AdaModalX: Synchronization-Free MFM Inference for Distributed Multimodal Mobile Sensing via Progressive Data Orchestration

Xiangrui Xu, Sicong Liu, *Member, IEEE*, Zhiwen Yu, *Senior Member, IEEE*, Zimu Zhou, *Member, IEEE*, and Bin Guo, *Senior Member, IEEE*

***Abstract*—Multimodal foundation models (MFMs) support generalizable inference in distributed mobile sensing applications by integrating heterogeneous data streams. However, existing MFM inference pipelines typically require complete and synchronized multimodal inputs, whereas mobile sensing streams arrive asynchronously due to network variability and heterogeneous data volumes across modalities. Waiting for all modalities increases end-to-end latency, while inference with incomplete inputs can reduce accuracy. We argue that multimodal inference should depend on the utility of available information rather than modality completeness. Based on this principle, we propose AdaModalX, a progressive multimodal inference framework that decouples inference execution from modality synchronization through runtime utility-aware orchestration. At each device, importance-aware data uploading uses a lightweight probe distilled from the edge-side MFM to allow reliable samples to exit locally while uncertain samples continue uploading. At the edge, on-demand MFM activation constructs candidates from fully received samples and, when needed, imputed features derived from partially received delayed inputs. A lightweight contribution assessor built on frozen MFM encoders estimates candidate utility and activates the MFM once the candidate meets its corresponding utility threshold. Experiments on four mobile sensing applications show that AdaModalX improves the accuracy-latency trade-off, achieving up to 26.97% accuracy improvement and up to 64.75% end-to-end latency reduction.**

***Index Terms*—Multimodal foundation models, distributed mobile sensing, progressive multimodal inference.**

I. Introduction

MOBILE applications increasingly rely on distributed multimodal sensing to support intelligent services and improve user experience [1], [2], [3], [4]. These systems employ a network of distributed mobile devices, each equipped with either different or the same sensors, working collaboratively to gather multimodal data streams. The collected multimodal data are then transmitted to edge or cloud servers, where advanced models perform robust, context-rich inference (see Fig. 1). For example, in *smart homes*, wearable devices track physiological signals such as heart rate and skin temperature, while ambient sensors monitor environmental factors like humidity and air quality. These data streams are processed at the home gateway to detect health anomalies, predict stress levels, or provide personalized wellness insights [5]. Similarly, in *autonomous vehicles*, cameras offer visual scene interpretation, radar identifies objects under varying weather conditions, and LiDAR provides precise spatial mapping. These heterogeneous modalities are integrated in *real time* to enable safe navigation and informed decision-making in complex driving environments [6].

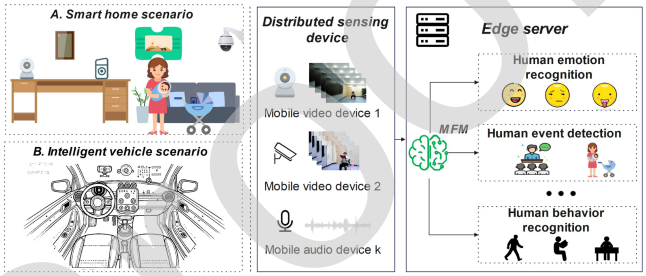


Fig. 1. Example distributed mobile sensing applications in smart homes and intelligent vehicles.

These applications are enabled not only by the widespread deployment of mobile and embedded sensors and wireless networks but also by advances in multimodal models for edge-side inference. In particular, multimodal deep models [7] exploit complementary information across heterogeneous sensor streams to support accurate perception and inference. However, their performance often degrades sharply when deployment involves *zero-shot* or *few-shot* classes, which commonly occurs in mobile environments with evolving users, activities, and contexts. To address these limitations, multimodal foundation models (MFMs) [8], [9], [10] have gained prominence for their ability to generalize to zero- or few-shot classes with little or no fine-tuning [11], [12]. This capability supports emerging applications across diverse scenarios, including in-vehicle

Received 15 May 2026; revised 29 August 2026; accepted 30 September 2026. This work was supported in part by the National Key R&D Program of China under Grant 2024YFB4505502 and in part by the National Natural Science Foundation of China under Grant 62522215, Grant 62532009, and Grant 62472354. Recommended for acceptance by A. Bhattacharya. *(Corresponding authors: Sicong Liu; Zhiwen Yu.)*

Xiangrui Xu, Sicong Liu, and Bin Guo are with Northwestern Polytechnical University, Xi'an 710072, China (e-mail: xiangruixu@mail.nwpu.edu.cn; scliu@nwpu.edu.cn; guob@nwpu.edu.cn).

Zhiwen Yu is with Northwestern Polytechnical University, Xi'an 710072, China, and also with Harbin Engineering University, Harbin 150001, China (e-mail: zhiwenyu@nwpu.edu.cn).

Zimu Zhou is with the City University of Hong Kong, Hong Kong, SAR 999077, China (e-mail: zimuzhou@cityu.edu.hk).

Digital Object Identifier 10.1109/TMC.2026.3742480

1536-1233 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

IEEE Proof

behavior monitoring (*e.g.*, Intel [13], Tesla [14]), human activity recognition (*e.g.*, Microsoft [15]), and smart-home agents (*e.g.*, Amazon [16], Google Astra [17]).

Although MFMs improve the reliability of multimodal inference, their performance is often challenged by *modality asynchrony*, which commonly arises in distributed mobile sensing systems. In real-world mobile sensing deployments, heterogeneous sensors operate at different sampling rates, generate vastly different data volumes, and experience dynamically varying wireless transmission conditions. The resulting modality streams rarely arrive at the edge server simultaneously. Existing MFMs are largely built upon a *synchronization-dependent execution paradigm*, where inference can proceed only after all modalities become available. Although recent approaches support flexible modality combinations [18], [19], their inference pipelines still rely on *synchronized modality completeness*, making responsiveness fundamentally constrained by the slowest modality stream (see Section II-B). As a result, waiting for delayed modalities increases inference latency [20], [21], while performing inference on incomplete modality states often degrades prediction accuracy due to insufficient cross-modal information [22], [23]. Existing approaches that reconstruct delayed modalities through raw-data synthesis [20], [21] further suffer from error propagation across downstream multimodal fusion pipelines.

We argue that multimodal inference should not be treated as a synchronization barrier problem. Instead, it should be viewed as a progressive inference process that evolves continuously with modality arrivals. Our key insight is that accurate multimodal inference does not always require globally synchronized modality completeness. Different modalities contribute unequally to inference outcomes, and delayed modalities often provide only marginal incremental utility beyond the information already accumulated from currently available streams (see Section II-B). This observation suggests a fundamentally different execution paradigm for distributed multimodal systems, *i.e.*, inference should evolve progressively with information accumulation, rather than wait for fully synchronized modality arrivals before execution.

Guided by this principle, we introduce AdaModalX, a progressive multimodal inference runtime for distributed mobile sensing systems. Rather than modifying MFM architectures or retraining large multimodal models into dynamic networks, which demand extensive data and computational resources to preserve their accuracy [24], [25], [26], AdaModalX introduces an *orthogonal runtime orchestration layer* that enables existing MFMs [12], [27] to operate efficiently under asynchronous modality arrivals. Specifically, AdaModalX first constructs a candidate from all fully received samples and evaluates its utility. If the candidate does not meet its corresponding utility threshold, it uses the partially received input from the most informative incomplete modality to generate an imputed feature and augment the candidate. It activates MFM inference once the evaluated candidate meets its corresponding utility threshold, thereby avoiding unnecessary synchronization delays.

To realize this capability, we re-purpose frozen MFM encoders with a shared lightweight assessor to learn modality contribution patterns across diverse inputs offline, without altering the original MFM architectures. During online execution, AdaModalX progressively updates the current modality candidate as new modalities arrive and selectively activates MFM inference according to the estimated candidate utility. To further reduce end-to-end latency, AdaModalX introduces a hierarchical inference architecture spanning mobile devices and the edge server, where lightweight distilled models run locally to process incoming samples and only uncertain samples are offloaded for edge-side MFM inference, thereby significantly reducing wireless transmission overhead and inference latency without sacrificing accuracy.

We implement AdaModalX and test it across four tasks and four real-world mobile sensing applications. Extensive experiments on four representative scenarios with two MFM inference architectures, four datasets, and four types of mobile devices demonstrate that AdaModalX consistently outperforms existing synchronous and asynchronous MFM inference methods, achieving superior performance in both accuracy (*e.g.*, up to 26.97% improvement) and responsiveness (*e.g.*, up to 64.75% latency reduction). Notably, these gains are observed under varying modality availability patterns, heterogeneous device conditions, and asynchronous input arrivals, highlighting the robustness and adaptability of our approach. By dynamically adapting the inference process to the currently available inputs, AdaModalX effectively advances the accuracy–latency trade-off frontier in MFM inference. Our main contributions are summarized as follows.

- We identify a fundamental mismatch between synchronization-dependent MFM execution and asynchronous distributed mobile sensing environments, and introduce progressive multimodal inference as a new execution paradigm that evolves with modality arrivals.
- We design AdaModalX, a progressive multimodal inference runtime that decouples MFM execution from globally synchronized modality completeness. It determines whether to exit early based on device-side confidence and, otherwise, when to activate the edge-side MFM according to candidate utility as modality data progressively arrive.
- Extensive experiments with four tasks across four devices and dynamic scenarios show that AdaModalX outperforms existing synchronous/asynchronous methods [18], [20], [21], [27], [28], [29], [30], [31] in trading off between inference accuracy and low latency across diverse mobile tasks, devices, and scenarios.

II. Motivation

A. MFM-Based Distributed Mobile Sensing

Distributed mobile sensing systems have become a core component of mobile computing applications, such as smart homes and autonomous driving systems. In these systems, distributed devices collaborate to gather multimodal sensory data, which are uploaded to the edge server for accurate and robust inference. For example, AliGenie [32] collects audio and vision from smartphones, cameras, and other devices, while edges fuse and analyze these multimodal streams using multimodal

IEEE Proof

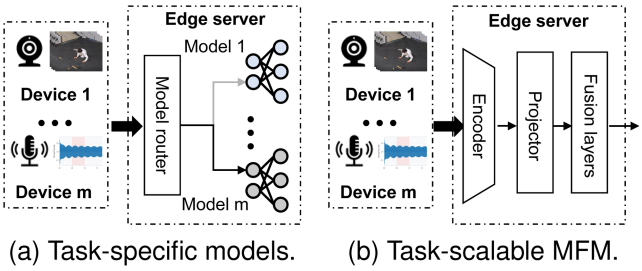


Fig. 2. Comparison of (a) conventional deep models and (b) task-scalable MFM. The former explicitly selects models based on tasks and input modalities, whereas the latter automatically adapts to tasks and input modalities using the same model.

deep models (*e.g.*, AVEL [33]) to support mobile applications including emotion recognition [34], event detection [33], and human activity recognition [35], [36].

To scale with the diversity of mobile users and downstream tasks, multimodal foundation models (MFMs) have gained increasing attention for edge-side inference [11]. Compared to traditional task-specific deep models, MFMs offer the following advantages (see Fig. 2).

- *Generalization to Diverse Modality Combinations:* MFMs can easily handle diverse input modalities without modifying model architectures [12], [37]. For example, in driver behavior recognition, an MFM can integrate RGB, infrared, and depth inputs and adapt to different modality combinations through modality projection [8].
- *Few-shot Generalization to New Environments:* MFMs excel in few-shot and even zero-shot learning [11], *e.g.*, through prompt learning [38]. For instance, our preliminary experiment on event detection shows that a pre-trained MFM based on ImageBind [39] can achieve an accuracy of 70.38%, substantially outperforming task-specific models [40], which achieve only 19.88% under the 3-shot per-class setting (see Fig. 3(b)).

B. MFM Inference With Asynchronous Modalities

Challenges: Standard multimodal models typically assume that data from all modalities are available for accurate inference [20], [21]. However, this assumption is often violated in real-world distributed mobile sensing systems. First, devices often lack synchronization. Second, transmission delays may vary substantially across modalities because different modalities generate different data volumes. For example, under the same bandwidth, vision clips often arrive later than audio because of their larger sizes. Consequently, distributed multimodal data streams rarely reach the edge server simultaneously. Such modality asynchrony creates a fundamental trade-off between waiting for slower modalities, which increases latency [20], [21], and performing early inference with incomplete modalities, which can degrade accuracy [22], [23].

To illustrate this challenge, consider human event detection in a distributed mobile sensing system with audio and vision inputs, where the vision modality arrives later because of its larger size (see Fig. 3(a)). The model uses full audio data and partial vision data for inference. The missing rate denotes the percentage of vision data not yet arrived when the audio data has been fully received at the edge server. If we allow the MFM (ImageBind [12] in our experiment) to perform inference with partial vision data, the inference latency decreases from 0.486 s to 0.149 s, yet the accuracy also drops drastically from 83.40% to 58.59% on the AVE test dataset [33] (see Fig. 3(c)). The experiment shows that although MFMs can support flexible modality combinations, their accuracy still degrades with (partially) incomplete modalities caused by asynchronously arriving data.

Opportunities: Inspired by the *conditional computation* principle for efficient inference [41], [42], we explore the accuracy-latency trade-off by investigating *whether all modalities are equally important for every sample.* Fig. 4(a) zooms into the accuracy and latency of five classes when performing MFM inference with either only audio or both audio and vision inputs, using the same model and test dataset as Fig. 3. We observe that the accuracy of certain classes remains high even when using only audio data (*e.g.*, 71.42% with audio only and 68.7% with both audio and vision for the class of Mandolin). Since inference for these classes does not wait for vision data, its latency is reduced from 0.454 s to 0.143 s. Fig. 4(b) shows similar sample-wise results, where we count the number of samples that can be correctly processed by audio-only or audio-vision fusion. For example, for the class of Mandolin, 96.33% of samples can be correctly processed with audio only, while merely 3.67% need to be processed with audio-vision fusion. The above results imply the potential to *dynamically* trigger inference upon receiving a *subset* of *important* multimodal data for *different samples*, rather than *blindly* waiting for *all modalities to arrive*. Furthermore, prior work [29] has shown that lightweight on-device models and hierarchical inference can effectively reduce system overhead.

Motivated by this opportunity, we theoretically analyze the optimal execution policy for dynamic MFM inference, as illustrated in Fig. 4(c). The analysis shows that compared with existing methods, *progressive multimodal inference* can potentially improve accuracy by 38.72% while reducing latency by 73.70%. This observation motivates the design of our synchronization-free MFM inference framework via progressive data orchestration for efficient multimodal intelligence.

III. System Overview

This section provides an overview of AdaModalX, an *orthogonal runtime orchestration layer* for accurate, low-latency MFM inference at the edge, over distributed multimodal inputs arriving asynchronously from mobile devices. For ease of presentation, we assume that each mobile device only collects, processes, and sends *single-modality* data to the edge for MFM inference. Our solution easily applies to mobile devices with multimodal sensors.

Rationales: As in Section II-B, AdaModalX improves the accuracy-latency trade-off by *dynamically* triggering MFM inference on a *subset* of *important* modalities rather than waiting

IEEE Proof

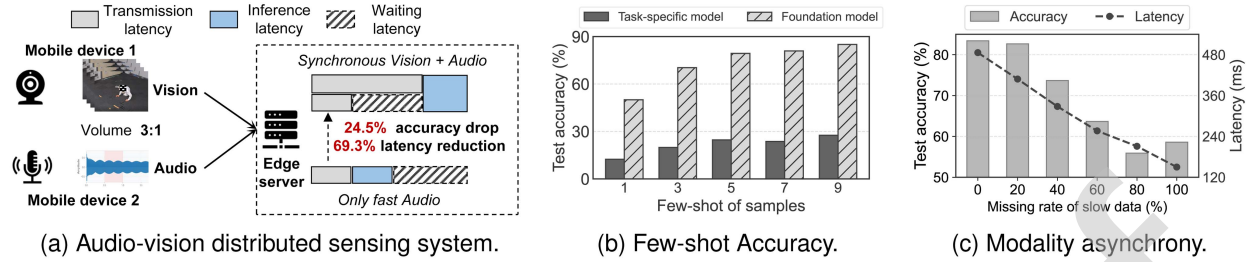


Fig. 3. Impact of modality asynchrony on MFM inference. (a) Illustration of asynchronous data in an audio-vision-based distributed mobile sensing system. (b) Inference accuracy with different few-shot settings. (c) Inference latency and accuracy with full audio data but partial vision data.

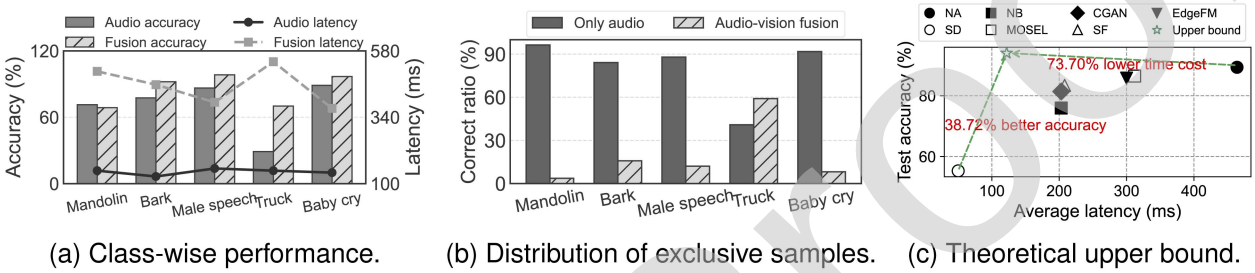


Fig. 4. Impact of modalities on different samples. (a) Sample-wise inference accuracy and latency with audio-only *v.s.* both audio and vision inputs. (b) Distribution of mutually exclusive data samples correctly processed by different modalities. (c) Theoretical upper bound of dynamic multimodal inference, illustrating the potential accuracy gain and latency reduction over existing methods.

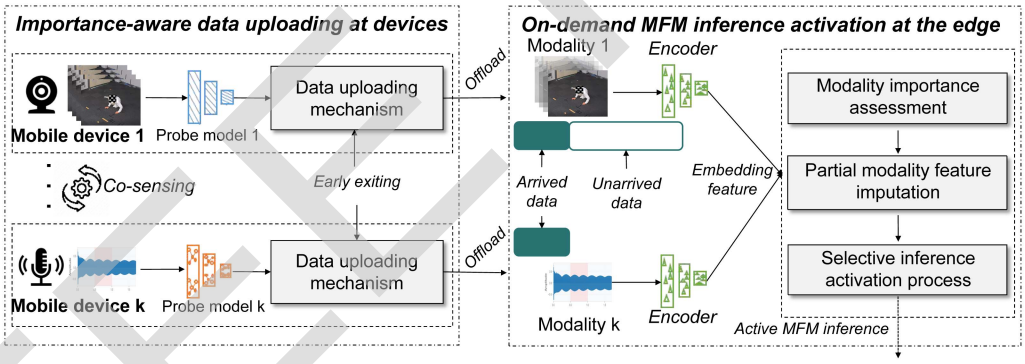


Fig. 5. Overview of AdaModalX for MFM-based distributed sensing systems. Partial inputs are arrived data available for feature extraction; arrival states indicate per-modality upload progress, not cross-modal data-size correspondence.

for additional delayed modalities. We realize this idea in distributed mobile sensing systems via the following principles.

- *Progressive Multimodal Inference:* We learn to assess incremental utility for diverse inputs offline by reusing the frozen MFM encoders and training a shared lightweight assessor. During online inference, we trigger MFM inference once the available modalities, including arrived samples and partially received delayed inputs when needed, satisfy semantic sufficiency.
- *Hierarchical Inference:* We utilize mobile devices that can execute lightweight on-device models to perform inference on *local* samples first and only upload *uncertain cases* to the edge for MFM inference. This further reduces the overall latency without compromising accuracy.

Workflow: AdaModalX consolidates the above rationales into a standard multi-device edge architecture. Fig. 5 illustrates the workflow of AdaModalX, which consists of two modules.

- *Importance-aware Data Uploading at Devices* (Section IV). Each mobile device runs a lightweight confidence estimator distilled from the edge-side MFM to compute class-wise semantic similarities. AdaModalX derives upload importance from the top-1 similarity and the margin between the top-1 and top-2 similarities. Under the joint confidence rule, samples meeting class-wise importance thresholds exit locally; otherwise, uploading continues for edge-side MFM inference.
- *On-demand MFM Inference Activation at Edge* (Section V). The edge caches multimodal data

IEEE Proof

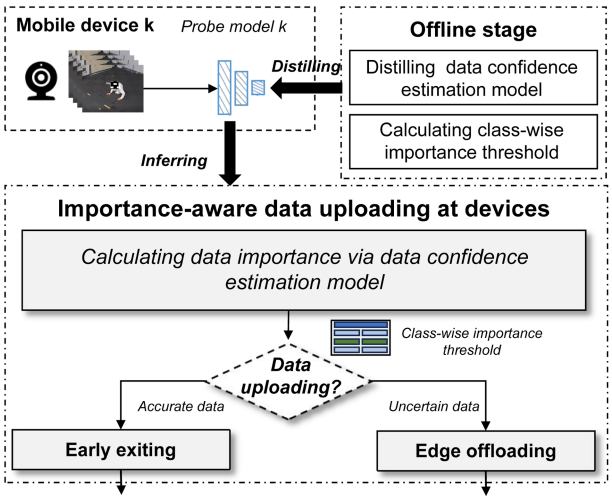


Fig. 6. Overview of importance-aware data uploading at devices (Section IV).

asynchronously uploaded from mobile devices and evaluates a candidate formed by fully received modalities using a lightweight contribution assessor. If the candidate provides insufficient utility, AdaModalX imputes a feature from the partially received input of the most informative incomplete modality and re-evaluates the augmented candidate. It triggers MFM inference once the candidate satisfies the activation rules.

Next, we elaborate on the detailed designs of the two modules.

IV. Importance-Aware Data Uploading At Devices

As shown in Fig. 6, this module enables each mobile device to decide whether to upload a data sample for edge-side MFM inference using a *confidence-based* importance metric. It leverages a lightweight *confidence estimation model* (Section IV-A) to obtain sample-wise inference results and confidence locally, which the *data uploading mechanism* (Section IV-B) then maps to upload importance. Samples with high confidence exit early from the mobile device, whereas only uncertain samples are uploaded for MFM inference.

A. Data Confidence Estimation

We develop a lightweight probe model that estimates the inference confidence of each local sample on each device and uses low-confidence evidence as a proxy for upload importance. To obtain accurate confidence estimates under device resource constraints, we distill the probe from the edge-side MFM.

Model Architecture: The confidence estimator consists of three modules: a *modality-specific feature extractor*, a *feature projector*, and a *semantic contrastive classifier*.

- The feature extractor uses lightweight modality-specific backbones for resource-constrained mobile devices, such as CNN14-based models [43] and CRNN [44] for audio, and ResNet-18 [45] and MobileNetV3-Small [46], [47] for vision.
- The feature projector aligns embeddings produced by each modality-specific device-side estimator with the semantic space of the edge-side MFM, providing a unified distillation target across heterogeneous devices and modalities.
- The semantic contrastive classifier computes class-wise semantic similarities and produces both the local prediction and its confidence score.

Why Distillation: We train the confidence estimator by distilling knowledge from the edge-side MFM rather than by training an independent lightweight classifier, for two reasons: *First*, the estimator is not a standalone classifier; instead, it estimates each sample's *upload importance* for device-side early-exit control. Distillation from the edge-side MFM provides an *informative decision boundary*, reducing unnecessary uploads while avoiding unreliable early exits. *Second*, the edge-side MFM provides unified supervision across heterogeneous devices and modalities, whereas each device solves only a simpler local subproblem using its local inputs. This process transfers shared semantic knowledge to lightweight models, especially when local supervision is scarce.

Confidence-guided Prototype Distillation: The lightweight estimator jointly produces local predictions and confidence scores to determine whether a sample should exit locally or be uploaded to the edge-side MFM. Conventional distillation methods [48], [49] mainly improve prediction accuracy by transferring logits or feature representations, without explicitly optimizing the confidence margin that governs upload decisions. This may lead to unnecessary uploads and unreliable early exits. To address this issue, we introduce a *confidence-guided prototype distillation* loss.

- *Feature-alignment distillation loss:* Let z_i and e_i denote the device-side and MFM semantic embeddings of sample x_i, respectively. We define the feature-alignment loss as

$$\mathcal{L}_{\text{FA}} = \frac{1}{N}\sum_{i=1}^{N} \|z_i - e_i\|_2^2. \tag{1}$$

  Minimizing $\mathcal{L}_{\text{FA}}$ aligns device-side embeddings with the MFM semantic space, preserving the discriminative structure of the edge-side MFM.
- *Pseudo-text contrastive distillation loss:* Using the MFM embedding, we select the nearest class text prototype as the pseudo target $\tilde{t}_i$ and obtain its confidence q_i. Building on CLIP-style bidirectional contrastive learning [39], [50], we treat $(z_i, \tilde{t}_i)$ as a positive pair and the remaining mini-batch pairs as negatives, and weight the two directional cross-entropy terms by q_i, normalized by $\sum_i q_i$, to reduce the effect of uncertain pseudo targets:

$$\mathcal{L}_{\text{PTC}} = \frac{1}{2}\left(\frac{\sum_i q_i\, \ell_{\text{CE}}\left(z_i \tilde{T}^\top/\tau,\, i\right)}{\sum_i q_i} + \frac{\sum_i q_i\, \ell_{\text{CE}}\left(\tilde{t}_i Z^\top/\tau,\, i\right)}{\sum_i q_i}\right). \tag{2}$$

  where z_i denotes the device-side embedding, $\tilde{t}_i$ the pseudo-text target, q_i the confidence weight, Z and $\tilde{T}$ the sets of device-side and pseudo-text embeddings, and τ the temperature.

IEEE Proof

- *Class-prototype distillation loss:* Inspired by cross-modal self-distillation [51], we adopt a cosine-similarity triplet loss to transfer semantic discrimination from text embeddings:

$$\mathcal{L}_{\text{CPD}} = \max\left(0,\, \lambda - \text{sim}(z_i, t_i^+) + \text{sim}(z_i, t_i^-)\right), \quad (3)$$

where z_i denotes the device-side embedding, t_i^+ and t_i^- denote the positive and negative class text embeddings, $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and λ is the margin.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{PTC}} + \alpha\mathcal{L}_{\text{FA}} + \beta\mathcal{L}_{\text{CPD}}, \quad (4)$$

where α and β are weighting hyperparameters. We empirically set $\alpha = 1$, $\beta = 1$, and $\lambda = 0.3$ on the AVE dataset [33].

Confidence-aware Upload Importance Estimation: Before all modality data arrive at the edge, each device uses its locally captured sample to compute class-wise semantic similarities between the device-side embedding and class prototypes, as defined in (6). These similarities provide local confidence estimates for upload decisions, rather than class-wise probabilities over the global multimodal input. Given the confidence estimator, one can quantify inference confidence using an auxiliary model [52], [53] or multiple forward/backward passes [54], [55]; however, these methods introduce non-trivial computation and latency overhead. For rapid confidence estimation on data streams, we follow the softmax-based methods [56] to quantify inference confidence (and thus data importance).

Specifically, we characterize inference confidence using two complementary metrics: the top-1 similarity score $s_{(1)}(x_i)$, which reflects the *absolute confidence* of the prediction, and the margin score $M_{\text{score}}(x_i)$, which captures *relative class separability*. The margin score is defined as

$$M_{score}(x_i) = s_{(1)}(x_i) - s_{(2)}(x_i), \quad (5)$$

where

$$s_k(x_i) = \text{sim}(E_d(x_i), c_k), \qquad c_k \in C, \quad (6)$$

and $s_{(1)}(x_i)$ and $s_{(2)}(x_i)$ denote the highest and second-highest similarities among all class prototypes. The margin score captures relative uncertainty, whereas $s_{(1)}(x_i)$ reflects absolute confidence. Each metric alone is insufficient: the former ignores overall confidence levels, and the latter overlooks inter-class ambiguity. Combining them yields a more robust characterization of upload importance.

B. Data Uploading Mechanism

We therefore design a rule-based decision mechanism that jointly considers absolute confidence and relative class separability and uses *class-wise* importance thresholds to decide whether to upload data or exit early.

Joint confidence rules: To make upload decisions robust, we jointly evaluate absolute confidence and relative class separability. Specifically, we define two complementary decision rules:

- *High-confidence rule:*

$$s_{(1)}(x_i) > \tau_{top1}^{high} \quad \text{or} \quad M_{\text{score}}(x_i) > \tau_{gap}^{high}, \quad (7)$$

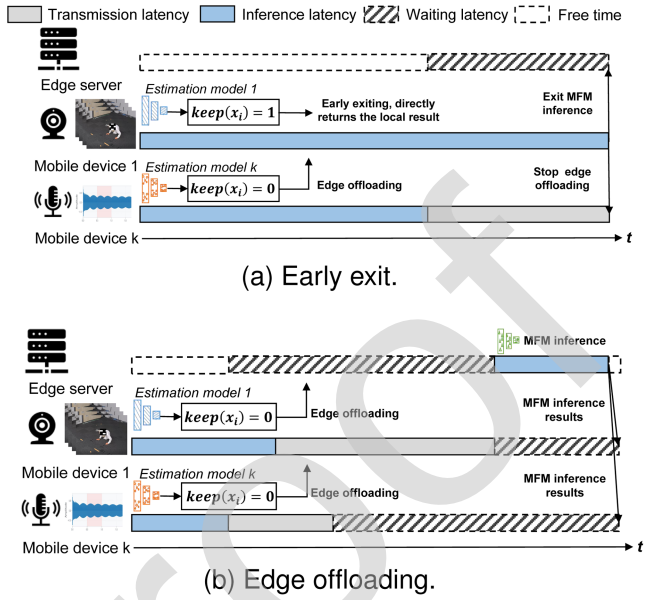


Fig. 7. Illustration of the confidence-aware uploading mechanism under multi-device coordination. Each mobile device performs local probe inference in parallel with data uploading. (a) If the estimated confidence is sufficiently high, the device triggers early exit and produces a local prediction. (b) If the estimated confidence is insufficient, the device continues uploading, and the edge-side MFM completes the inference.

- *Pairwise-confidence rule:*

$$s_{(1)}(x_i) > \tau_{top1}^{pair} \quad \text{and} \quad M_{\text{score}}(x_i) > \tau_{gap}^{pair}. \quad (8)$$

The device treats sample x_i as *locally reliable* if either rule is satisfied:

$$\text{keep}(x_i) = \mathbb{I}(\text{high-conf} \ \vee \ \text{pair-conf}). \quad (9)$$

This hybrid rule improves decision robustness by filtering out low-confidence predictions and boundary-ambiguous samples, thereby reducing unnecessary offloading while avoiding unreliable early exits.

Class-wise Importance Thresholds: A single global threshold is suboptimal for heterogeneous devices and modalities because their score distributions can differ substantially. Recent studies [29] set device-level thresholds but overlook intra-modality variations across semantic classes. Using a validation set, we perform an accuracy-constrained search over candidate thresholds derived from sample scores and select the smallest thresholds that satisfy the target accuracy. We use different targets for the two rules: a higher target for the high-confidence rule to ensure reliability and a lower target for the pairwise-confidence rule to maintain coverage. To improve robustness, we incorporate two mechanisms: (i) falling back to global thresholds when class-wise samples are insufficient and (ii) disabling local early exit when no feasible threshold satisfies the target accuracy.

Runtime Data Uploading Decision: At runtime, each mobile device independently computes $s_{(1)}(x_i)$ and $M_{\text{score}}(x_i)$ from its local input stream. It then applies the calibrated class-wise thresholds to decide whether to exit early or continue uploading,

IEEE Proof

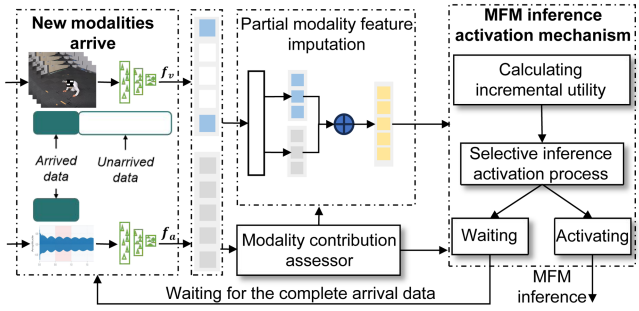


Fig. 8. Overview of on-demand MFM inference activation at the edge.

as illustrated in Fig. 7.

$$\text{decision}_i = \begin{cases} \text{Early exiting}, & \text{keep}(x_i) = 1, \\ \text{Edge offloading}, & \text{otherwise}. \end{cases} \tag{10}$$

- *Local early exiting:* If the confidence rules are satisfied, the device treats its prediction as reliable, stops uploading, and returns the local inference result.
- *Continued uploading:* Otherwise, the device continues uploading its data stream for edge-side MFM inference.

To minimize latency, we use an opportunistic early-exit strategy: once any device triggers early exit, the system immediately terminates further uploads and uses that device's prediction as the final result. If multiple devices satisfy the early-exit condition, the system selects the result from the device that exits first. This design prioritizes low-latency responses while maintaining reliability through confidence-aware filtering, thereby balancing accuracy and system responsiveness.

V. On-Demand MFM Inference Activation At Edge

This module runs at the edge and determines when to activate MFM inference under asynchronous arrivals of multimodal inputs. Let $\{x_1, x_2, \ldots, x_K\}$ denote offloading samples from K modalities. To avoid blocking inference until all K modalities arrive, AdaModalX formulates MFM activation as progressive data orchestration. Upon the complete arrival of a modality sample, the edge evaluates the incremental utility of the available modalities and triggers MFM inference once the utility score of the current candidate meets its candidate-specific activation threshold. This design consists of two modules: a *modality contribution assessor* (Section V-A) and an *inference activation mechanism* (Section V-B)(see Fig. 8).

A. Modality Contribution Assessor

We insert a simple assessor between the encoders and the fusion network and train it to estimate the incremental utility of different modality combinations by learning the discrepancy between a frozen MFM's predictions and the true labels.

Utility Score Definition: For each modality k, the frozen MFM encoder extracts its semantic feature $f_k = E_k(x_k)$. Given $F_S = \{f_k \mid k \in S\}$, the assessor predicts a utility score $s(F_S) \in [0, 1]$, which estimates whether the candidate feature set contains sufficient information for reliable MFM inference. A higher $s(F_S)$ indicates that the candidate set is more informative for the current sample. Since modality contribution varies across samples and modality combinations, AdaModalX retains all fully received modality features rather than permanently discarding any modality, while avoiding unnecessary waiting by activating inference once the current candidate meets its utility threshold.

Model Architecture: Since training a separate model for all $2^K - 1$ modality subsets is impractical, we introduce a shared lightweight assessment model to estimate the utility of each candidate modality set. The frozen MFM encoders map heterogeneous modalities into a unified latent space, allowing the same assessment model to evaluate different modality combinations. We process the candidate feature set F_S with an assessor encoder selected according to the fusion layer of the underlying MFM. For LLM-based fusion, we use a lightweight Transformer encoder and take the CLS output as the task-specific semantic representation: $h_S = Trans_\theta(F_S)_{cls}$. For MLP-based fusion, we use an MLP encoder: $h_S = MLPEnc_\theta(F_S)$. The utility score is then computed as

$$s(F_S) = \begin{cases} \sigma(g_\theta(h_S, v_{\text{stat}})), & \text{MLP-based fusion}, \\ \sigma(g_\theta(h_S)), & \text{LLM-based fusion}. \end{cases} \tag{11}$$

where v_{stat} denotes statistical features used only for MLP-based fusion, and $\sigma(\cdot)$ maps the output to [0,1].

Offline Training: For offline training, we statistically profile the modality arrival order and construct candidate inputs accordingly to train the assessor. For each candidate feature set F_S, we derive the supervisory signal y_S from the predictions of a frozen MFM and the ground-truth labels, and train the contribution assessor by minimizing $L_{\text{imp}} = \sum_S \left(s(F_S) - y_S\right)^2$. We formulate the selection of candidate-specific activation thresholds as an accuracy-constrained latency minimization problem. Let $\text{Acc}_{F_S}(V)$ and $\text{Lat}_{F_S}(V)$ denote the validation accuracy and average latency under the activation rule $s(F_S) \geq V$, respectively. The threshold for F_S is selected by solving

$$V_{F_S} = \arg \min_{V \in \mathcal{V}_{F_S}} \text{Lat}_{F_S}(V), \quad \text{s.t.} \quad \text{Acc}_{F_S}(V) \geq A_{\text{tar}},$$

where $\mathcal{V}_{F_S}$ contains the candidate thresholds derived from validation-set utility scores, and A_{tar} is the target accuracy. We solve this problem by grid search and select the feasible threshold with the minimum average latency. During inference, AdaModalX triggers MFM inference once the current candidate satisfies $s(F_S) \geq V_{F_S}$.

For dynamic arrival orders and modality subsets whose candidate feature sets F_S are covered during offline training, AdaModalX reuses the shared contribution assessor and corresponding V_{F_S} without online retraining, reselecting only the thresholds when deployment conditions change. For an unseen combination of MFM-supported modalities, it keeps the MFM frozen, updates only the assessor, and selects the corresponding V_{F_S} on a validation set. An MFM-unsupported modality additionally requires an encoder or projector to map it into the MFM semantic space before the assessor is updated and the corresponding thresholds are selected offline.

IEEE Proof

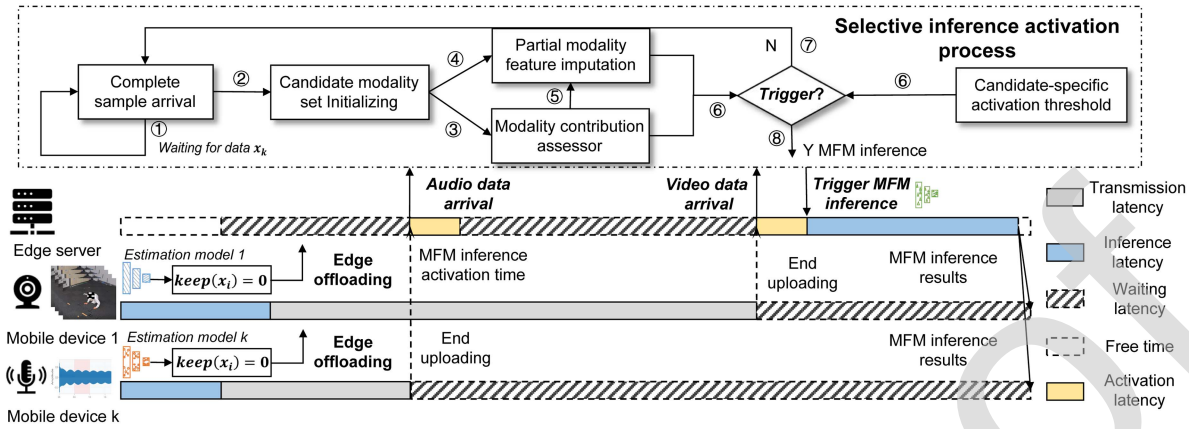


Fig. 9. Overview of the MFM inference activation mechanism. Mobile devices 1 to k use a confidence-aware uploading mechanism to continuously upload multimodal data. Once the audio sample arrives completely (①), it triggers MFM inference activation (②), calculates the modality utility via assessor (③), uses partial-modality feature imputation to recover complementary multimodal information (④), predicts the imputed feature (⑤), and combines them into a candidate set to check if the utility meets its candidate-specific activation threshold (⑥). If the candidate set does not meet the threshold (⑦), AdaModalX waits for the next data arrival (①); otherwise, it triggers MFM inference (⑧).

B. MFM Inference Activation Mechanism

Instead of relying on complete modality arrival synchronization [28] or strict maximum latency deadlines [18], we progressively evaluate whether to execute MFM inference or await further inputs whenever a new modality is fully received.

For ease of presentation, we assume that the samples x_k arrive at the edge in ascending order; *i.e.*, the complete sample x_k (*e.g.*, an audio frame) is received earlier than x_{k+1} (*e.g.*, a video clip). Once x_k fully arrives, AdaModalX updates the current candidate $\mathcal{F}_k$ with the features of $\{x_1, \ldots, x_k\}$. If $k = K$, AdaModalX directly runs MFM inference. Otherwise, it evaluates $\mathcal{F}_k$ and triggers inference if it meets its activation rule. If $\mathcal{F}_k$ fails its threshold, AdaModalX combines the partial feature f_p^{part} of the most informative incomplete modality with $\mathcal{F}_k$ to impute $\hat{f}_p$ and evaluates the augmented candidate $\widetilde{\mathcal{F}}_k$. AdaModalX waits for the next modality only if the augmented candidate also fails its threshold.

AdaModalX implements the above decision process with two optimizations: *(i)* partial modality feature imputation and *(ii)* a selective inference activation process, as explained next.

Partial Modality Feature Imputation: Rather than relying only on fully received modality inputs or waiting for the remaining modality samples to arrive completely, we exploit partially received modalities via *feature-level imputation*. For the most informative incomplete modality $x_p \in x_{k+1}, \ldots, x_K$, we encode its partially received input into a feature f_p^{part} and impute the missing task-relevant semantics conditioned on the available feature set $\mathcal{F}_k$. The resulting augmented candidate $\tilde{\mathcal{F}}_k = \mathcal{F}_k \cup \hat{f}_p$ enables earlier MFM activation while reducing accuracy degradation due to incomplete modality arrivals.

Specifically, AdaModalX performs feature imputation in the shared semantic space of the MFM, rather than reconstructing the raw missing data [57]. Given the available feature set $\mathcal{F}_k = f_1, \ldots, f_k$ and the partial feature $f_p^{\text{part}} = E_p(x_p^{\text{part}})$ produced by the modality encoder, we extract compact cross-modal context tokens as

$$C_k = \text{Ctx}_\theta(\mathcal{F}_k \cup \{f_p^{\text{part}}\}), \tag{12}$$

where $\text{Ctx}_\theta(\cdot)$ is a modality-aware token compressor that projects heterogeneous modality features into a unified hidden space and summarizes them through lightweight token mixing and latent-query cross-attention. Conditioned on the partial feature f_p^{part} and the compact context C_k, the model generates a task-relevant semantic residual:

$$\Delta f_p = G_\theta(f_p^{\text{part}}, C_k), \quad \hat{f}_p = f_p^{\text{part}} + \Delta f_p. \tag{13}$$

Here, $G_\theta(\cdot)$ is implemented as a lightweight cross-attention imputer followed by an MLP residual head. By adding only the predicted residual, the imputation preserves the semantic evidence already contained in the partial feature while recovering complementary information for early MFM activation. In the offline stage, we freeze the MFM and train the imputation module using the following objective: $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{reg}}$, where λ is set to 0.2, and $\mathcal{L}_{\text{reg}}$ denotes the regularization loss. Here, $\mathcal{L}_{\text{task}}$ is the task-specific loss (*e.g.*, an MLLM uses the causal language modeling loss).

Selective Inference Activation Process: When x_k is fully received, the modality samples $\{x_1, \ldots, x_k\}$ are complete, whereas $\{x_{k+1}, \ldots, x_K\}$ remains incomplete (see Fig. 9).

- AdaModalX incrementally encodes the newly arrived sample x_k into f_k and updates the cached feature set as $\mathcal{F}_k = \mathcal{F}_{k-1} \cup \{f_k\}$. If $k = K$, all modality samples have arrived, and AdaModalX directly performs MFM inference with $\mathcal{F}_K$. Otherwise, it first evaluates the candidate formed by fully received modalities. If this candidate satisfies the activation rule, i.e., $s(\mathcal{F}_k) \geq V_{\mathcal{F}_k}$, AdaModalX triggers MFM inference with $\mathcal{F}_k$.
- If the complete candidate set is insufficient, AdaModalX selects the most informative incomplete modality x_p from $\{x_{k+1}, \ldots, x_K\}$ for imputation rather than simply waiting for the next modality sample to become complete. AdaModalX encodes the partial modality sample x_p^{part}

IEEE Proof

into a partial feature f_p^{part}. The imputation module then combines f_p^{part} with $\mathcal{F}_k$ to generate an augmented feature $\hat{f}_p$. The assessor scores the augmented candidate $\widetilde{\mathcal{F}}_k = \mathcal{F}_k \cup \{\hat{f}_p\}$ to make the next activation decision. If this candidate satisfies the activation rule, i.e., $s(\widetilde{\mathcal{F}}_k) \geq V_{\widetilde{\mathcal{F}}_k}$, AdaModalX triggers MFM inference with $\widetilde{\mathcal{F}}_k$; otherwise, it waits until the next modality sample from the mobile client is fully received at the edge.

VI. Experiment

A. Experiment Setup

Implementation: We implement AdaModalX in Python 3.12.3 and PyTorch 2.5.1 for the server and mobile devices. The edge server is equipped with an RTX 3090 GPU, Intel(R) Xeon(R) Gold 6133 CPU @ 2.50 GHz, and 256 GB of RAM. We evaluate AdaModalX on four mobile devices: Jetson Orin NX (D_1), Jetson Orin Nano (D_2), Xiaomi 13 Ultra (D_3), and Vivo X300 Pro (D_4). To simulate the distributed multimodal asynchronous arrival problem, we also evaluate the system's performance across varying network bandwidths. For importance-aware data uploading (Section IV), we use a shallow MLP for text and lightweight CNNs for audio and vision as data-confidence estimators. We repeat each experiment five times with different random seeds and report the mean $\pm$ standard deviation. For key comparisons with small performance differences, we additionally conduct statistical significance tests, with the results indicated in the corresponding figures.

Task, Datasets, and Model: We implement two representative MFM inference architectures to evaluate system performance:

- *MFM-based Inference* (M_1): This architecture consists of a multimodal encoder, a projector, a fusion module, and a contrastive classifier. It maps inputs into a unified space, fuses them, and matches the fused representation against class text embeddings for classification. We implement it using ImageBind [39]. For example, in event detection tasks, the prompt template is: "Event is CLS."
- *MLLM Inference* (M_2): This architecture uses the complete MLLM to directly output the final class label texts from multimodal inputs and task prompts. We implement it using Qwen3.5-2B [58]. For example, in multimodal sentiment analysis [34], the prompt is: "You are a multimodal sentiment classification expert. Based on the input, classify the sentiment as positive, neutral, or negative. The sentiment is ()."

We further evaluate AdaModalX on four representative real-world distributed multimodal applications:

- *Event Detection* (T_1): Audio-visual event recognition on the AVE dataset [33], which contains 4,143 videos across 28 event categories; we divide each video into 1-second clips for recognition.
- *Driver Activity Recognition* (T_2): Multimodal driver behavior understanding on drive&Act [35], a multi-view, multi-modal driver activity dataset.
- *Human Activity Recognition* (T_3): Human action recognition on the Kinetics benchmark [59], a large-scale video-audio dataset for action classification.
- *Human Sentiment Analysis* (T_4): Sentiment classification on CMU-MOSI [34], which contains aligned text, audio, and visual modalities.

Baseline: We compare AdaModalX with *nine* representative baselines that span both synchronous and asynchronous multimodal inference. For fair comparison, we reproduce all baseline methods in the same evaluation pipeline and evaluate them under the same settings. The baselines include:

- *Synchronous MFM inference:* The inference pipeline waits for all input data to arrive before the server performs MFM inference on the complete modalities.
- *No-adaptation (NA) [39], [58]:* The edge server waits for all modalities to arrive and then runs the MFM inference using all modalities.
- *Modality Gating (MG) [28], [60], [61]:* It uses a lightweight policy network to dynamically select input modalities for MFM inference, thereby reducing latency and computational overhead.
- *Minimal Modality Inference (Min) [62]:* Triggers inference immediately with the fastest-arriving modality (*e.g.*, audio), prioritizing early inference but possibly sacrificing accuracy.
- *Asynchronous MFM inference [21]:* The MFM inference pipeline can perform inference when partial multimodal data arrive.
  - *Original Non-Blocking (NB) [21]:* It directly runs MFM inference on the currently available partial multimodal input once one modality fully arrives.
  - *Standalone (SD) [45]:* This runs a lightweight single-modal model, distilled from the MFM, exclusively on mobile devices for local inference.
  - *MOSEL [18]:* It dynamically selects a subset of available modalities for MFM inference to meet latency and accuracy requirements.
  - *CGAN-based Modality Imputation (CGAN) [20], [21], [31], [57]:* It uses a CGAN to impute slow modality features, thereby improving inference accuracy.
  - *Shared Feature (SF) [30]:* It uses a specific- and shared-feature network to reconstruct missing modality features for asynchronous MFM inference.
  - *EdgeFM [29]:* It selectively uploads uncertain samples to edge-side FM and performs model switching according to data uncertainty and network conditions.

B. Performance Comparison

1) System Latency v.s. *Accuracy:* We evaluate the inference accuracy and system latency of AdaModalX under different network conditions across four tasks (T_1–T_4) and two multimodal foundation models (M_1 [39] and M_2 [58]), comparing it with nine baselines (NA [39], MG [28], Min [62], NB [21], SD [45], MOSEL [18], CGAN [57], SF [30], and EdgeFM [29]). *First*, as in Fig. 10, AdaModalX achieves the best accuracy-latency trade-off across tasks, models, and network conditions. Compared with NA [39], it reduces the average latency by 50.89% over the four tasks and two models, with less than 2% accuracy degradation. Compared with Min [62], it improves the average accuracy by 21.52%. *Second*, it achieves the lowest

IEEE Proof

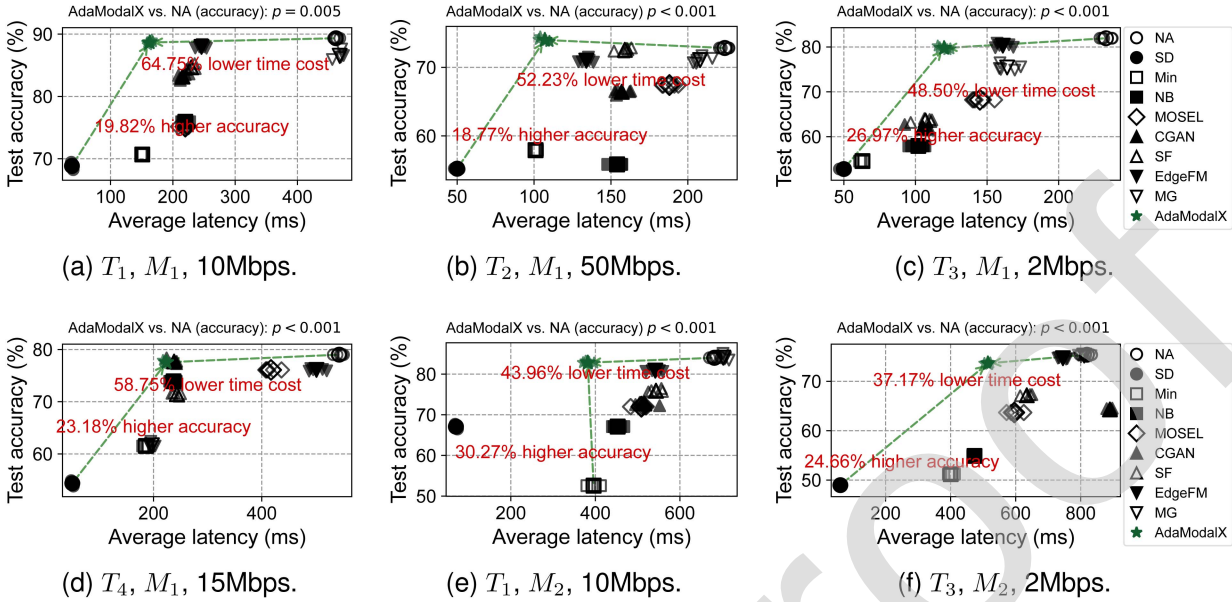


Fig. 10. Comparison of accuracy *v.s.* latency between AdaModalX and other baselines on diverse tasks. Translucent markers show the five individual runs, while opaque markers show their means. The annotations report paired-test p-values for the accuracy comparison between AdaModalX and NA.

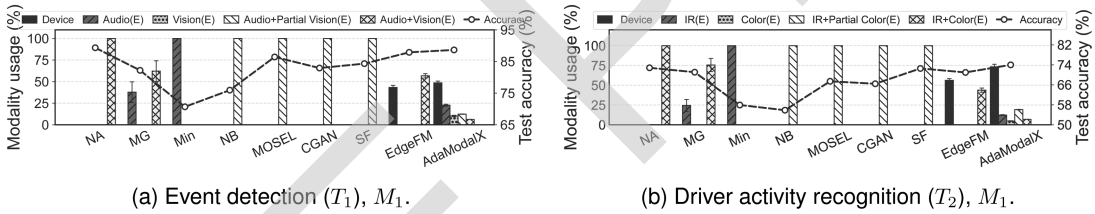


Fig. 11. Comparison of modality usage rate(%) *v.s.* accuracy(%) between AdaModalX and other approaches on diverse tasks.

system latency while maintaining competitive accuracy. For example, as shown in Fig. 10(b), on the T_2 task under 50 Mbps network bandwidth, it reduces latency by 52.23%, 48.56%, 30.52%, 43.09%, 31.85%, 32.70%, and 20.15% compared with NA [39], MG [28], NB [21], MOSEL [18], CGAN [57], SF [30], and EdgeFM [29], respectively. It also improves accuracy by 16.07% and 18.77% compared with Min [62] and SD [45], respectively.

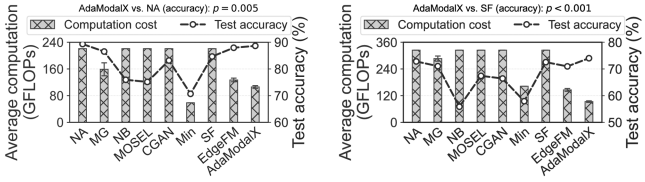
Summary: AdaModalX achieves the best accuracy-latency trade-off in distributed asynchronous multimodal sensing via importance-aware uploading and on-demand activation.

2) Modality Usage: We evaluate the modality usage of AdaModalX under different network bandwidths for the event detection and driver activity recognition tasks, comparing it with baselines that require data uploading for edge-side inference, including NA [39], MG [28], Min [62], NB [21], MOSEL [18], CGAN [57], SF [30], and EdgeFM [29]. *First*, AdaModalX achieves the lowest edge-side modality usage among all methods. As shown in Fig. 11(b), 73.31% of samples are inferred directly on mobile devices, improving the early-exit ratio by 17.10% compared with EdgeFM [29] by avoiding unnecessary data uploading. *Second*, AdaModalX significantly reduces the usage of slow modalities. Compared with MG [28] and NA [39] in Fig. 11, it reduces the usage of the slow modality by 53.76% and 71.25%, respectively. *Third*, it also reduces the usage of the IR-Color modality combination by 93.11% compared with NA [39] and reduces the usage of the IR-Partial Color modality combination by 80.83% compared with NB [21], MOSEL [18], CGAN [57], and SF [30].

Summary: AdaModalX reduces modality usage and latency through early exiting at devices and on-demand activation at the edge while preserving competitive accuracy.

3) Computational Cost: We further evaluate the computational cost and inference accuracy of AdaModalX using M_1 on the event detection (T_1) and driver activity recognition (T_2) tasks under 10 Mbps and 20 Mbps, comparing it with eight baselines: NA [39], MG [28], Min [62], NB [21], MOSEL [18], CGAN [57], SF [30], and EdgeFM [29]. *First*, AdaModalX achieves the lowest computational cost while maintaining competitive accuracy across different tasks, as shown in Fig. 12. For example, in Fig. 12(b), AdaModalX reduces the computational cost by 71.31%, 67.52%, 71.31%, 71.31%, 71.31%, 42.61%, 71.31% and 36.11% compared with NA [39], MG [28], NB [21], MOSEL [18], CGAN [57], Min [62], SF [30], and EdgeFM [29], respectively. It improves accuracy by 2.97%, 16.11%, 18.18%, 6.61%, 7.60%, and 1.48% compared with MG [28], Min [62], NB [21], MOSEL [18], CGAN [57], and SF [30], respectively.

IEEE Proof



(a) T_1, M_1, 10Mbps. (b) T_2, M_1, 20Mbps.

Fig. 12. Comparison of computational cost between AdaModalX and other approaches on diverse tasks. The annotations report paired-test p-values for AdaModalX versus NA in (a) and AdaModalX versus SF in (b).

TABLE I
ABLATION STUDY ON IMPORTANCE-AWARE DATA UPLOADING AT DEVICES (#D1) AND ON-DEMAND INFERENCE AT EDGE (#D2) ON T_1 AND T_2 TASKS UNDER DIFFERENT BANDWIDTHS

Method	Event detection (T_1), 10 Mbps		Driver activity recognition (T_2), 20 Mbps	
	Accuracy (%)	Latency (ms)	Accuracy (%)	Latency (ms)
w/o #D1, w/o #D2	89.35 ± 0.00	462 ± 4.05	72.82 ± 0.00	383 ± 3.72
w/ #D1, w/o #D2	87.93 ± 0.25	258 ± 5.94	73.44 ± 0.32	179 ± 4.18
w/o #D1, w/ #D2	**88.73 ± 0.63**	234 ± 2.86	**75.25 ± 0.41**	247 ± 3.67
w/ #D1, w/ #D2	88.66 ± 0.28	**163 ± 3.93**	73.99 ± 0.27	**107 ± 3.26**

Summary: AdaModalX effectively reduces computational costs while maintaining competitive accuracy via importance-aware uploading and on-demand inference activation.

4) Bandwidth Sensitivity: We evaluate system latency and accuracy on the driver activity recognition task (T_2) under different network bandwidths (*i.e.*, 10 Mbps, 20 Mbps, 25 Mbps, and 50 Mbps), as shown in Fig. 13. First, AdaModalX maintains competitive accuracy while consistently reducing latency across different network bandwidths. Compared with NA [39], AdaModalX reduces latency by 73.16%, 67.10%, 64.13%, and 52.23% under 10 Mbps, 20 Mbps, 25 Mbps, and 50 Mbps, respectively. *Second*, AdaModalX shows greater latency advantages as the network bandwidth decreases. For example, when the bandwidth decreases from 50 Mbps to 10 Mbps, the latency reduction of AdaModalX over NA [39], Min [62], NB [21], MOSEL [18], CGAN [57], SF [30], MG [28], and EdgeFM [29] further increases by 20.93%, 44.75%, 18.16%, 21.13%, 17.13%, 16.72%, 21.27%, and 26.33%, respectively.

Summary: AdaModalX maintains competitive accuracy across bandwidth settings, with larger latency advantages under bandwidth-constrained networks.

C. Understanding AdaModalX's Improvement

1) Impact of Design 1 (Section IV) and Design 2 (Section V): In Table I, we evaluate the impact of importance-aware data uploading at devices (#D1) and on-demand MFM inference activation at the edge (#D2) on the T_1 and T_2 tasks using M_1. *First*, enabling #D1 alone reduces latency by 48.71% on average while maintaining competitive accuracy, indicating that on-device early exit effectively avoids unnecessary data uploads for both T_1 and T_2. *Second*, enabling #D2 alone reduces latency by 42.43% on average, suggesting that on-demand MFM inference activation mitigates waiting for slow modalities. *Third*, jointly enabling #D1 and #D2 reduces latency by 68.39% on average while maintaining competitive accuracy, demonstrating the complementarity of the two designs for distributed multimodal mobile sensing.

2) Performance of Importance-Aware Data Uploading At Devices: We test overall accuracy and offloading rate of the device-side importance-aware data uploading module (#D1) on the T_1 and T_2 tasks using M_1 and compare it with EdgeFM(margin) [29], EdgeFM(max) [63], Softmax(margin) [29], Softmax(max) [63], AdaModalX (high), and AdaModalX (pair). Here, margin uses the gap between the top-1 and top-2 scores as the confidence metric, while max directly uses the top-1 score. AdaModalX (high) and AdaModalX (pair) use only the high-confidence rule or the pairwise-confidence rule. As shown in Fig. 14, #D1 achieves the lowest offloading rates of 51.44% and 26.68% on T_1 and T_2, respectively, while maintaining competitive overall accuracy. Specifically, compared with EdgeFM*(margin)* [29], EdgeFM*(max)* [63], Softmax*(margin)* [29], Softmax*(max)* [63], AdaModalX *(high)*, and AdaModalX *(pair)*, #D1 reduces the offloading rate by 17.10%, 17.01%, 20.21%, 17.64%, 5.99%, and 3.29% on T_2, respectively.

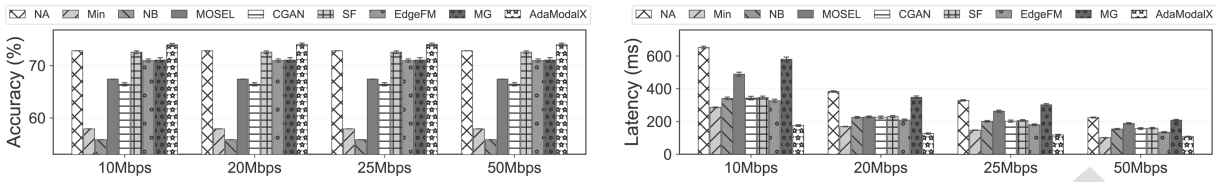
3) Performance of On-Demand MFM Inference Activation At Edge: We evaluate the accuracy and latency of the edge-side on-demand MFM inference activation module (#D2) on T_1, T_2, and T_3 tasks under diverse bandwidths, compared with seven baselines (NA [39], Min [62], NB [21], MOSEL [18], CGAN [57], SF [30], and MG [28]). As shown in Fig. 15, #D2 independently reduces waiting latency while maintaining competitive accuracy across tasks and bandwidth settings. Compared with NA [39], it reduces the average latency by 37.76% over the three tasks, with competitive accuracy, and it improves the average accuracy by 20.06% compared with Min [62]. For example, in Fig. 15(b) for T_2 under 50 Mbps, #D2 reduces latency by 30.36%, 17.02%, and 25.0% compared with NA [39], MOSEL [18], and MG [28], respectively.

4) Sensitivity to Slow-Modality Arrival Rates: In Fig. 16, we evaluate the performance of AdaModalX under different asynchronous settings and compare it with eight baselines (NA [39], MG [28], Min [62], NB [21], MOSEL [18], CGAN [57], SF [30], and EdgeFM [29]) on the T_1 task using M_1. We vary the arrival rate of the slow modality (vision), defined as the proportion of vision data that has arrived when the audio fully arrives. At the 50% arrival rate, it reduces latency by 65.9% over NA [39] while maintaining competitive accuracy, and reduces latency by 45.3% over the strongest non-NA baseline SF [30]. Second, AdaModalX remains effective under a lower arrival rate. At the 25% arrival rate, it reduces latency by 63.4% over NA [39] while maintaining competitive accuracy. These results demonstrate the robustness of AdaModalX to different degrees of modality asynchrony.

D. Micro-Benchmarks and Ablation Studies

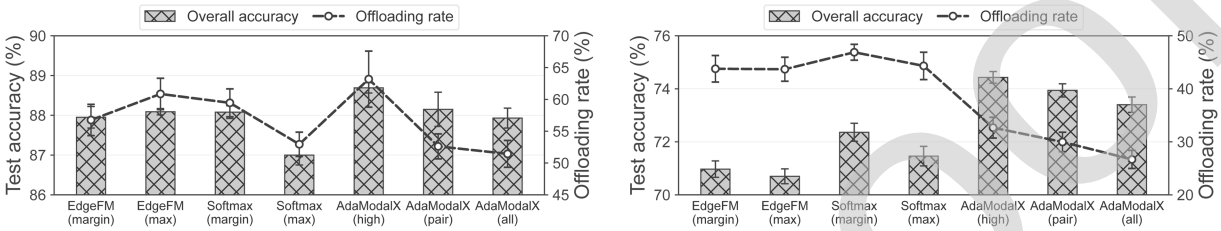
1) Impact of Different Distillation Loss: We evaluate the impact of different distillation loss components on the T_2 task by mobile model accuracy, overall accuracy, and offloading rate. W/ $\mathcal{L}_{PTC}$, w/ $\mathcal{L}_{FA}$, and w/ $\mathcal{L}_{CPD}$ achieves the lowest offloading rate while maintaining competitive accuracy in

IEEE Proof



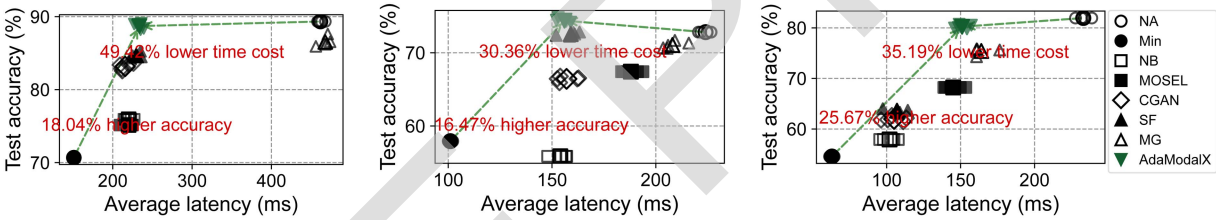
(a) T_2, M_1, Accuracy. (b) T_2, M_1, Latency.

Fig. 13. Bandwidth sensitivity of AdaModalX on T_2 with M_1 under 10 Mbps, 20 Mbps, 25 Mbps, and 50 Mbps.



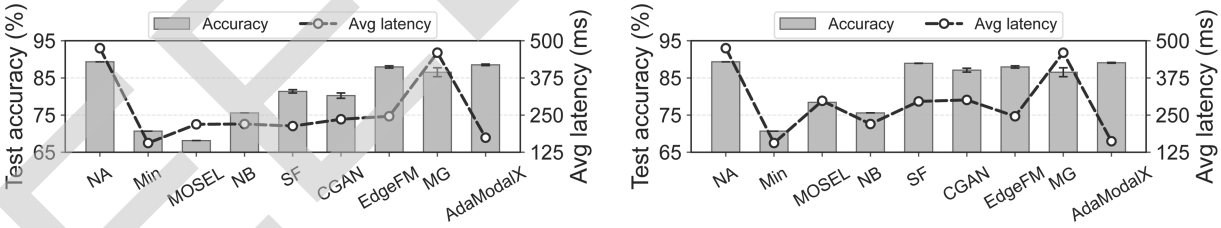
(a) Event detection (T_1), M_1. (b) Driver activity recognition (T_2), M_1.

Fig. 14. Comparison of accuracy *v.s.* offloading rate(%) between importance-aware data uploading at devices (#D1) and six baselines on T_1 and T_2 tasks.



(a) T_1, M_1, 10Mbps. (b) T_2, M_1, 50Mbps. (c) T_3, M_1, 2Mbps.

Fig. 15. Effectiveness of on-demand MFM inference activation at edge (#D2) with the importance-aware data uploading at devices (#D1) disabled.



(a) Event detection (T_1), M_1, arriving rate=25%. (b) Event detection (T_1), M_1, arriving rate=50%.

Fig. 16. Sensitivity to slow-modality arrival rates on T_1 with M_1.

TABLE II
ABLATION STUDY ON DIFFERENT DISTILLATION LOSS COMPONENTS IN THE IR MODALITY

Distillation loss	Model acc. (%)	Overall acc. (%)	Offloading rate (%)
w/ $\mathcal{L}_{PTC}$, w/o $\mathcal{L}_{FA}$, w/o $\mathcal{L}_{CPD}$	63.55 ± 0.56	74.03 ± 0.42	46.67 ± 2.14
w/ $\mathcal{L}_{PTC}$, w/ $\mathcal{L}_{FA}$, w/o $\mathcal{L}_{CPD}$	63.77 ± 0.44	74.39 ± 0.37	44.87 ± 1.86
w/ $\mathcal{L}_{PTC}$, w/o $\mathcal{L}_{FA}$, w/ $\mathcal{L}_{CPD}$	63.23 ± 0.52	74.39 ± 0.41	44.51 ± 2.48
w/ $\mathcal{L}_{PTC}$, w/ $\mathcal{L}_{FA}$, w/ $\mathcal{L}_{CPD}$	67.69 ± 0.36	75.70 ± 0.54	38.61 ± 1.63

TABLE III
OVERALL DEVICE-SIDE EARLY-EXIT OUTCOMES OF AdaModalX ON T_1 WITH M_1

Method	Overall early-exit rate (%)	Overall false early-exit rate (%)	Overall edge rescue rate (%)
AdaModalX	52.62	4.24	2.69

Table II. Compared with the configurations w/ $\mathcal{L}_{PTC}$, w/o $\mathcal{L}_{FA}$, w/o $\mathcal{L}_{CPD}$; w/ $\mathcal{L}_{PTC}$, w/ $\mathcal{L}_{FA}$, w/o $\mathcal{L}_{CPD}$; and w/ $\mathcal{L}_{PTC}$, w/o $\mathcal{L}_{FA}$, w/ $\mathcal{L}_{CPD}$, the full configuration (w/ $\mathcal{L}_{PTC}$, w/ $\mathcal{L}_{FA}$, and w/ $\mathcal{L}_{CPD}$) reduces the offloading rate by 8.06%, 6.26%, and 5.9%, respectively.

2) Benefits of On-Demand MFM Inference Activation At Edge: We evaluate the benefits of on-demand MFM inference

IEEE Proof

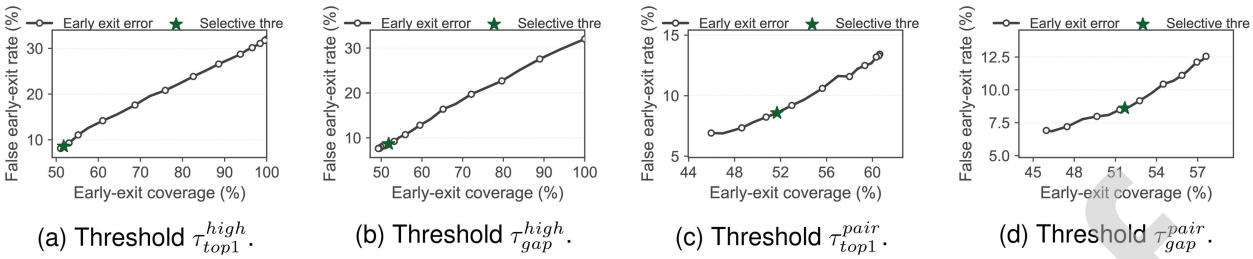


Fig. 17. Reliability of false early-exit rate (FEER) and early-exit coverage under four device-side confidence thresholds on T_1 with M_1.

TABLE IV
ABLATION STUDY ON THE EFFECTS OF PARTIAL MULTIMODAL FEATURE IMPUTATION (I) AND SELECTIVE INFERENCE ACTIVATION (S) ON THE ACCURACY AND LATENCY OF EVENT DETECTION

Method	Event detection	
	Average accuracy (%)	Average latency (ms)
w/o I, w/o S	75.95 ± 0.00	220 ± 5.20
w/ I, w/o S	83.53 ± 0.60	223 ± 0.41
w/o I, w/ S	86.25 ± 0.51	254 ± 6.70
w/ I, w/ S	88.73 ± 0.63	244 ± 2.86

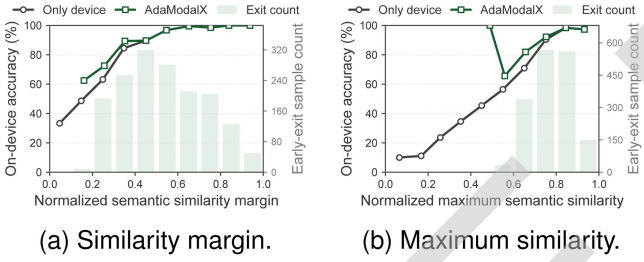


Fig. 18. Reliability of device-side early exit under two confidence scores.

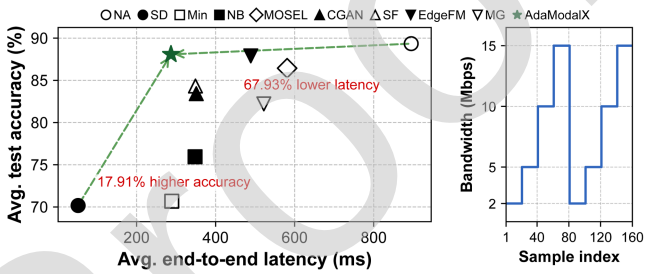


Fig. 19. Comparison of accuracy *v.s.* latency under network fluctuation on T_1 with M_1.

activation at the edge by ablating partial multimodal feature imputation (I) and selective inference activation (S) on T_1 tasks. As in Table IV, compared with w/o I and w/o S, w/ I, w/o S improves accuracy by 7.58%. Enabling inference activation yields a larger accuracy gain of 10.30%. The w/ I and w/ S setting achieves the best accuracy, improving accuracy by 12.78%.

3) Reliability of Device-Side Early Exit: We evaluate the reliability of the device-side confidence-aware early-exit rule on T_1 with M_1. *First*, as shown in Fig. 17, we evaluate the trade-off between the false early-exit rate (FEER) and early-exit coverage, i.e., early-exit rate, by varying each of the four confidence thresholds in the hybrid rule while fixing the other three. Across all four threshold sweeps, FEER generally increases with early-exit coverage, while the selected thresholds achieve a balanced FEER–coverage trade-off. *Second*, as shown in Fig. 18, we group test samples into score intervals to assess the empirical reliability of the two semantic confidence scores. The early-exited samples achieve comparable or higher on-device accuracy in most populated score intervals, indicating that the hybrid rule retains a more reliable subset of device-side predictions. *Third*, Table III quantifies the false early-exit outcomes under the selected threshold setting. AdaModalX early-exits 52.62% of test samples, while only 4.24% of all samples are false early exits; among all samples, 2.69% correspond to false early exits that would have been corrected by the edge model if uploaded. This shows that AdaModalX reduces edge inference while limiting unrecovered local errors.

4) Performance Under Bandwidth Fluctuations: We evaluate AdaModalX on T_1 with M_1 using a sample-sequence bandwidth trace that cycles through $\{2, 5, 10, 15\}$ Mbps every 20 samples. As shown in Fig. 19, AdaModalX achieves the best accuracy-latency trade-off. Compared with NA [39], it reduces the average latency by 67.93% while maintaining competitive accuracy. At comparable latency to Min [62], it improves accuracy by 17.91%. These results demonstrate the robustness of AdaModalX to bandwidth fluctuations in modality transmission delays.

5) Overhead of Proposed Modules: We evaluate the device-side confidence estimators on two smartphones and two embedded devices. As shown in Table V, the estimators require 0.13–3.64 GFLOPs and contain 2.19–12.75 M parameters. Across the four deployments, their mean on-device inference latency ranges from 7.02 to 63.72 ms, with mean local accuracy of 43.67–67.26% and mean offloading rates of 49.05–79.49%. Supporting a new device requires re-distilling its lightweight local estimator and recalibrating its class-wise upload thresholds. This one-time offline adaptation uses 38–50 samples per class and takes, on average, 2.8–11.3 minutes for distillation and 4.65–14.5 seconds for threshold calibration. Table VI further reports the runtime overhead of the edge-side modules on T_1 with M_1. The modality contribution assessor and feature imputation module together require only 6.38 ms and 0.022 GFLOPs, equivalent to 1.38% of the full MFM latency and 0.01% of its computational cost, respectively. Since device-side estimation runs concurrently with data transmission, its latency is not fully added to the end-to-end

IEEE Proof

TABLE V
PER-DEVICE PERFORMANCE AND RESOURCE COSTS OF THE DEVICE-SIDE CONFIDENCE ESTIMATORS

Device	Task	GFLOPs	Params.	Infer. Latency (ms)	Accuracy (%)	Offload. Rate (%)	Distill. Data	Distill. Time (min)	Calib. Time (s)
Vivo X300 Pro	T_1 (Image)	1.82	12.75M	36.10 ± 2.69	67.26 ±1.01	49.05 ±2.07	50-shot, 28 classes	11.3 ± 0.2	14.5 ± 0.04
Jetson Orin Nano	T_3 (RGB vision)	3.64	12.75M	63.72 ± 1.91	43.67 ± 1.75	67.27 ± 2.29	38-shot, 15 classes	7.2 ± 0.1	9.32 ± 0.26
Jetson Orin NX	T_3 (IR vision)	3.64	12.75M	60.89 ± 1.86	55.71 ± 0.64	79.49 ± 2.82	38-shot, 15 classes	5.8 ± 0.1	8.86 ± 0.18
Xiaomi 13 Ultra	T_3 (IMU)	0.13	2.19M	7.02 ± 0.75	52.02 ± 0.45	77.49 ± 0.64	38-shot, 15 classes	2.8 ± 0.1	4.65 ± 0.08

TABLE VI
LATENCY AND COMPUTATIONAL COST OF THE EDGE-SIDE MODULES ON T_1 WITH M_1

Module	Latency (ms)	GFLOPs
Modality contribution assessor	2.59 ± 0.05	0.019
Feature imputation module	3.79 ± 0.11	0.003
Full MFM inference	462 ± 4.05	220

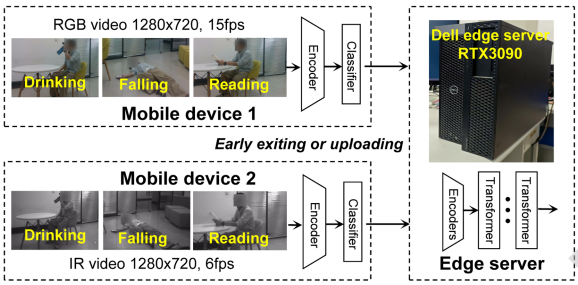


Fig. 20. Illustration of the case study with mobile devices.

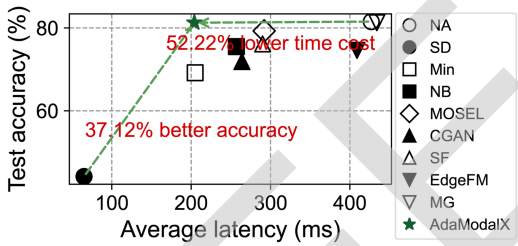


Fig. 21. Comparison of accuracy *v.s.* inference time between AdaModalX and other baselines in the real-world case study.

latency. Thus, the introduced overhead is negligible relative to the end-to-end latency reduction achieved by AdaModalX.

E. Real-World Case Study

We develop a distributed sensing system for smart-home human activity recognition covering 15 daily-life activity classes. This system aims to enhance user experience through intelligent services, as shown in Fig. 20. Specifically, the system integrates a Jetson Orin Nano connected to an RGB Camera (1280x720, 15 FPS), and a Jetson Orin NX connected to an IR Camera (1280x720, 6 FPS). We evaluate the inference accuracy and system latency of AdaModalX, comparing it with nine baselines (NA [39], MG [28], Min [62], NB [21], SD [45], MOSEL [18], CGAN [57], SF [30], and EdgeFM [29]) under 10 Mbps. *First*, as shown in Fig. 21, AdaModalX achieves the best accuracy-latency trade-off in distributed asynchronous multimodal sensing via importance-aware uploading and on-demand activation. Compared with NA [39], it reduces the average latency by

TABLE VII
COMPARISON OF FLOPS ON T_3 WITH M_1

Method	NA	MG	CGAN	EdgeFM	AdaModalX
GFLOPs	335.19	337.01	335.21	280.24	235.26

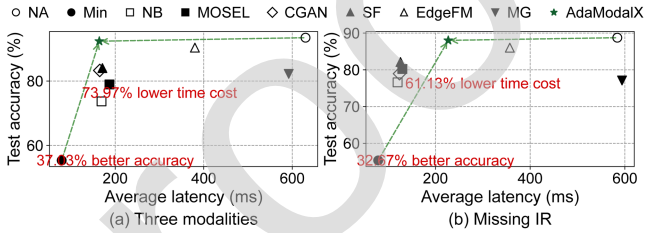


Fig. 22. Comparison of accuracy *v.s.* latency in the three-modality human activity recognition case.

52.22% and it improves the average accuracy by 37.12% compared with Min [62]. *Second*, as shown in Table VII, AdaModalX achieves the lowest computational cost while maintaining competitive accuracy. It reduces the computational cost by 29.81%, 30.19%, 29.82%, and 16.05% compared with NA [39], MG [28], CGAN [57], and EdgeFM [29], respectively.

We further add a Xiaomi 13 Ultra to provide an IMU stream, forming a three-device RGB/IR/IMU sensing configuration under 5 Mbps. *First*, as shown in Fig. 22, AdaModalX achieves the best accuracy-latency trade-off under the complete three-modality setting, reducing average latency by 73.97% over NA [39] and improving accuracy by 37.03% over Min [62]. *Second*, AdaModalX still achieves the best accuracy-latency trade-off when sensor failure makes one modality unavailable and only the remaining two modalities support inference.

VII. RELATED WORK

A. Distributed Multimodal Sensing

Distributed multimodal sensing systems are common in ubiquitous computing, such as autonomous driving [6], action recognition [64], [65], and intelligent healthcare [5]. For example, VIPS [6] proposes an infrastructure-assisted autonomous driving framework that fuses in-vehicle sensor data for improved accuracy. RF-Camera [64] combines video and RFID data to enhance human gesture recognition. Wavoice [66] integrates audio and millimeter-wave signals for precise speech recognition. To enhance accuracy and generalization, researchers are also exploring how MFM can be integrated into distributed sensing systems. CAT [65] explores how an MFM describes audiovisual events in complex environments. Med-MFM [5] enhances clinical decision-making by improving diagnostic efficiency. EarthGPT [37] utilizes visual enhancements and

IEEE Proof

multi-granularity data for multi-sensor remote sensing. However, they mainly focus on the development of MFMs for multimodal data or accurate inference, neglecting the inference latency caused by asynchronous data. In contrast, AdaModalX introduces *importance-aware data uploading at devices* and *on-demand MFM inference activation at servers* to overcome inference blocking from asynchronous data, reducing latency while maintaining accuracy.

B. Asynchronous Multimodal Inference

Existing approaches improve the efficiency and responsiveness of multimodal inference through *modality selection*, *non-blocking multimodal inference*, and *collaborative inference*.

Modality Selection: Modality-selection methods improve inference efficiency through *policy networks* (*e.g.*, AdaMML [28]), *modality gating* (*e.g.*, SMG [61]), or *latency-constrained selection* (*e.g.*, MOSEL [18]). However, these methods make selection decisions within a request or segment without considering how modality availability changes during device–edge transmission. They therefore do not directly address distributed mobile sensing, where modality data arrive asynchronously and become available progressively. In contrast, AdaModalX progressively updates the inference candidate as modality data arrive.

Non-blocking Multimodal Inference: It avoids waiting for all modalities by *exploiting cross-modal features* [25] or *imputing missing data* (*e.g.*, AdaFlow [57], AdaFlowLite [31], and PATCH [21]). However, these methods mainly focus on using available inputs or completing unavailable modality data, without determining when to activate the edge-side MFM according to candidate utility as modalities progressively arrive. AdaModalX activates edge-side MFM inference when the candidate formed by fully arrived modalities provides sufficient utility; otherwise, it imputes features from partially received inputs and re-evaluates the augmented candidate, avoiding unnecessary imputation and premature inference.

Collaborative Inference: Collaborative-inference methods coordinate mobile devices and edge servers through *model partitioning* [67], *data compression* [68], or *confidence-based big–small model switching* (*e.g.*, big/little [63], AppealNet [56], and EdgeFM [29]). However, *model partitioning* tightly couples device–edge execution and is bandwidth-sensitive, whereas *data compression* incurs processing overhead but still requires waiting for delayed modalities before inference. *Big–small model switching* relies on device-side confidence estimates; inaccurate estimates may cause false early exits or unnecessary offloading. AdaModalX uses joint confidence rules for device-side early exit and separately determines when to activate the edge-side MFM according to candidate utility as modality data progressively arrive.

Overall, AdaModalX coordinates two runtime decisions for asynchronous multimodal inference: whether to exit early based on device-side confidence and, otherwise, when to activate the edge-side MFM according to candidate utility as modality data progressively arrive.

VIII. Conclusion

In this paper, we address the challenge of modality asynchrony in MFM-based distributed multimodal sensing. Unlike prior methods that require extensive retraining or architectural modifications, we introduce AdaModalX, a progressive multimodal inference framework that decouples inference execution from modality synchronization via runtime utility-aware orchestration. By learning modality utility offline and dynamically triggering inference only when sufficient data is available, AdaModalX achieves a balance between accuracy and latency. Additionally, AdaModalX leverages lightweight models on mobile devices to reduce data transmission and processing delays, further improving responsiveness. Evaluations across four mobile tasks and real-world scenarios on four mobile devices demonstrate that AdaModalX achieves an accuracy improvement of up to 26.97% and a system latency reduction of up to 64.75%. Our work demonstrates the potential of AdaModalX in asynchronous distributed environments, although large-scale evaluations primarily emulate asynchronous modality arrivals under controlled network conditions. Future work will extend AdaModalX to more architectures, modalities, and devices and validate it in larger live deployments under diverse sensing and network conditions.

References

[1] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," *Proc. ACM InterAct. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–22, 2018.

[2] S. Liu et al., "AdaEnlight: Energy-aware low-light video stream enhancement on mobile devices," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 4, pp. 1–26, 2023.

[3] Z.-M. Wang, M.-H. Rao, S.-H. Ye, W.-T. Song, and F. Lu, "Towards spatial computing: Recent advances in multimodal natural interaction for extended reality headsets," *Front. Comput. Sci.*, vol. 19, no. 12, 2025, Art. no. 1912708.

[4] G. Cong, "Toward city foundation models: From task-specific spatial approaches to urban general intelligence," *Front. Comput. Sci.*, vol. 20, no. 12, 2026, Art. no. 2012628.

[5] F. Liu et al., "A medical multimodal large language model for future pandemics," *NPJ Digit. Med.*, vol. 6, no. 1, 2023, Art. no. 226.

[6] S. Shi et al., "VIPS: Real-time perception fusion for infrastructure-assisted autonomous driving," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw.*, 2022, pp. 133–146.

[7] L. Wang et al., "AdaEvo: Edge-assisted continuous and timely DNN model evolution for mobile devices," *IEEE Trans. Mobile Comput.*, vol. 24, no. 4, pp. 2485–2503, Apr. 2025.

[8] D. Zhang et al., "MM-LLMs: Recent advances in multimodal large language models," in *Proc. Findings Assoc. Comput. Linguist.: ACL 2024*, 2024, pp. 12401–12430, *arXiv:2401.13601*.

[9] S. Jiang et al., "Edge large language models: A comprehensive survey," *CCF Trans. Pervasive Comput. Interaction*, vol. 8, pp. 181–210, 2026.

[10] H. Wei et al., "A survey of foundation models for IoT: Taxonomy and criteria-based analysis," *CCF Trans. Pervasive Comput. Interaction*, vol. 8, no. 1, pp. 1–29, 2026.

[11] Y. He, R. Cheng, G. Balasubramaniam, Y.-H. H. Tsai, and H. Zhao, "Efficient modality selection in multimodal learning," *J. Mach. Learn. Res.*, vol. 25, no. 47, pp. 1–39, 2024.

[12] J. Han et al., "OneLLM: One framework to align all modalities with language," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 26574–26585.

[13] Mobileye, "Mobileye official website–smart cabin and driver assistance solutions," Accessed: Sep. 1, 2025. [Online]. Available: https://www.mobileye.com/

[14] Inc. Tesla, "Tesla model y Owner's manual," Accessed: Sep. 1, 2025. [Online]. Available: https://www.tesla.com/ownersmanual/modely/en_us/

IEEE Proof

[15] S. Dai et al., "Babel: A scalable pre-trained model for multi-modal sensing via expandable modality alignment," in *Proc. 23 rd ACM Conf. Embedded Netw. Sensor Syst.*, 2025, pp. 240–253.

[16] Amazon, "Introducing Alexa, The next generation of Alexa," Accessed: Feb. 26, 2025. [Online]. Available: https://www.aboutamazon.com/news/devices/new-alexa-generative-artificial-intelligence

[17] Google, "Project Astra," 2023. [Online]. Available: https://deepmind.google/technologies/gemini/project-astra/

[18] B. Hu, L. Xu, J. Moon, N. J. Yadwadkar, and A. Akella, "MOSEL: Inference serving using dynamic modality selection," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2024, pp. 8872–8886.

[19] S. Liu et al., "DeepSwarm: Towards swarm deep learning with bi-directional optimization of data acquisition and processing," *Front. Comput. Sci.*, vol. 19, no. 3, 2025, Art. no. 193501.

[20] T. Li, J. Huang, E. Risinger, and D. Ganesan, "Low-latency speculative inference on distributed multi-modal data streams," in *Proc. 19th Annu. Int. Conf. Mobile Syst., Appl., Serv.*, 2021, pp. 67–80.

[21] J. Wang et al., "PATCH: A plug-in framework of non-blocking inference for distributed multimodal system," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 7, no. 3, pp. 1–24, 2023.

[22] C. Chen and P. Zhang, "Modality-collaborative transformer with hybrid feature reconstruction for robust emotion recognition," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 20, no. 5, pp. 1–23, 2024.

[23] Y.-L. Lee, Y.-H. Tsai, W.-C. Chiu, and C.-Y. Lee, "Multimodal prompting with missing modalities for visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 14943–14952.

[24] Z. Lian, L. Chen, L. Sun, B. Liu, and J. Tao, "GCNet: Graph completion network for incomplete multimodal learning in conversation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8419–8432, Jul. 2023.

[25] J. Chen and A. Zhang, "HGMF: Heterogeneous graph-based fusion for multimodal data with incompleteness," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1295–1305.

[26] S. Liu et al., "CrowdHMTware: A cross-level co-adaptation middleware for context-aware mobile DL deployment," *IEEE Trans. Mobile Comput.*, vol. 24, no. 8, pp. 7615–7631, Aug. 2025.

[27] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, "PandaGPT: One model to instruction-follow them all," in *Proc. 1st Workshop Taming Large Lang. Model.*, Controllability in the Era of Interactive Assistants, 2023, pp. 11–23.

[28] R. Panda et al., "AdaMML: Adaptive multi-modal learning for efficient video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7556–7565.

[29] B. Yang et al., "EdgeFM: Leveraging foundation model for open-set learning on the edge," in *Proc. 21st ACM Conf. Embedded Networked Sensor Syst.*, 2023, pp. 111–124.

[30] H. Wang, Y. Chen, C. Ma, J. Avery, L. Hull, and G. Carneiro, "Multi-modal learning with missing modality via shared-specific feature modelling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 15878–15887.

[31] S. Liu et al., "AdaFlowLite: Scalable and non-blocking inference on asynchronous mobile data," *IEEE Trans. Mobile Comput.*, vol. 24, no. 11, pp. 12206–12220, Nov. 2025.

[32] Y. Chen, Y. Bai, R. Mitev, K. Wang, A.-R. Sadeghi, and W. Xu, "FakeWake: Understanding and mitigating fake wake-up words of voice assistants," in *Proc. 2021 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 1861–1883.

[33] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, "Audio-visual event localization in unconstrained videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 247–263.

[34] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, "MOSI: Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos," in *IEEE Intell. Syst.*, vol. 31, no. 6, pp. 82–88, 2016.

[35] M. Martin et al., "Drive&Act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2801–2810.

[36] X. Li, S. Liu, Z. Zhou, Y. Xu, B. Guo, and Z. Yu, "ClassTer: Mobile shift-robust personalized federated learning via class-wise clustering," *IEEE Trans. Mobile Comput.*, vol. 24, no. 3, pp. 2014–2028, Mar. 2025.

[37] W. Zhang, M. Cai, T. Zhang, Y. Zhuang, and X. Mao, "EarthGPT: A universal multi-modal large language model for multi-sensor image comprehension in remote sensing domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5917820.

[38] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3," in *Proc. Deep Learn. Inside Out 3rd Workshop Knowl. Extraction Integration Deep Learn. Architectures*, pp. 100–114, 2022.

[39] R. Girdhar et al., "ImageBind: One embedding space to bind them all," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 15180–15190.

[40] S. Liu, B. Guo, K. Ma, Z. Yu, and J. Du, "AdaSpring: Context-adaptive and runtime-evolutionary deep model compression for mobile applications," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–22, 2021.

[41] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7436–7456, Nov. 2022.

[42] S. Liu et al., "Enabling resource-efficient AIoT system with cross-level optimization: A survey," *IEEE Commun. Surv. Tut.*, vol. 26, no. 1, pp. 389–427, Firstquarter 2024.

[43] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[44] D. Luo, Y. Zou, and D. Huang, "Investigation on joint representation learning for robust feature extraction in speech emotion recognition," in *Proc. Interspeech*, 2018, pp. 152–156.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[46] A. G. Howard, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[47] S. Liu et al., "AdaDeep: A usage-driven, automated deep model compression framework for enabling ubiquitous intelligent mobiles," *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3282–3297, Dec. 2021.

[48] M. Rahimpour et al., "Cross-modal distillation to improve MRI-based brain tumor segmentation with missing MRI sequences," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 7, pp. 2153–2164, Jul. 2022.

[49] H. Wang et al., "Learnable cross-modal knowledge distillation for multi-modal learning with missing modality," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Interv.*, 2023, pp. 216–226.

[50] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[51] M. Li et al., "A unified self-distillation framework for multimodal sentiment analysis with uncertain missing modalities," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 9, pp. 10074–10082.

[52] F. Meyniel, M. Sigman, and Z. F. Mainen, "Confidence as Bayesian probability: From neural origins to behavior," *Neuron*, vol. 88, no. 1, pp. 78–92, 2015.

[53] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21464–21475.

[54] S. Hernández, D. Vergara, M. Valdenegro-Toro, and F. Jorquera, "Improving predictive uncertainty estimation using dropout–hamiltonian monte carlo," *Soft Comput.*, vol. 24, no. 6, pp. 4307–4322, 2020.

[55] H. Chen, W. Lu, and R. Song, "Statistical inference for online decision making via stochastic gradient descent," *J. Amer. Stat. Assoc.*, vol. 116, no. 534, pp. 708–719, 2021.

[56] M. Li, Y. Li, Y. Tian, L. Jiang, and Q. Xu, "AppealNet: An efficient and highly-accurate edge/cloud collaborative architecture for DNN inference," in *Proc. 58th ACM/IEEE Des. Automat. Conf.*, 2021, pp. 409–414.

[57] F. Wu et al., "AdaFlow: Opportunistic inference on asynchronous mobile data with generalized affinity control," in *Proc. 22nd ACM Conf. Embedded Netw. Sensor Syst.*, 2024, pp. 606–618.

[58] Qwen Team, "Qwen3.5: Accelerating productivity with native multimodal agents," Feb. 2026, Accessed: Apr. 15, 2026. [Online]. Available: https://qwen.ai/blog?id=qwen3.5

[59] W. Kayet al., "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[60] Z. Xue and R. Marculescu, "Dynamic multimodal fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2023, pp. 2575–2584.

[61] X. Hou et al., "SMG: A system-level modality gating facility for fast and energy-efficient multimodal computing," in *Proc. 2023 IEEE Real-Time Syst. Symp.*, 2023, pp. 291–303.

[62] X. Hou et al., "Architecting efficient multi-modal AIoT systems," in *Proc. 50th Annu. Int. Symp. Comput. Architecture*, 2023, pp. 1–13.

IEEE Proof

[63] E. Park et al., "Big/little deep neural network for ultra low power inference," in *Proc. 2015 Int. Conf. Hardware/Softw. Codesign Syst. Synth.*, 2015, p. 124–132.
[64] X. Liu, D. Liu, J. Zhang, T. Gu, and K. Li, "RFID and camera fusion for recognition of human-object interactions," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, 2021, pp. 296–308.
[65] Q. Ye, Z. Yu, R. Shao, X. Xie, P. Torr, and X. Cao, "CAT: Enhancing multimodal large language model to answer questions in dynamic audio-visual scenarios," in *Comput. Vis., Lecture Notes Comput. Sci.*, vol. 15068, Cham, Switzerland: Springer, pp. 146–164, 2025.
[66] T. Liu et al., "Wavoice: A noise-resistant multi-modal speech recognition system fusing mmWave and audio signals," in *Proc. 19th ACM Conf. Embedded Networked Sensor Syst.*, 2021, pp. 97–110.
[67] S. Liu et al., "AdaKnife: Flexible DNN offloading for inference acceleration on heterogeneous mobile devices," *IEEE Trans. Mobile Comput.*, vol. 24, no. 2, pp. 736–748, Feb. 2025.
[68] K. Huang and W. Gao, "Real-time neural network inference on extremely weak devices: Agile offloading with explainable AI," in *Proc. 28th Annu. Int. Conf. Mobile Comput. And Netw.*, 2022, pp. 200–213.



Xiangrui Xu received BE degree from the Harbin Institute of Technology, in 2023. He is currently working toward the Doctoral degree with Northwestern Polytechnical University, China. His research interests include resource-efficient mobile deep learning and mobile systems.



Sicong Liu (Member, IEEE) received the PhD degree from the School of Computer Science and Technology, Xidian University, in 2020. She is currently an associate professor with the School of Computer Science, Northwestern Polytechnical University. Her research interests include mobile computing and resource-efficient mobile deep learning. She was a TPC member of MobiSys 2021 and BigCom 2021.



Zhiwen Yu (Senior Member, IEEE) received the PhD degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2005. He was an Alexander Von Humboldt fellow with Mannheim University, Germany, and a research fellow with Kyoto University, Kyoto, Japan. He is currently the vice president with Harbin Engineering University and also a professor with the School of Computer Science, Northwestern Polytechnical University. His research interests include ubiquitous computing, HCI, and mobile sensing and computing.



Zimu Zhou (Member, IEEE) received the BE degree from Tsinghua University, in 2011, and the PhD degree from the Hong Kong University of Science and Technology, in 2016. He is currently an assistant professor with the Department of Data Science, City University of Hong Kong. His research interests include mobile and ubiquitous computing, and focusing on AIoT.



Bin Guo (Senior Member, IEEE) received the PhD degree in computer science from Keio University, Japan. He was a postdoc researcher with Institut Telecom SudParis, France. He is currently a professor with Northwestern Polytechnical University, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI. He was as an associate editor for *IEEE Communications Magazine* and *IEEE Transactions on Human-Machine-Systems*, and the guest editor of *ACM Transactions on Intelligent Systems*.

IEEE Proof