Dominant modalities (*e.g.*, higher quality or more frequent availability) can contribute disproportionately to gradient updates, leaving weaker modalities under-optimized. This imbalance degrades representation quality and multi-modal accuracy [6, 13].

### 3.2 Challenges and Objectives

We investigate modality laziness in MMFL under *asynchronous modality availability* and aim to address it without incurring unnecessary latency.

**Asynchronous Modality Availability.** In IoT-centric deployments, different modalities often arrive at different times due to heterogeneous sensing rates, buffering, or network fluctuations (*e.g.*, home gateways aggregating wearables and ambient sensors for fall detection [28]). Formally, at the beginning of round $t$, client $c_n$ can access only a subset of modalities $\mathcal{Q}_n \subseteq \{1, \ldots, M\}$, while the remaining modalities arrive after a finite delay. This asynchrony complicates local training, as clients must decide whether to proceed with the available modalities or wait for the missing ones.

**Limitations of Prior Art.** Existing approaches [23, 22, 8] mitigate modality laziness by reweighting gradients or adjusting losses according to modality-specific contributions. However, they implicitly assume *simultaneous* access to all modalities to compute such metrics. Directly applying them in asynchronous settings is problematic: *(i)* waiting until $|\mathcal{Q}_n| = M$ before training increases wall-clock latency; and *(ii)* training using only $|\mathcal{Q}_n| < M$ modalities biases gradients toward frequently available modalities, further amplifying imbalance.

**Objective.** We aim for an MMFL design that enables *low-latency* training without waiting for all modalities, while ensuring *balanced learning* across modalities.

# 4 Method

## 4.1 Overview

We propose MBA (Modality Balancing via Asynchrony), a lightweight scheme for mitigating modality laziness in MMFL under asynchronous modality availability. Instead of treating asynchronous modalities as obstacles, MBA exploits them as *opportunistic anchors*, *i.e.*, signals that enable *estimating* and *reducing* modality imbalance without waiting for full availability. MBA implements *two-tier modality balancing* that seamlessly integrates with asynchronous federated learning (see Fig. 1).

- **Opportunistic Local Modality Balancing** (Sec. 4.2). Early modalities trigger uni-modal training whose features are cached as *feature anchors.* Once all modalities are present, the client performs multi-modal training with a *feature-level regularizer* to avoid local gradient dominance.
- **Balance-Aware Asynchronous Aggregation** (Sec. 4.3). Clients upload the model updates and the per-modality feature distances derived from the anchors. The server estimates the staleness-weighted global modality imbalance rates, and uses them to modulate *per-modality* encoder aggregation, while updating the shared classifier using standard staleness-aware decay.

## 4.2 Opportunistic Local Modality Balancing

This subsection mitigates modality laziness during local training. We turn early modalities into *feature anchors* and use them as *regularization* once *all* modalities are available. This mitigates imbalance without idle waiting.

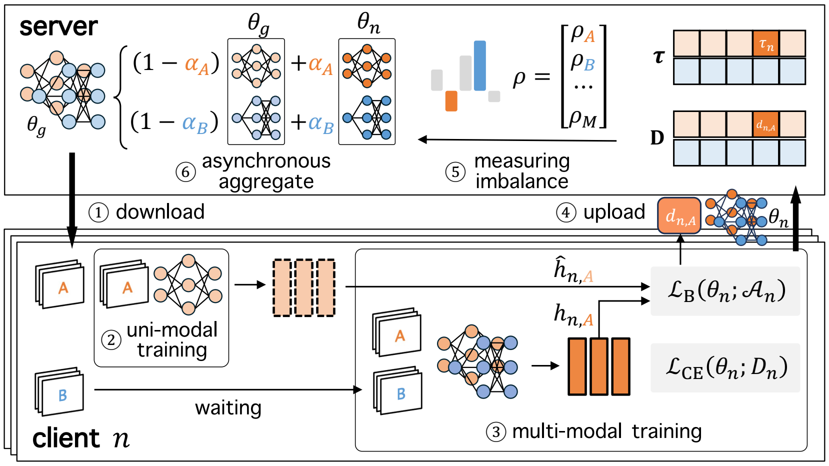

**Fig. 1.** MBA workflow. ① Client $c_n$ downloads the global multi-modal model $\theta_g$. ② Client performs uni-modal training when $|\mathcal{Q}_n| < M$ and caches feature anchors $\hat{h}_{n,m}$. ③ Client conducts multi-modal training with anchor regularization. ④ Client uploads the updated model $\theta_n$ and feature distances $\{d_{n,m}\}$ to server, which updates $\mathbf{D}$ and $\boldsymbol{\tau}$. ⑤ Server measures the imbalance rate $\rho$. ⑥ Server asynchronously aggregates models.

**Uni-Modal Feature Anchors.** At the start of round $t$, the server broadcasts $\theta_g = (\{\phi_{g,m}\}_{m=1}^{M}, \psi_g)$. Client $c_n$ does not wait until all $M$ modalities arrive. For each modality $m \in \mathcal{Q}_n^{(t)}$ that arrives early, it performs *uni-modal* encoder training on $(\mathbf{x}_{n,m}, \mathbf{y}_n)$ using a *local* classifier head that is excluded from aggregation [21]. Let $\hat{\phi}_{n,m}$ be the updated uni-modal encoder, and $\hat{h}_{n,m} = \hat{\phi}_{n,m}(\mathbf{x}_{n,m})$ be the corresponding features. Features from arrived (but not yet fused) modalities are cached as anchors:

$$\hat{h}_{n,\mathcal{A}_n} = \{\hat{h}_{n,m} : m \in \mathcal{A}_n\},\ \mathcal{A}_n \subset \{1, \dots, M\},\ |\mathcal{A}_n| = M - 1. \tag{4}$$

**Multi-Modal Training with Anchor Regularization.** Once all $M$ modalities are available, *i.e.*, $|\mathcal{Q}_n| = M$, client $c_n$ proceeds to multi-modal training. Client $c_n$ initializes $\theta_n = (\{\phi_{n,m}\}_{m=1}^{M}, \psi_n)$ from $\theta_g$ and optimizes

$$F_n(\theta_n) = \mathcal{L}_{\mathrm{CE}}(\theta_n; D_n) \ + \ \beta\, \mathcal{L}_{\mathrm{B}}(\theta_n; \mathcal{A}_n), \tag{5}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the standard task loss, $\mathcal{L}_{\mathrm{B}}$ is the modality-balancing regularization, and $\beta > 0$ trades off the task and balancing terms.

Let $h_{n,m} = \phi_{n,m}(\mathbf{x}_{n,m})$ be the current multi-modal features. The balancing loss aligns the modalities with their uni-modal anchors as:

$$\mathcal{L}_{\mathrm{B}}(\theta_n; \mathcal{A}_n) = \frac{1}{(M-1)} \sum_{m \in \mathcal{A}_n} d_{n,m},\ d_{n,m} = \frac{1}{|\mathbf{x}_{n,m}|} \sum_{\mathbf{x}_{n,m}} [1 - \cos(h_{n,m}, \hat{h}_{n,m})] \tag{6}$$

for the first $M-1$ arrived modalities.

**Discussions.** We make the following notes on the local modality balancing.

- Feature anchors are computed while awaiting remaining modalities, so the procedure does not increase wall-clock time.
- Anchors come from uni-modal updates and are unaffected by cross-modal gradient dominance. Hence, they serve as meaningful per-modality references for modality balancing.
- If a weaker modality arrives early, its anchors steer fusion toward underrepresented signals and mitigate laziness. If a stronger modality arrives first, the regularizer remains small (current and anchored features are already aligned), so modality imbalance does not worsen.

### 4.3 Balance-Aware Asynchronous Aggregation

Building on locally balanced multi-modal training (Sec. 4.2), this subsection enforces *global* modality balance at the server. Unlike synchronous aggregation used in prior work [23, 22, 8], we adopt *asynchronous* aggregation for efficiency. Asynchrony introduces *staleness*, which must be handled both when *(i)* estimating global modality imbalance and *(ii)* aggregating model updates in a modality-balanced manner.

**Measuring Imbalance under Staleness.** Recall that the client-side distance $d_{n,m}$ from Eq. (6) quantifies, for client $c_n$ and modality $m$, how far the current multi-modal feature $h_{n,m}$ deviates from its uni-modal anchor $\hat{h}_{n,m}$, *i.e.*, a per-modality indicator of under-optimization. After local training, each client uploads its parameters $\theta_n$ together with $\{d_{n,m}\}$. The server stores $\mathbf{D} = [d_{n,m}] \in \mathbb{R}^{N\times M}$ and $\boldsymbol{\tau} = [\tau_{n,m}] \in \mathbb{R}^{N\times M}$, where $\tau_{n,m}$ is the number of server updates since client $c_n$ last reported modality $m$ (its staleness).

To align heterogeneous arrival times, the server forms a staleness-weighted imbalance estimate for each modality:

$$\tilde{d}_m = \sum_{n=1}^{N} w_{n,m}\, d_{n,m},\; w_{n,m} = \frac{\exp(-\lambda\tau_{n,m})}{\sum_{k=1}^{N}\exp(-\lambda\tau_{k,m})}, \tag{7}$$

with $\lambda > 0$ controlling the staleness penalty (exponential decay is standard in asynchronous FL). Let $\mu = \frac{1}{M}\sum_{m=1}^{M}\tilde{d}_m$ be the cross-modality average. We then define the global modality imbalance rate

$$\rho_m = \frac{\mu - \tilde{d}_m}{\tilde{d}_m + \epsilon}, \tag{8}$$

with small $\epsilon > 0$ for numerical stability. Thus, $\rho_m > 0$ indicates that modality $m$ is better aligned (smaller discrepancy) than average, whereas $\rho_m < 0$ indicates under-optimization. Normalizing by $\tilde{d}_m$ sharpens sensitivity when distances are small. These rates $\{\rho_m\}$ guide modality-balanced aggregation below.

**Algorithm 1: MBA:** Modality Balancing via Asynchrony

```
Input: Global model $\theta = (\{\phi_m\}_{m=1}^M, \psi)$; local epochs $E$; asynchronous decay
       $\alpha_0$, $a$; regularization $\beta$; staleness penalty $\lambda$; imbalance sensitivity $\gamma$
Server:
Initialize $\theta_g = (\{\phi_{g,m}\}_{m=1}^M, \psi_g)$; initialize tables $\mathbf{D}, \boldsymbol{\tau}$;
for each upload from client $n$ at time $t$ do
    Receive $(\theta_n, \{d_{n,m}\}, \text{timestamps})$; compute staleness $\tau$;
    Update $\mathbf{D}, \boldsymbol{\tau}$ for client $c_n$;
    // Staleness-aware imbalance estimation
    Compute $\tilde{d}_m$ by Eq. (7); compute $\rho_m$ by Eq. (8);
    // Modality-specific asynchronous aggregation
    $s(\tau) \leftarrow \alpha_0(\tau+1)^{-a}$; $\alpha_m \leftarrow \text{clip}\big(s(\tau)(1-\gamma\rho_m), 0, 1\big)$;
    Update $\phi_{g,m}$ by Eq. (11) for $m = 1, \ldots, M$; update $\psi_g$ by Eq. (12);
Client $c_n$:
Download $\theta_g$; set $\theta_n \leftarrow \theta_g$;
while $|\mathcal{Q}_n| < M$ do
    for each newly arrived $m \in \mathcal{Q}_n$ do
        // Opportunistic uni-modal anchoring
        Train $\phi_{n,m}$ with a local head; cache $\hat{h}_{n,m}$ by Eq. (4);
for $e = 1$ to $E$ do
    // Multi-modal training with anchor regularization
    Update $\theta_n$ on $F_n$ by Eq. (5); compute $d_{n,m}$ by Eq. (6);
Upload $(\theta_n, \{d_{n,m}\})$ to the server;
```

**Balancing Modalities during Asynchronous Aggregation.** When an update trained from $\theta_g^{(t-\tau)}$ arrives at server time $t$ with staleness $\tau \geq 0$, the server applies the base decay [29]:

$$\theta_g^{(t+1)} = (1 - s(\tau))\, \theta_g^{(t)} + s(\tau)\, \theta_n^{(t)},\; s(\tau) = \alpha_0\, (\tau + 1)^{-a}, \tag{9}$$

where $\alpha_0 > 0$ and $a > 0$ control magnitude and decay rate. While (9) mitigates staleness, it treats all modalities equally and can induce laziness.

With feature-level fusion $\theta = (\{\phi_m\}_{m=1}^M, \psi)$, we assign a *separate* aggregation rate per modality, modulated by its imbalance:

$$\alpha_m = \text{clip}(s(\tau)\, (1 - \gamma\, \rho_m),\, 0,\, 1), \tag{10}$$

where $\gamma > 0$ controls sensitivity and clipping ensures stability. Intuitively, under-optimized modalities ($\rho_m < 0$) are up-weighted relative to $s(\tau)$, while over-optimized ones ($\rho_m > 0$) are down-weighted to prevent dominance.

The server then updates the encoders modality-wise and the shared classifier with the base decay as follows:

$$\phi_{g,m}^{(t+1)} = (1 - \alpha_m)\, \phi_{g,m}^{(t)} + \alpha_m\, \phi_{n,m}^{(t)} \quad (m = 1, \ldots, M), \tag{11}$$

$$\psi_g^{(t+1)} = (1 - s(\tau))\, \psi_g^{(t)} + s(\tau)\, \psi_n^{(t)}. \tag{12}$$

**Summary.** The server *(i)* estimates global modality imbalance robustly under staleness via $\tilde{d}_m$ and $\rho_m$, and *(ii)* mitigates laziness by allocating larger effective weights to under-optimized modalities during aggregation. This balance-aware asynchronous aggregation reduces wall-clock latency while promoting balanced federated training across modalities.

### 4.4 Putting It Together

Algorithm 1 illustrates the overall workflow of MBA. The server initializes $\theta_g = (\{\phi_{g,m}\}_{m=1}^{M}, \psi_g)$ and the per-modality staleness matrix (line 2). A selected client $c_n$ downloads $\theta_g$ and, while $|\mathcal{Q}_n| < M$, runs uni-modal training for each arrived $m \in \mathcal{Q}_n$, producing anchors $\hat{h}_{n,m}$ (Eq. (4)) without increasing wall-clock time (lines 11-13). Once $|\mathcal{Q}_n| = M$, the client performs multi-modal training on $F_n$ with the anchor regularizer (Eq. (5)-Eq. (6)) and records per-modality distances $d_{n,m}$ (lines 14-16). The client uploads $\theta_n$ and $\{d_{n,m}\}$ (and timestamps) (line 16). Upon receipt, the server updates staleness counters, computes $\tilde{d}_m$ via exponential decay weights (Eq. (7)), and derives $\rho_m$ (Eq. (8)) (lines 4-6). The server computes $s(\tau)$ (base decay) and per-modality rates $\alpha_m = \mathrm{clip}\big(s(\tau)(1 - \gamma\rho_m), 0, 1\big)$, then updates the encoders modality-wise (Eq. (11)) and the shared classifier with $s(\tau)$ (Eq. (12)) (lines 6-7). New clients pull the latest $\theta_g$ and the process continues asynchronously.

## 5 Experiments

### 5.1 Experiment Setup

**Environment.** All experiments run on a workstation with two Intel Xeon Gold 6230R CPUs, 256 GiB RAM, and four NVIDIA V100 GPUs with 32 GiB memory.

**Datasets.** We evaluate on four multi-modal datasets: *(i)* UPMC Food-101 [24] with image and text, *(ii)* USC-HAD [30] with accelerometer and gyroscope signal, *(iii)* CREMA-D [3] with video and audio, and *(iv)* IEMOCAP [2] with video, audio, and text.

**Baselines.** We include six baselines methods in total: a naive concatenation-based fusion method, three for modality laziness (G-Blend [23], OGM-GE [22] and PMR [8]) and two for modality incompleteness (Harmony [21] and ModalityMirror [10]). Some of baselines are initially designed for centralized or synchronous FL scenario. To ensure fairness, we extend them to asynchronous FL versions for evaluation (see details in Appendix A.2).

**Models.** For the visual modality, we use ResNet-18 [11] for CREMA-D, IEMOCAP, and ViT [1] for Food-101. For the audio modality, we apply ResNet-18 [11], and adapt the input channel from three to one to fit the shape of audio features. For the textual modality, we use a standard BERT model with its tokenizer [5].

**Table 1.** Overall accuracy (%) results on double-modal datasets. "Acc" is the final multi-modal accuracy. "Acc-[Modality]" is the accuracy under the respective uni-modal input only.

| Method | Food-101 | | | USC-HAD | | | CREMA-D | | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc | Acc-V | Acc-T | Acc | Acc-A | Acc-G | Acc | Acc-V | Acc-A |
| Fed-Concat-Async | 80.91 | 56.78 | 57.53 | 93.78 | 49.83 | 59.73 | 44.86 | 15.42 | 44.14 |
| Fed-PMR-Async | 77.06 | 51.31 | 56.36 | 91.49 | 45.74 | 54.17 | 64.32 | 32.55 | 43.93 |
| Fed-G-Blend-Async | 81.02 | 53.75 | 64.44 | 93.58 | 90.76 | 84.13 | 48.34 | 27.03 | 44.24 |
| Fed-OGM-GE-Async | 79.86 | 55.08 | 59.13 | 92.51 | 48.77 | 60.36 | 45.23 | 28.07 | 36.77 |
| Harmony-Async | 71.98 | 16.08 | 66.80 | 95.07 | 90.20 | 68.97 | 62.86 | 34.40 | 49.14 |
| ModalityMirror-Async | 58.78 | 58.73 | 1.71 | 21.26 | 67.63 | 7.71 | 25.08 | 14.56 | 35.95 |
| MBA | **82.42** | 53.06 | 65.27 | **95.42** | 44.36 | 57.07 | **67.69** | 33.19 | 50.11 |

**Streaming Data Simulation.** To emulate the *sensing* of streaming data, we randomly shuffle the arrival order of modalities for each client in every training round. Only half of the modalities are initially available, and the remaining ones arrive after a 300-second delay [25]. On simulating the *transimission* of streaming data, each round samples a random bandwidth uniformly from $[200, 800]$ KBps due to unstable networks, consistent with empirical Bluetooth measurements on mobile devices [15]. Each modality incurs a transmission time determined by the sampled bandwidth and the size of its data on the client.

**Edge Device Simulation.** We simulate *training latency* of edge device with NVIDIA V100 GPUs following [14], where the training latency of Jetson TX2 is about 36× that of V100. We therefore scale the V100 training latency by the same factor to estimate edge device runtimes. To further emulate the *heterogeneity* of edge devices (*e.g.*, Jetson TX2 and Jetson Nano), we apply an additional scaling of slow devices requiring 5× training latency of fast devices [16].

**Configurations.** Following [29], the base decay coefficient $\alpha_0$ is set to 0.9, and the decay function adopts a *polynomial* form with exponent $a = 0.5$. We employ the SGD optimizer with an initial learning rate of 0.01, which decay by 0.99 each round. The training round is set to 250 for synchronous FL, and 2000 for asynchronous FL. The client sampling rate is fixed at 0.25. The hyperparameter $\beta$, $\gamma$ is tuned within $\{0.1, 0.2\}$ depending on the dataset.

### 5.2 Results

**Accuracy.** As shown in Table 1 and Table 2, MBA achieves the highest overall multi-modal accuracy across all datasets under asynchronous modality availability, which indicates effective mitigation of modality laziness. On Food-101, MBA delivers competitive overall performance while preserving balanced uni-modal accuracies with transformer-based backbones. On USC-HAD, it attains 95.42% and slightly exceeds the best baseline. On CREMA-D, MBA reaches 67.69%

**Table 2.** Accuracy (%) results on triple-modal dataset.

| Method | IEMOCAP | | | |
|---|---|---|---|---|
| | Acc | Acc-V | Acc-A | Acc-T |
| Fed-Concat-Async | 61.35 | 34.66 | 29.37 | 35.01 |
| Fed-G-Blend-Async | 39.20 | 36.03 | 35.94 | 33.93 |
| Harmony-Async | 63.56 | 55.50 | 34.98 | 46.18 |
| MBA | **64.11** | 49.60 | 26.57 | 34.11 |

and surpasses PMR by 3.37%. On the triple-modal IEMOCAP dataset, MBA attains 64.11%, which is higher than PMR at 61.35% and well above G-Blend at 39.20%, demonstrating stronger generalization in complex multi-modal scenarios. Compared with methods designed for modality incompleteness, MBA shows superior robustness and accuracy under asynchronous arrivals. This advantage arises because MBA additionally addresses modality laziness, so that weaker modalities continue to receive optimization anchors during training, even when Harmony achieves slightly higher uni-modal accuracy. This further indicates that uni-modal accuracy alone may not provide a sufficiently informative proxy for measuring the imbalance rate of modality laziness.

**Efficiency.** We evaluate time-to-accuracy under both synchronous and asynchronous FL settings. For comparability, all wall-clock times are normalized to

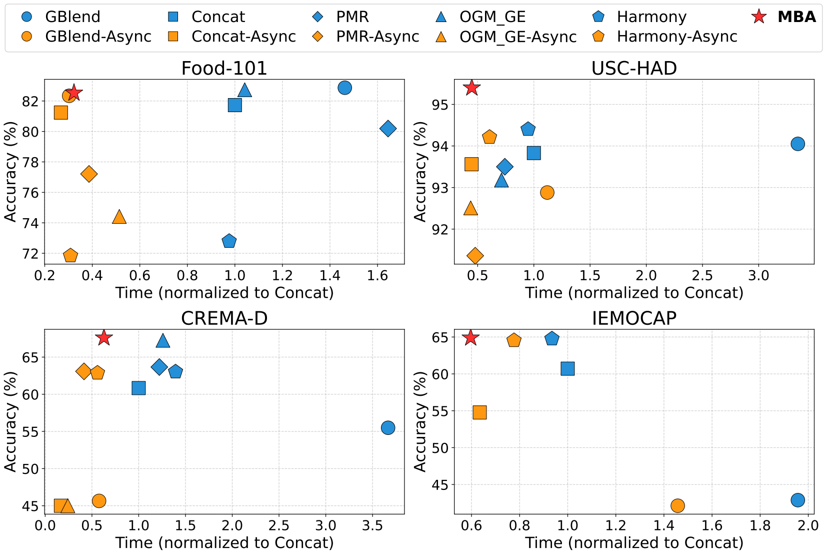

**Fig. 2.** Accuracy vs. normalized time cost.

**Table 3.** Component Ablations

| Method | Dataset | | | |
|---|---|---|---|---|
| | Food-101 | USC-HAD | CREMA-D | IEMOCAP |
| MBA | 81.97 | 95.42 | 67.69 | 64.11 |
| MBA (w/o regularization) | 80.83 | 93.18 | 61.35 | 63.83 |
| MBA (w/o balance-aware aggregation) | 79.50 | 93.12 | 62.87 | 63.21 |

the Concat baseline (set to 1.0×). As shown in Fig. 2, MBA requires substantially less time than synchronous baselines to reach strong accuracy. Among asynchronous approaches, convergence speeds are comparable, while MBA attains the highest final accuracy. This improvement is achieved by uni-modal feature anchors that exploit idle waiting time to guide optimization. ModalityMirror is omitted from the efficiency plots due to non-competitive accuracy in our setting.

### 5.3 Ablation Study

**Component Ablations.** To evaluate the effectiveness of each component in MBA, we conduct ablations that remove one component at a time. We evaluate two variants. *(i)* MBA (w/o regularization) disables the anchor-based feature regularization and trains with cross-entropy loss only. *(ii)* MBA (w/o balance-aware aggregation) replaces the proposed balance-aware asynchronous aggregation with the vanilla asynchronous aggregation in [29]. As shown in Table 3, the full MBA consistently outperforms both variants. Removing the anchor regularization yields a clear accuracy decrease, which indicates that feature-level alignment is key to constraining the optimization direction and mitigating modality laziness. Replacing the balance-aware aggregation also degrades accuracy, which confirms the effectiveness of balance-aware aggregation to mitigate modality laziness.

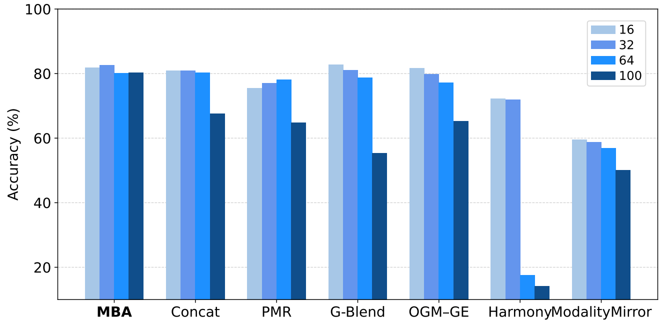

**Fig. 3.** Scalability experiments on Food-101 under 16, 32, 64, and 100 clients.

**Scalability.** Food-101 provides a sufficiently large sample size to examine scalability. We evaluate MBA and all asynchronous baselines under client partitions of 16, 32, 64, and 100, and we report the results in Fig. 3. As the number of clients increases, baseline performance declines markedly, whereas MBA degrades only marginally. Under the 100-client setting, MBA reaches 80.38% and outperforms the second-best method (Concat) by 12.71 percentage points. These findings indicate strong stability and scalability of MBA under increasingly fragmented and asynchronous participation.

**Hyperparameter Sensitivity.** MBA maintains stable accuracy across a broad range of settings for $\beta$ and $\gamma$ with only minor fluctuations. Performance on Food-101 and USC-HAD remains highly consistent, which suggests robustness to the regularization strength and the aggregation balance parameter. This stability shows that MBA does not rely on precise hyperparameter tuning and sustains strong performance across diverse datasets and training conditions.

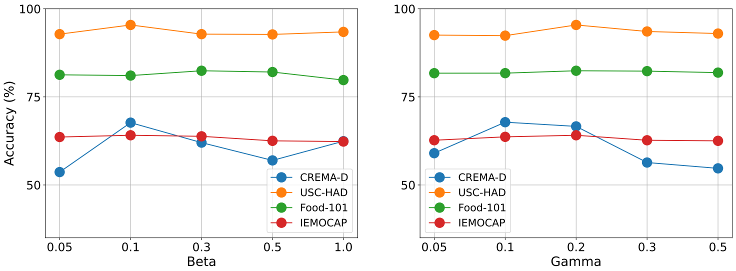

**Fig. 4.** Accuracy of MBA under different hyperparameters.

## 6 Conclusion

This paper explores how asynchrony, often viewed as a source of inefficiency, can be harnessed to balance modalities in multi-modal federated learning. We design MBA, a two-tier framework that effectively and efficiently mitigates modality laziness under asynchronous modality availability. On the client side, opportunistic local balancing uses early-arriving modalities to form feature anchors that regularize multi-modal training without idle waiting. On the server side, balance-aware asynchronous aggregation estimates the global cross-modality imbalance using staleness-weighted statistics and adaptively adjusts aggregation rates accordingly. Extensive experiments demonstrate that MBA consistently improves both accuracy and training efficiency across diverse benchmarks. Our findings suggest that leveraging asynchrony offers a promising new direction for robust and balanced multi-modal federated learning in realistic, heterogeneous environments.

**Acknowledgments.** We are grateful to anonymous reviewers for their constructive comments. The research of Yiming Ma, Boyi Liu and Yongxin Tong is supported by National Science Foundation of China (NSFC) (Grant Nos. 62425202, 62336003), the Beijing Natural Science Foundation (Z230001), the Fundamental Research Funds for the Central Universities No. JK2024-03, the Didi Collaborative Research Program, the State Key Laboratory of Complex & Critical Software Environment (SKLCCSE) . Zimu Zhou's research is supported by National Science Foundation of China (NSFC) (No. 62572412), City University of Hong Kong Shenzhen Research Institute (CityUSRI), and CityU APRC grant (No. 9610633). The corresponding authors are Yanfeng Wang and Yongxin Tong.

## References

1. Alexey, D., Lucas, B., Alexander, K., Dirk, W., Xiaohua, Z., Thomas, U., Mostafa, D., Matthias, M., Georg, H., Sylvain, G., Jakob, U., Neil, H.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR. pp. 611–631 (2021)
2. Busso, C., Bulut, M., Lee, C.C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J.N., Lee, S., Narayanan, S.S.: Iemocap: Interactive emotional dyadic motion capture database. Language resources and evaluation **42**, 335–359 (2008)
3. Cao, H., Cooper, D.G., Keutmann, M.K., Gur, R.C., Nenkova, A., Verma, R.: Crema-d: Crowd-sourced emotional multimodal actors dataset. IEEE transactions on affective computing **5**(4), 377–390 (2014)
4. Che, L., Wang, J., Zhou, Y., Ma, F.: Multimodal federated learning: A survey. Sensors **23**(15), 6986 (2023)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT. pp. 4171–4186 (2019)
6. Du, C., Teng, J., Li, T., Liu, Y., Yuan, T., Wang, Y., Yuan, Y., Zhao, H.: On unimodal feature learning in supervised multi-modal learning. In: ICML. pp. 8632–8656. PMLR (2023)
7. Fan, Y., Xu, W., Wang, H., Huo, F., Chen, J., Guo, S.: Overcome modal bias in multi-modal federated learning via balanced modality selection. In: ECCV. pp. 178–195. Springer (2024)
8. Fan, Y., Xu, W., Wang, H., Wang, J., Guo, S.: Pmr: Prototypical modal rebalance for multimodal learning. In: CVPR. pp. 20029–20038 (2023)
9. Feng, T., Bose, D., Zhang, T., Hebbar, R., Ramakrishna, A., Gupta, R., Zhang, M., Avestimehr, S., Narayanan, S.: Fedmultimodal: A benchmark for multimodal federated learning. In: KDD. pp. 4035–4045 (2023)
10. Feng, T., Zhang, T., Avestimehr, S., Narayanan, S.: Modalitymirror: Enhancing audio classification in modality heterogeneity federated learning via multimodal distillation. In: NOSSDAV. pp. 78–83 (2025)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
12. Huang, W., Wang, D., Ouyang, X., Wan, J., Liu, J., Li, T.: Multimodal federated learning: Concept, methods, applications and future directions. Information Fusion **112**, 102576 (2024)
13. Huang, Y., Lin, J., Zhou, C., Yang, H., Huang, L.: Modality competition: What makes joint training of multi-modal network fail in deep learning?(provably). In: ICML. pp. 9226–9259. PMLR (2022)

14. Kang, P., Jo, J.: Benchmarking modern edge devices for ai applications. IEICE TRANSACTIONS on Information and Systems **104**(3), 394–403 (2021)
15. Li, T., Huang, J., Risinger, E., Ganesan, D.: Low-latency speculative inference on distributed multi-modal data streams. In: MobiSys. pp. 67–80 (2021)
16. Liu, B., Ma, Y., Zhou, Z., Shi, Y., Li, S., Tong, Y.: Casa: Clustered federated learning with asynchronous clients. In: KDD. pp. 1851–1862 (2024)
17. Liu, Y., Wang, C., Yuan, X.: Fedmobile: Enabling knowledge contribution-aware multi-modal federated learning with incomplete modalities. In: Proceedings of the ACM on Web Conference 2025. pp. 2775–2786 (2025)
18. McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and music signal analysis in python. SciPy **2015**, 18–24 (2015)
19. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS. pp. 1273–1282. PMLR (2017)
20. Ouyang, X., Shuai, X., Li, Y., Pan, L., Zhang, X., Fu, H., Cheng, S., Wang, X., Cao, S., Xin, J., et al.: Admarker: A multi-modal federated learning system for monitoring digital biomarkers of alzheimer's disease. In: MobiCom. pp. 404–419 (2024)
21. Ouyang, X., Xie, Z., Fu, H., Cheng, S., Pan, L., Ling, N., Xing, G., Zhou, J., Huang, J.: Harmony: Heterogeneous multi-modal federated learning through disentangled model training. In: MobiSys. pp. 530–543 (2023)
22. Peng, X., Wei, Y., Deng, A., Wang, D., Hu, D.: Balanced multimodal learning via on-the-fly gradient modulation. In: CVPR. pp. 8238–8247 (2022)
23. Wang, W., Tran, D., Feiszli, M.: What makes training multi-modal classification networks hard? In: CVPR. pp. 12695–12705 (2020)
24. Wang, X., Kumar, D., Thome, N., Cord, M., Precioso, F.: Recipe recognition with large multimodal food dataset. In: ICMEW. pp. 1–6. IEEE (2015)
25. Wu, F., Qiu, C., Wu, T., Yuce, M.R.: Edge-based hybrid system implementation for long-range safety and healthcare iot applications. IEEE Internet of Things Journal **8**(12), 9970–9980 (2021)
26. Wu, F., Liu, S., Zhu, K., Li, X., Guo, B., Yu, Z., Wen, H., Xu, X., Wang, L., Liu, X.: Adaflow: Opportunistic inference on asynchronous mobile data with generalized affinity control. In: Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems. pp. 606–618 (2024)
27. Wu, N., Jastrzebski, S., Cho, K., Geras, K.J.: Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks. In: ICML. pp. 24043–24055. PMLR (2022)
28. Wu, Q., Chen, X., Zhou, Z., Zhang, J.: Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. IEEE Transactions on Mobile Computing **21**(8), 2818–2832 (2020)
29. Xie, C., Koyejo, O., Gupta, I.: Asynchronous federated optimization. In: OPT (2020)
30. Zhang, M., Sawchuk, A.A.: Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In: UbiComp. pp. 1036–1043 (2012)
31. Zhao, Y., Barnaghi, P., Haddadi, H.: Multimodal federated learning on iot data. In: 2022 IEEE/ACM seventh international conference on internet-of-things design and implementation (ioTDI). pp. 43–54. IEEE (2022)
32. Zheng, T., Li, A., Chen, Z., Wang, H., Luo, J.: Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving. In: MobiCom. pp. 1–15 (2023)

# A Appendix

## A.1 Data Preprocessing

For the visual modality, we uniformly sample 3 frames from each video. Each frame is horizontally cropped to remove uninformative side regions and resized to 224 × 224. The processed frames are then concatenated vertically to form a single composite image. For the audio modality, we load the waveform at 16 kHz and tile it to a fixed length of 3 seconds. The signal is clipped to the range $[-1, 1]$ for stability. We compute the spectrogram using Librosa [18] with a window size of 512 and an overlap of 353. The magnitude is transformed to the log scale and standardized per sample. The resulting feature has shape $[1, 257, 300]$ and serves as the model input. For the text modality, we tokenize the raw text using the BERT-base tokenizer [5] with a maximum input length of 256.

## A.2 Modifications for Asynchrony

We adapt all baselines to support asynchronous federated training. The specific modifications for each method are described below. Gradient Blending requires estimating the overfitting and generalization rates. We compute these metrics before and after local training. The server then performs standard asynchronous model aggregation without modification. OGM-GE is directly applied during local client training as described in the original method. The server employs naive asynchronous aggregation. PMR maintains modality-specific prototypes. We adapt this mechanism to run on the server, where prototypes are updated asynchronously via decayed aggregation, in line with the momentum-based update proposed in the original paper. Harmony involves two-stage training, with the second stage using clustering to handle non-IID distribution. Since non-IID distribution is not applied in our setting, we skip clustering and perform federated fusion training in stage 2. On the server side, we follow a schedule that 1000 rounds for stage 1 and 1000 rounds for stage 2. ModalityMirror aggregates multi-modal models with uni-modal submodels in Phase 1, which can be deployed client-side. Under the random arrival pattern, the client performs uni-modal training if the arriving modality belongs to its preassigned target set. Otherwise, it waits until all target modalities have arrived before proceeding with multi-modal training. Phase 2 of ModalityMirror, which aims to improve uni-modal client performance through knowledge distillation from multi-modal clients, is not applicable in our setting, where evaluation is based on the full set of modalities. Therefore, Phase 2 is omitted in our modification.