Mega-CE^2: A Multimodal Heterogeneous Aggregation Framework for End-Edge-Cloud Computing

Zinuo Cheng CHENGZI@MAIL.BNU.EDU.CN
Beijing Normal University

Haodi Wang HAODI.WANG@CITYU.EDU.HK
City University of Hong Kong

Rongfang Bie RFBIE@BNU.EDU.CN
Beijing Normal University

Editors: Hung-yi Lee and Tongliang Liu

Abstract

End-Edge-Cloud Computing (EECC) has emerged as the mainstream computing paradigm, integrating edge computing to overcome the limitations of traditional federated learning in communication efficiency and resource scheduling. However, existing studies reveal that most frameworks still struggle with challenges such as computing resource allocation and high end-to-end latency in EECC. To address these issues, we propose Mega-CE^2, a novel multi-modal heterogeneous aggregation framework. Mega-CE^2 establishes a closed-loop feedback mechanism from the bottom-up to the top down through end-device data serialization, edge-server model personalization, and cloud-based optimization. Notably Mega-CE^2 incorporates lightweight adapters for fine-tuning, enabling efficient deployment while preserving local model personalization. These adapters, with fewer parameters than the global model, optimize model parameters during edge-to-cloud aggregation, thereby achieving both lightweight and personalized capabilities. In experiments on two open-source standard datasets, we show that the performance of Mega-CE^2 improves by 3%–5%, while maintaining scalability with lightweight and low-latency characteristics.

Keywords: End-Edge-Cloud Computing , Multimodal , Aggregation , Resource Scheduling , Efficient

1. Introduction

Cloud computing has dominated computing paradigms for a decade, delivering on-demand computational power that accelerated AI advancements Xiong et al. (2019). However, exponential data growth in the IoT era challenges cloud-centric approaches, so sole reliance on clouds fails to meet stringent latency/accuracy demands of real-time applications like autonomous driving Chang et al. (2021). While federated learning avoids centralized bottlenecks, its cloud-client architecture suffers from inefficient communication and resource scheduling in large-scale distributed training. This imposes excessive computational loads on cloud servers and high communication overhead. Client heterogeneity and dynamic networks further complicate training, potentially degrading final performance. Thus, End-Edge-Cloud Computing (EECC) Duan et al. (2022) has emerged as a mainstream solution. As Figure 1 shows, its core innovation introduces edge devices between clouds and clients, creating a three-tier cloud-edge-end architecture that reduces cloud computational pressure, optimizes communication resources, and minimizes system workload Ren et al. (2019).

However, existing multimodal heterogeneous aggregation framework for EECC face several challenges. Firstly, conventional distributed learning enforces uniform model architectures across nodes, creating bottleneck effect that constrain training scalability and underutilize edge/cloud resources. Secondly, the end-layer data and modality heterogeneity challenges effective multimodal integration in EECC environments. Finally, existing approaches either achieve high accuracy through complex, parameter-heavy architectures with excessive computational overhead, or favor speed via oversimplified designs with limited scalability. Addressing these challenges requires developing a more efficient and balanced multimodal heterogeneous aggregation framework that can reassemble complex multimodal tasks while achieving coordinated collaboration across different architectural layers. In this paper, we introduce a new multimodal heterogeneous aggregation framework optimized for cloud-edge-end environments as Figure 2 shows. Our goal is to enhance the real-time performance and accuracy of compute-intensive tasks like autonomous driving by effectively coordinating layer interactions and efficiently decomposing and aggregating multimodal tasks. First, the end processes collected multimodal data (text, images, audio) to generate location-aware feature sequences, synchronously aggregating them to the edge layer to reduce transmission delay and network costs. The edge layer then employs cross-modal attention modules and lightweight self-attention Transformer models to capture multimodal dependencies, optimizing its parameters before asynchronously sending the data to the cloud. Finally, the cloud aggregates edge model information to optimize global parameters, and sends them back to the edge layer, enabling it to fine-tune and promote local personalized models, thus completing the closed-loop feedback mechanism from the bottom-up to the top down.

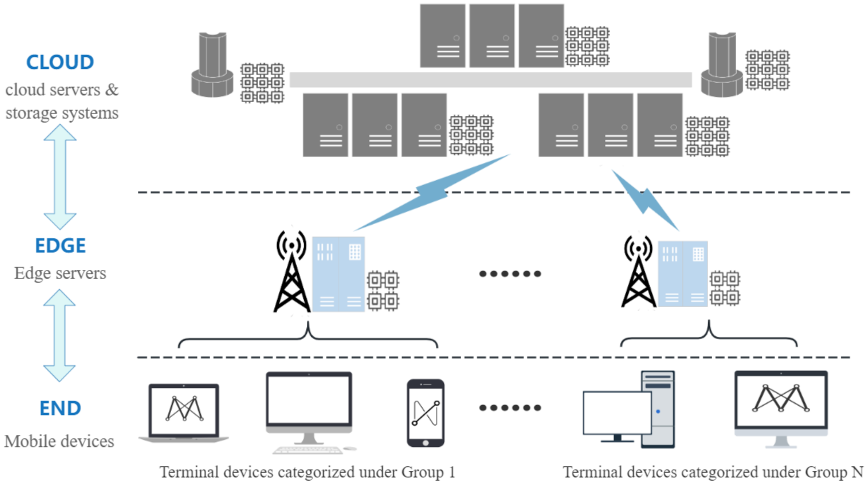
CLOUD
cloud servers & storage systems
EDGE
Edge servers
END
Mobile devices
Terminal devices categorized under Group 1
Terminal devices categorized under Group N

Figure 1: Framework Diagram of the EECC

Compared with existing multi-modal heterogeneous aggregation frameworks, our proposed Mega-CE^2 supports multi-level collaboration, achieving high speed while maintaining high accuracy. This allows it to better adapt to the fast-paced environments of numerous intelligent application scenarios (e.g., smart education and intelligent transportation) and meet the demands for real-time performance and accuracy.

We summarize our contribution as follows:

• We propose a novel multi-modal heterogeneous aggregation framework Mega-CE^2, which integrates text, image, and audio data across EECC and allocates complex multi-modal tasks reasonably across the three tiers. learning.

• We introduce lightweight adapters for fine-tuning to optimize model parameters during edge-to-cloud aggregation, which have fewer parameters than the global model, enabling efficient deployment while preserving local edge model personalization.

• After experiments on three open-source multi-modal sentiment analysis datasets, we found that our proposed framework maintains high accuracy while improving efficiency.

2. Related Work

In deep learning, while increasing model size with abundant training data enhances accuracy and generalization Hu et al. (2021), existing federated learning methods Li et al. (2020); Karimireddy et al. (2020) are constrained by homogeneous model requirements across central servers and resource-limited devices, preventing training of larger models. Although partial training Nguyen et al. (2023) approaches that partition global models into distributed submodels (e.g., Nebula Zhuang et al. (2024)) enable server-side model scaling beyond client limitations, the dense parameter coupling in neural networks poses significant challenges Ashish (2017); He et al. (2016)for creating compact, personalized submodels - challenges exacerbated by dynamic environmental adaptations and computationally intensive edge requirements Fang et al. (2018).The methods based on knowledge distillation Wu et al. (2024b); Cho et al. (2022) utilize the model outputs from client devices as regularization terms for the training of the server-side model, thereby facilitating the transfer of comprehensive representations learned by clients on decentralized private data. But such methods remain limited to two-layer architectures, lacking support for end-edge-cloud collaborative training. For three-layer architectures, current verification frameworks like BSBODP protocol Wu et al. (2024a) typically employ single-modal datasets,text or images, failing to address multimodal data processing challenges in cloud-edge-end environments.

For the three-tier cloud-edge-end environment,Wu et al. (2024a) proposed a novel aggregated federated learning framework which organizes computations through a customized Bridging Sample-Based Online Distillation Protocol (BSBODP) to collaboratively train models, where the largest model (i.e., the cloud model) ultimately serves as the target model. Although this framework supports heterogeneous model architectures across different computational tiers, its validation and evaluation phases typically utilize datasets containing only unimodal data (e.g., text or images). This approach overlooks multimodal data fusion and training, failing to adequately address effective multimodal data processing in cloud-edge-end architectures.

The demand for real-time analysis and processing of multi-modal data has gradually permeated various aspects of daily life, such as autonomous driving and intelligent education. These fields necessitate cloud-edge-end collaboration and multimodal data fusion. While Jia et al. (2024) proposes modality-specific feature extraction with weighted fusion, it overlooks low-level semantic cross-modal correlations and dynamic adaptability. Wang et al. (2023) developed a Transformer-based multimodal encoder-decoder that converts non-linguistic features to natural language through modality-enhanced cross-attention, whereas Dumpala et al. (2019) employed cross-modal auto-encoders for audiovisual alignment via latent space consistency constraints, albeit requiring aligned supervision signals. Crucially, these centralized architectures cannot accommodate distributed multi-level collaboration demands, particularly between edge device real-time processing and cloud-based global optimization.

The cloud-edge-end computing paradigm has revolutionized intelligent applications like education and transportation by enabling collaborative task processing. When end nodes cannot adequately analyze complex multimodal tasks Hou and Zhang (2021), edge nodes provide intermediate processing before resorting to cloud resources, thereby optimizing service quality. While edge nodes handle collaborative computations, the cloud primarily aggregates edge-layer model parameters for training updates, manages data storage, and assists edges in processing particularly demanding tasks.

3. Proposed Methods

In this section, we will first briefly introduce some preliminary knowledge about cross-modal transformers, and then elaborate on the framework and implementation of our Mega-CE^2 in detail.

3.1. Preliminaries

3.1.1. Cross-Modal Transformers

The model we use Tsai et al. (2019) utilizes stacked bidirectional cross-modal attention blocks to process feature sequences. Unlike encoder-decoder architectures Shaw et al. (2018), that perform explicit modality translation, Cross-modal attention enables direct model focus on salient signal regions across modalities, thereby strengthening inter-modal relationship comprehension Ashish (2017).Now, considering two modalities α and β , which can refer to text, audio, or video, $X_\alpha \in \mathbb{R}^{T_\alpha \times d_\alpha}$ and $X_\beta \in \mathbb{R}^{T_\beta \times d_\beta}$, where $T(\cdot)$ and $d(\cdot)$ represent the sequence length and feature dimension. We define the query Q as $Q_\alpha = X_\alpha W_{Q_\alpha}$, the key K as $K_\beta = X_\beta W_{K_\beta}$, the value V as $V_\beta = X_\beta W_{V_\beta}$, where $W_{Q_\alpha} \in \mathbb{R}^{d_\alpha \times d_k}$, $W_{K_\beta} \in \mathbb{R}^{d_\beta \times d_k}$, $W_{V_\beta} \in \mathbb{R}^{d_\beta \times d_v}$ are weights. Through different weights, the inputs of different modalities are projected to unified query, key, and value spaces for subsequent calculations. Based on this, The potential adaptation fromβ to α can be represented as $Y_\alpha := CM_{\beta\rightarrow\alpha}(X_\alpha, X_\beta) \in \mathbb{R}^{T_\alpha \times d_v}$,

$$\begin{aligned} Y_\alpha =& CM_{\beta\rightarrow\alpha}(X_\alpha, X_\beta) \\ =& softmax\left(\frac{Q_\alpha K_\beta^T}{\sqrt{d_k}}\right) V_\beta \\ =& softmax\left(\frac{X_\alpha W_{Q_\alpha} W_{K_\beta}^T X_\beta^T}{\sqrt{d_k}}\right) X_\beta W_{V_\beta} \end{aligned} \tag{1}$$

The equation involves Q_α, K_β and V_β, where the scaling factor $\sqrt{d_k}$ prevents the dot product from becoming too large. After that, the score matrix is row-normalized using softmax, and the final softmax$(\cdot) \in \mathbb{R}^{T_\alpha \times T_\beta}$ has its (i, j)-the element representing the attention weight between the i-the position in α and the j-th position in β.We refer to the equation as single-head cross-modal attention.

Within a single cross-modal attention block, low-level input features first undergo cross-modal attention computation to establish inter-modal semantic relationships. Residual connections then combine the original input with cross-modal interaction results, followed by nonlinear transformation through position-wise feed-forward sublayers.Taking visual (V) to language (L) information transfer (denoted as V→L) as an example, where d represents the fixed dimensionality of each cross-modal attention block, the cross-modal Transformer performs feed-forward computation for layers $i = 1, 2, ..., D$:

$$\begin{aligned} Z^{[0]}_{V\to L} &= Z^{[0]}_{L}, \\ \hat{Z}^{[i]}_{V\to L} &= \mathrm{CM}^{[i],\mathrm{mul}}_{V\to L}(\mathrm{LN}(Z^{[i-1]}_{V\to L}), \mathrm{LN}(Z^{[0]}_{V})) + \mathrm{LN}(Z^{[i-1]}_{V\to L}), \\ Z^{[i]}_{V\to L} &= f_{\theta^{[i]}_{V\to L}}(\mathrm{LN}(\hat{Z}^{[i]}_{V\to L})) + \mathrm{LN}(\hat{Z}^{[i]}_{V\to L}), \end{aligned} \tag{2}$$

where CM($\cdot$) denotes cross-modal operation, LN represents layer normalization Ba et al. (2016), f_θ is the position-wise feed-forward sublayer parameterized by θ, and mul indicates multi-head attention.

In this process, each modality updates its sequence by incorporating low-level external information from multi-head cross-modal attention blocks, where source modality signals are transformed into distinct key-value pairs for target modality interaction, ultimately yielding a model based on pairwise cross-modal interaction modeling.

3.1.2. Self-Attention And Prediction

The self-attention's core computation can be expressed as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3}$$

where the query matrix Q, key matrix K, and value matrix V are all derived from linear projections of the input sequence $X \in \mathbb{R}^{T\times d}$. The scaling factor $\sqrt{d_k}$ prevents excessively large matrix values from the dot product that could lead to gradient vanishing. We concatenate outputs sharing the same target modality to generate $Z_{L,V,A} \in \mathbb{R}^{T_{T,V,A}\times 2d}$, e.g., $Z_L = [Z^{[D]}_{V\to L}; Z^{[D]}_{A\to L}]$, where $[\cdot;\cdot]$ denotes feature-wise concatenation. In the prediction stage, the Transformer sequence model based on self-attention mechanism (Ashish, 2017) encodes the temporally fused cross-modal features. The model extracts feature vectors from the last k timesteps of the output, followed by nonlinear mapping and dimensionality reduction through a fully-connected layer.

3.2. Overall Framework Of Mega-CE^2

An overview of our method is shown in Figure 2, which divides and aggregates complex multimodal tasks in EECC, enabling EECC collaborative processing of language, audio, and video modalities across different tiers. Next, in this section, we will specifically introduce the main tasks that each layer is responsible for after the division, while the aggregation process between layers will be detailed in Sections 3.3 and 3.4.

At the end layer, each device possesses multimodal data acquisition and preprocessing capabilities. The preprocessed multimodal data is then batched and serialized, with positional encoding information injected to generate low-level position-aware feature sequences

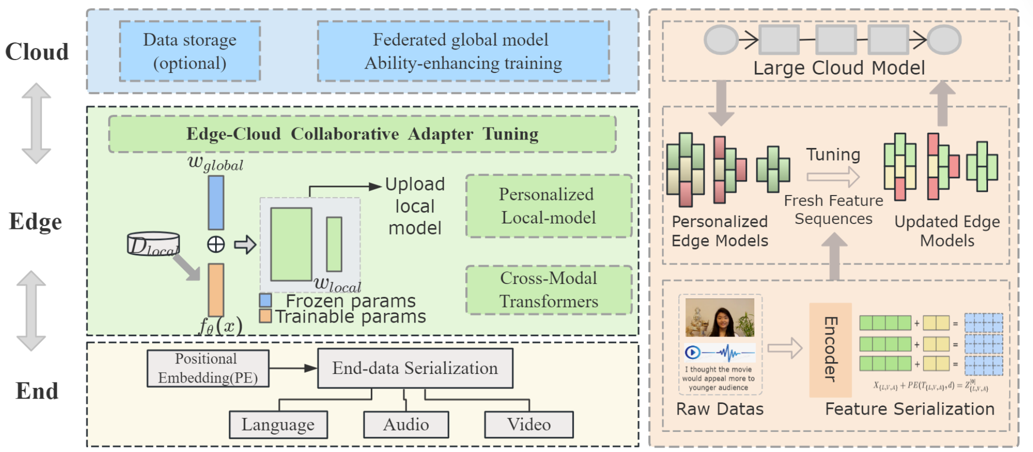

Figure 2: An overview of Mega-CE^2. Our Mega-CE^2 consists of three tiers: (1) end-device data serialization, (2) edge-server model personalization, (3) cloud-based optimization.

$Z^{[0]}_{\{L,V,A\}}$. End devices can provide real-time responses based on their local models while simultaneously collecting new data to support iterative model optimization. For data of different modalities, different serialization methods are adopted. Language undergoes tokenization and stop-word removal, audio data requires noise reduction and sample rate adjustment while image data needs scaling and normalization.

Each edge node is equipped with a lightweight model comprising multiple cross-modal attention blocks ,referring to Equation (2) and self-attention Transformers. This model can directly return the prediction results to the end device, while transmitting intermediate results (such as model parameters) to the cloud server. At the same time, it accepts cloud model information to achieve edge-cloud collaboration for adapter fine-tuning. The specific process is detailed in Section 3.4. Since edge nodes are geographically closer to data sources than cloud data centers, edge processing effectively reduces service latency and provides auxiliary computational resources for compute-intensive intelligent applications.

The cloud layer is equipped with substantial computational capabilities and abundant storage resources, enabling the training of higher-accuracy AI models. It asynchronously aggregates information from edge models to build deeper and more complex models, and the specific aggregation methods will be detailed in Section 3.3. The cloud ensures that edge sub-models maintain both personalization and lightweight characteristics through parameter distribution and fine-tuning mechanisms.

3.3. Device Aggregation Methods In Mega-CE^2

Our Mega-CE^2 method employs synchronous aggregation between end-edge devices and asynchronous aggregation between edge-cloud devices. When an idle edge device participates in model training, it requests the global model from the cloud. Upon receiving the request, the cloud immediately checks the current state of the global model and transmits the result to the corresponding edge device. After receiving the global model, the edge device promptly broadcasts it to its subordinate end devices. The end devices then train their local models using their private datasets and transmit the model information back to the edge device in a synchronous manner, participating in synchronous aggregation to derive a new edge model. During the training process, once the cloud server receives a model update from any edge device, it directly performs aggregation and updates the global model without waiting for other devices to complete their updates. This design reduces the waiting time, significantly improving training efficiency.

To further aggregate edge-side model information to the cloud timely, the cloud server employs a weighted aggregation approach on the received model updates, as shown in Equation (4), ensuring the accuracy and stability of the global model. This approach computes global model parameters by weighting locally-trained edge models rather than raw data, where edges with larger datasets receive higher weights. After completing cloud-edge aggregation, each edge device uploads its model to the cloud server.

$$w_{\text{avg}}[k] = \sum_{i=1}^{N} \alpha_i w_i[k], \quad \alpha_i = \frac{n_i}{\sum_{j=1}^{N} n_j}, \tag{4}$$

where $w_{\text{avg}}[k]$ denotes the k-th parameter of the aggregated global model, α_i represents the normalized data volume ratio as the edge weight, $w_i[k]$ is the k-th parameter of the i-th edge model, and n_i indicates the local dataset size of edge i. Only model parameters w_i are transmitted without sharing raw data n_i.

As for the edge and the end, during device initialization, edge nodes register capability profiles (including hardware configuration, available sensors, and computational resources) with edge servers, which respond with configuration packets specifying data acquisition parameters and transmission standards. Subsequently, in periodic data transmission phase, end devices package multimodal data into standardized packets—containing metadata like timestamps, device IDs, and quality indicators—while employing adaptive compression Kim et al. (2020) to significantly reduce transmission overhead.

3.4. Adapter Tuning Method Between Edge And Cloud

After the cloud model completes its training, it distributes the parameters to the edge models. The edge models adopt a lightweight design, utilizing local adapters for fine-tuning to preserve their personalized characteristics. Adapter-based fine-tuning involves inserting small, trainable modules between the layers of a pre-trained model while freezing the original parameters, allowing only the adapters to be trained. The adapters employed in this paper consist solely of attention layers and linear layers, with a parameter count significantly smaller than that of the global model. Through multi-head self-attention mechanisms, the adapters dynamically adjust the weight distribution of global features, capturing contextual dependencies in local data.

Specific process as in the equation 5 :

$$\begin{aligned} w_{\text{local}} &= \text{FineTune}(w_g, \mathcal{D}_{\text{local}}) \\ &= w_g \oplus \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{local}}} \mathcal{L}\big(w_g(x) + f_\theta(x), y\big) \end{aligned} \tag{5}$$

where f_θ contains only attention & linear layers, $\theta \sim \mathcal{N}(0, 0.02^2)$ initialized.

The adapter parameters are trained only locally and don't participate in transmission, which avoids global model oscillation caused by heterogeneous information at the edge devices, while maintaining the local personalized models at the edge.

4. Experiments and Analyses

4.1. Experimental Details

The cloud tier is deployed on high-performance server nodes equipped with six NVIDIA GeForce RTX 3090 GPUs, leveraging `torch2.4.1+cu124` for optimized model aggregation. Notably, although real-world endpoints should include data acquisition capabilities, this study employs standardized public datasets to ensure data consistency and control experimental variables.

Datasets. We use two datasets in the experiment:CMU-MOSI Zadeh et al. (2016) and CMU-MOSEI Zadeh et al. (2018).We randomly allocated 70%, 10%, and 20% of the data to the training, validation, and test sets, respectively. In subsequent experiments, the same dataset partitioning method was also used for all the baselines.

- **CMU-MOSI** comprises nearly 2,200 short monologue videos (one sentence per clip), where each sample is annotated by human raters with sentiment scores ranging from -3 (strongly negative) to 3 (strongly positive). Acoustic and visual features are extracted at 12.5Hz and 15Hz, respectively, while text is tokenized and encoded as discrete word embeddings.

- **CMU-MOSEI** contains over 23,000 YouTube movie review videos , where each sample is annotated by human raters with sentiment scores ranging from -3 (strongly negative) to 3 (strongly positive) . Each data sample in this dataset consists of three modalities: audio data sampled at a rate of 44.1 kHz, text transcripts, and image frames sampled from the videos at a frequency of 30 Hz.

Baselines. We use the following five state-of-the-art models in previous work as the baselines in this paper.

- **EF-LSTM** : It is a early-fusion method using LSTM.

- **LF-LSTM** : It is a lost-fusion method using LSTM.

- **RAVEN** Wang et al. (2019) : This method is called recurrent attention variants embedding network, which solves multimodal relational reasoning via attention-based alignment.

- **MCTN** Pham et al. (2019) : This method is called multimodal cyclic translation network, which establishes modality translation through cyclic latent space constraints.

- **V2EM** Wei et al. (2022) : This method uses the hierarchical attention spectrum computing module to obtain detailed spectral information.

4.2. Experimental Results

4.2.1. Performance

To compare the models comprehensively, we adapt the connectionist temporal classification (CTC) Graves et al. (2006) method to the prior approaches (e.g., EF-LSTM, MCTN, RAVEN) that cannot be applied directly to the unaligned setting. Table 1 shows the results of various models on the CMU-MOSI dataset. Compared with other centralized models , the accuracy of the model proposed in this paper has been significantly improved, 12.0% and 9.5% compared with EF-LSTM and LF-LSTM , 7.8% compared with MCTN , 5.7% compared with RAVEN , which indicates the structural excellence of the distributed model on Mega-CE^2 proposed in this paper. Because V2EM has the best performance,we choose V2EM as the reference model. The model proposed in this paper is significantly better than V2EM , with anaccuracy increase of 2.8%. F1 scores increased by 2.9%. MAE decreased by 0.025%.

Table 1: Performance comparison on CMU-MOSI dataset using binary accuracy Acc$_2$, weighted F1, and MAE

Model	**Acc$_2$**	**F1**	**MAE**
EF-LSTM	74.1	44.0	0.680
LF-LSTM	76.6	45.8	0.678
MCTN (Pham et al., 2019)	78.3	50.4	0.586
RAVEN (Wang et al., 2019)	80.4	53.1	0.575
V2EM Wei et al. (2022)	82.5	56.3	0.554
Ours	**85.3**	**59.2**	**0.529**

Table 2: Performance comparison on CMU-MOSEI dataset (10× larger than MOSI) using binary accuracy Acc$_2$, weighted F1, and MAE

Model	**Acc$_2$**	**F1**	**MAE**
EF-LSTM	74.5	44.7	0.675
LF-LSTM	76.8	46.1	0.667
MCTN (Pham et al., 2019)	79.2	51.3	0.581
RAVEN (Wang et al., 2019)	81.2	51.1	0.564
V2EM Wei et al. (2022)	82.2	54.9	0.529
Ours	**86.6**	**59.7**	**0.514**

In Table 2 , we further evaluate the results of each model on the CMU-MOSEI dataset.Mega-CE^2's performance gains are particularly significant on larger-scale CMU-MOSEI , evidencing enhanced generalization and scalability for complex multimodal scenarios.

Empirically, to better demonstrate the convergence rate of our method, we conducted experiments on the CMU MOSEI (non-IID) dataset. The results in Figure 3 show that our Mega-CE2 obtains superior convergence compared to baselines.

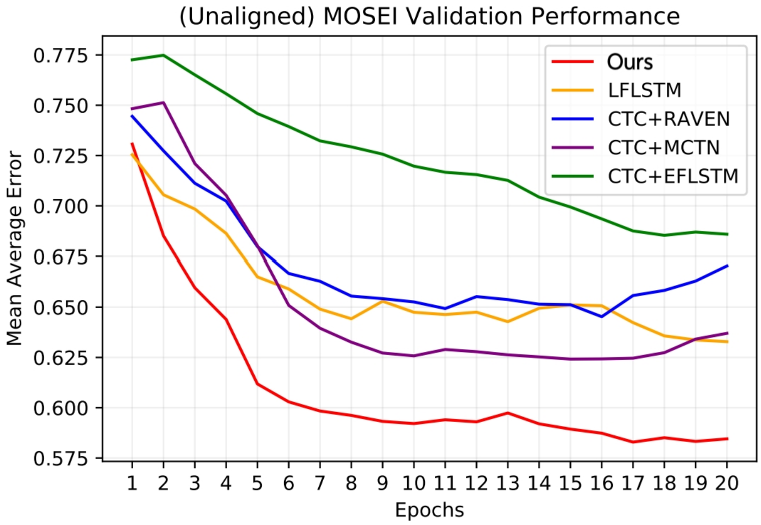

Figure 3: Framework Diagram of the EECC

4.2.2. Computational Efficiency And Overhead

To evaluate the computational efficiency and overhead of Mega-CE^2 in EECC , we measure the average processing time per sample on CMU-MOSEI dataset, comparing key time metrics between EECC collaborative deployment and cloud-only deployment, as shown in Table 3. The feature generation time t_{gen} refers to the duration required to convert raw multimodal data into trainable feature sequences. The feature transmission time t_{trans} captures the latency of transferring the generated feature files (in .pkl format) from end devices to edge servers. The model inference time t_{infer} represents the computation time for generating predictions. The result transmission time $t_{trans\text{-}score}$ denotes the delay in delivering the prediction results to end users. In the Cloud-Only, the raw data transmission time $t_{trans\text{-}raw}$ becomes the dominant factor, representing the substantial latency required to transfer uncompressed multimodal data (.txt/.wav/.mp4 formats) from end devices to the cloud.

In the cloud-only deployment, where the cloud server must complete the full sequence generation, model inference, and result transmission, the total latency reaches 3.1322s. This monolithic process lacks intermediate feedback, creating a synchronization delay phase for end users that significantly degrades interaction experience, particularly during network congestion or task queuing. By contrast, Mega-CE^2 achieves merely 0.0452s,enabling immediate user feedback while reducing latency by approximately 96.2% compared to the centralized approach. This efficiency gain liberates substantial cloud computational resources to enhance system concurrency while dramatically shortening the time-to-first-response.

Table 3: End-to-end latency comparison between Mega-CE^2 and Cloud-Only approaches. Mega-CE^2 significantly reduces latency by offloading feature generation to the edge, while Cloud-Only incurs higher delay due to centralized training.

Metric	**Mega-CE^2**	**Cloud-Only**
Feature Generation t_{gen}	0.0450 s	-
Feature Transmission t_{trans}	0.0001 s	–
Raw Data Transmission $t_{trans-raw}$	-	1.7500s
Model Inference t_{infer}	3.2086 s	1.3821 s
Result Transmission $t_{trans-score}$	0.0001	0.0001
End-to-End Latency	**0.0452 s**	**3.1322 s**
Transmission type	Feature data (.pkl)	Multimodal data (.txt/.wav/.mp4)

Although the training duration remains comparable, users experience smoother interaction as local processing eliminates dependency on cloud computation and transmission.

Network transmission constitutes a critical factor affecting both the response performance and service scalability of generation systems. As evidenced in Table 3, centralized deployment necessitates transferring multimodal data (.txt /.wav /.mp4) from cloud to client, where audio/video files ranging from 5000-10000KB inevitably introduce substantial transmission latency.In contrast, Mega-CE^2 only requires transmitting lightweight perceptual features (.pkl), which maintain a compact size of 96 - 98KB with measured transmission time of merely 0.0001s , significantly lower than multimodal data transfer overhead. These feature files encapsulate irreversible semantic information, offering both reduced size and enhanced security that collectively alleviate network load.

Consequently, Mega-CE^2 compresses the 'first-packet delay' to sub-50ms levels, demonstrating superior real-time responsiveness that strongly supports low-latency, high-interactivity application scenarios. And Mega-CE^2 demonstrates superior network transmission efficiency, which is suitable for bandwidth-constrained, latency-sensitive, or large-scale user deployment scenarios.

4.2.3. Ablation Experiments

To verify whether cross-modal attention enables the model to achieve better representation learning, we conducted ablation experiments on the larger dataset CMU-MOSEI based on Mega-CE^2.

Specifically, we investigated two scenarios: (1) unimodal Transformers that process only single-modal data, including models handling solely language, audio, or visual data; and (2) models employing only a single cross-modal Transformer, i.e., the [V,A→L], [L,A→V], or [L,V→A] networks. In Table 4, We observe that neither Unimodal Transformers nor Single Cross-modal Transformers outperform the original model. The text-only modality achieves the best performance across all metrics, which may be attributed to the fact that non-linguistic information (e.g., tone, facial expressions) requires more sophisticated modeling or

Table 4: Ablation Study Results Comparison. We investigated two scenarios: (1) unimodal Transformers that process only single-modal data, including models handling solely language, audio, or visual data; and (2) models employing only a single cross-modal Transformer, i.e., the [V,A→L], [L,A→V], or [L,V→A] networks.

Type	**$\mathbf{Acc_2}$**	**F1**	**MAE**
Unimodal Transformers			
Text-only	79.4	50.2	0.583
Audio-only	65.6	48.6	0.664
Video-only	66.4	48.3	0.659
Single Cross-modal Transformer			
only V,A→L	82.1	52.6	0.575
only L,A→V	79.7	52.1	0.611
only L,V→A	79.2	51.4	0.620
Ours	**86.6**	**59.7**	**0.514**

complementary integration with textual data. This finding aligns with previous observations reported in prior work Pham et al. (2019).

4.2.4. Number Of Parameters

According to empirical research data, the parameter size of adapter models typically accounts for only 0.5% - 8% of the total parameters in the language model Zhang et al. (2023). As shown in Table 5 , it can be observed that the parameters of the adapter are significantly fewer than those of the global model. Adapter-based fine-tuning reduces the number of parameters requiring adjustment while maintaining model performance comparable to full fine-tuning. This facilitates model adaptation under resource-constrained edge environments and preserves more pre-trained knowledge for specific tasks, thereby promoting the development of personalized edge models. The additional parameters of the global model primarily originate from cross-modal interaction and memory layers, with the self-attention layers and linear projection layers each accounting for approximately 40% , collectively constituting around 80% of the total parameters. The remaining parameters (e.g., Embedding, Norm, and Head) account for roughly 20%.

5. Conclusion

In this paper, we propose a multi-modal heterogeneous aggregation framework Mega-CE^2 for EECC scenarios, which systematically decomposes and aggregates complex multi-modal tasks to mitigate the limitations of conventional distributed frameworks, including inflexible resource allocation and inefficient multi-modal fusion. Specifically, the end preprocess collected multi-modal data to generate location-aware feature representations, which are synchronously aggregated at the edge layer to minimize transmission latency and communication overhead. Additionally, the edge layer utilizes cross-modal attention modules

Table 5: Comparison of the number of the global model and the adapters parameters employed in this study under Mega-CE^2.

Module	**Global Model (MB)**	**Adapter (MB)**
Self-attention Layer	19.48	0.36
Linear Projection	20.57	0.09
Trainable Parameters	50.41	0.45

to capture dependencies among multi-modal data, while also receiving globally optimized model parameters from the cloud. It then fine-tunes these parameters using local adapters to create personalized local models. Finally, experiments on two multi-modal analysis datasets demonstrate that our framework achieves a better balance in terms of accuracy, computational efficiency, and cost, providing users with a more interactive experience.

However, the proposed framework still has some limitations, such as the lack of a comprehensive trust management model, which makes it difficult to ensure computational trustworthiness during cross-domain collaboration. In the future, we will continue to optimize our framework to address trust deficiency issues in complex scenarios and introduce trust management models to better adapt to the requirements of low latency and high accuracy in real-time tasks in cloud-edge-end environments.

Acknowledgement

This work was supported by the National Science and Technology Major Project (No. 2022ZD0115901), in part by the National Natural Science Foundation of China Project under Grants No. 62177007.

References

Vaswani Ashish. Attention is all you need. *Advances in neural information processing systems*, 30:I, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Zhuoqing Chang, Shubo Liu, Xingxing Xiong, Zhaohui Cai, and Guoqing Tu. A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet of Things Journal*, 8(18):13849–13875, 2021.

Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *arXiv preprint arXiv:2204.12703*, 2022.

Sijing Duan, Dan Wang, Ju Ren, Feng Lyu, Ye Zhang, Huaqing Wu, and Xuemin Shen. Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. *IEEE Communications Surveys & Tutorials*, 25(1):591–624, 2022.

Sri Harsha Dumpala, Imran Sheikh, Rupayan Chakraborty, and Sunil Kumar Kopparapu. Audio-visual fusion for sentiment classification using cross-modal autoencoder. In *32nd conference on neural information processing systems (NIPS 2018)*, pages 1–4, 2019.

Biyi Fang, Xiao Zeng, and Mi Zhang. Nestdnn: Resource-aware multi-tenant on-device deep learning for continuous mobile vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 115–127, 2018.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Biao Hou and Junxing Zhang. Real-time surveillance video salient object detection using collaborative cloud-edge deep reinforcement learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. Model complexity of deep learning: A survey. *Knowledge and Information Systems*, 63:2585–2619, 2021.

Liang Jia, Jin Tan, Lijin Qi, and Mingwen Lin. A more efficient inference model for multimodal emotion recognition. In *The 16th Asian Conference on Machine Learning (Conference Track)*, 2024.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

Yeachan Kim, Kang-Min Kim, and SangKeun Lee. Adaptive compression of word embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3950–3959, 2020.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Quan Nguyen, Hieu H Pham, Kok-Seng Wong, Phi Le Nguyen, Truong Thao Nguyen, and Minh N Do. Feddct: Federated learning of large convolutional neural networks on resource-constrained devices using divide and collaborative training. *IEEE Transactions on Network and Service Management*, 21(1):418–436, 2023.

Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6892–6899, 2019.

Ju Ren, Deyu Zhang, Shiwen He, Yaoxue Zhang, and Tao Li. A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for computational linguistics. Meeting*, volume 2019, page 6558, 2019.

Fan Wang, Shengwei Tian, Long Yu, Jing Liu, Junwen Wang, Kun Li, and Yongtao Wang. Tedt: transformer-based encoding–decoding translation network for multimodal sentiment analysis. *Cognitive Computation*, 15(1):289–303, 2023.

Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7216–7223, 2019.

Qinglan Wei, Xuling Huang, and Yuan Zhang. Fv2es: A fully end2end multimodal system for fast yet effective video emotion recognition inference. *IEEE Transactions on Broadcasting*, 69(1):10–20, 2022.

Zhiyuan Wu, Sheng Sun, Yuwei Wang, Min Liu, Bo Gao, Quyang Pan, Tianliu He, and Xuefeng Jiang. Agglomerative federated learning: Empowering larger model training via end-edge-cloud collaboration. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 131–140. IEEE, 2024a.

Zhiyuan Wu, Sheng Sun, Yuwei Wang, Min Liu, Quyang Pan, Junbo Zhang, Zeju Li, and Qingxiang Liu. Exploring the distributed knowledge congruence in proxy-data-free federated distillation. *ACM Transactions on Intelligent Systems and Technology*, 15(2): 1–34, 2024b.

Zehui Xiong, Yang Zhang, Dusit Niyato, Ruilong Deng, Ping Wang, and Li-Chun Wang. Deep reinforcement learning for mobile 5g and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine*, 14(2):44–52, 2019.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88, 2016.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

Yan Zhuang, Zhenzhe Zheng, Yunfeng Shao, Bingshuai Li, Fan Wu, and Guihai Chen. Nebula: An edge-cloud collaborative learning framework for dynamic edge environments. In *Proceedings of the 53rd International Conference on Parallel Processing*, pages 782–791, 2024.