FEDMAGE: ARCHITECTURE-AWARE GRAPH HYPERNETWORK FOR HETEROGENEOUS MULTIMODAL FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal federated learning aims to collaboratively train multimodal models across decentralized clients while preserving data privacy. However, existing methods struggle with highly heterogeneous client tasks and architectures, often rely on public data for knowledge distillation, and incur substantial communication overhead. To address these challenges, we propose **FedMAGE**, a graph hypernetwork-based multimodal federated learning framework. FedMAGE represents client models built from a predefined operator vocabulary as computational graphs and employs a server-side hypernetwork to generate architecture-specific parameters, enabling implicit parameter sharing across heterogeneous clients without requiring a shared parameter space. To improve the generalization of a shared decoder across diverse operator types, we further introduce Type-aware Decoding, which conditions the parameter generation process on operator-specific information. We conduct extensive experiments on three multimodal tasks: image-text retrieval, sentiment recognition, and scene recognition. Compared with a broad range of federated learning baselines, FedMAGE consistently achieves superior performance, improving accuracy by an average of 3.8% over the strongest baseline, while substantially reducing communication time compared with existing multimodal federated learning methods.

1 INTRODUCTION

Federated learning (FL) enables multiple clients to collaboratively train models without sharing their raw data and has become an important paradigm for privacy-preserving machine learning (McMahan et al., 2017; Kairouz et al., 2021; Yang et al., 2019). As modern mobile devices, IoT platforms, and autonomous systems increasingly generate image, text, audio, and sensor streams simultaneously, multimodal federated learning (MFL) is gaining traction (Liu et al., 2020; Zhao et al., 2022). Prior studies have shown that MFL involves not only the statistical heterogeneity commonly encountered in classical FL, but also inherent differences in modality combinations, task objectives, and resource constraints (Xiong et al., 2022; Che et al., 2023; Lin et al., 2023; Feng et al., 2023).

A central obstacle is that the dominant FL research still assumes a shared parameter space, or at least a set of mutually alignable model components, across clients (Li et al., 2020; Karimireddy et al., 2020; Wang et al., 2020; Acar et al., 2021; Li et al., 2021c). This assumption often fails in multimodal settings. In practice, different clients observe different modality combinations, serve distinct downstream tasks, and operate under heterogeneous constraints in computation, and communication bandwidth. Consequently, they may need to adopt local models with different depths or even from different architecture families. Recent studies have explicitly emphasized that realistic multimodal federated systems often exhibit **personalized architectures**, **diverse fusion strategies**, and **varying task complexity** across clients. These characteristics fall outside the comfortable regime of combining a shared backbone with local adaptation modules.

Existing attempts to handle this challenge can be broadly divided into two categories (Bao et al., 2026; Le et al., 2024; Wang et al., 2024; Nguyen et al., 2024; Liu et al., 2025b;a; Nguyen et al., 2025; Qiu et al., 2025a). The first circumvents direct weight aggregation through knowledge transfer, typically in the form of distillation, contrastive learning, or prototype exchange over public or

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

proxy data (Li & Wang, 2019; Lin et al., 2020; He et al., 2020; Zhu et al., 2021; Cho et al., 2022; Le et al., 2023; Shi et al., 2024). These methods are appealing because they avoid strict weight alignment, but the trade-off is equally clear: they often require public data, and repeated exchange of logits, representations, or prototypes. This overhead is especially burdensome in multimodal settings. The second category instead rethinks collaboration through parameter generation (Shamsian et al., 2021; Litany et al., 2022; Chen & Zhang, 2024). These methods take a more fundamental step by representing neural architectures as computational graphs and learning a graph hypernetwork that maps architecture descriptions to model parameters, thereby enabling implicit parameter sharing across heterogeneous architectures without requiring external data. This is a consequential conceptual shift: **implicit parameter sharing** provides a promising alternative to explicit parameter aggregation for architecture-heterogeneous FL.

Yet two issues remain insufficiently resolved. First, heterogeneous multimodal client models require a common graph representation that accommodates differences in modality composition, task-specific components, and fusion structure within a predefined operator vocabulary. Second, once a single shared decoder is responsible for generating parameters for operators of very different types, shapes, and semantic roles, generation quality can degrade systematically across operator categories.

To address these challenges, we propose **FedMAGE**, a **Fed**erated **M**ultimodal **A**rchitecture-aware **G**raph-based parameter g**E**neration framework. Rather than constructing a hand-designed common parameter space for heterogeneous clients, FedMAGE learns a shared graph hypernetwork that conditionally generates parameters across tasks, modalities, and architectures. A single shared MLP decoder maps representations of heterogeneous operators through the same unconditional function, which can limit its generalization to certain operator types. This issue is amplified in multimodal models, where modality-specific encoders, fusion modules, and prediction heads can differ substantially in semantics and tensor structure. We therefore introduce Type-aware Decoding (TAD), which explicitly conditions parameter generation on operator type. TAD preserves cross-architecture parameter sharing while allowing the decoder to better accommodate the distinct requirements of convolutional layers, linear projections, fusion blocks, and task-specific heads.

We evaluate FedMAGE on three representative multimodal tasks: image-text retrieval, sentiment recognition, and scene recognition. We compare it with a diverse set of baselines spanning naive, personalized, and multimodal FL. Across the three tasks, FedMAGE improves the primary evaluation metric by an average of 3.8% over the strongest competing method. By avoiding additional distillation rounds and large-scale representation exchange, FedMAGE also substantially reduces communication time relative to existing MFL approaches. These results demonstrate that graph-based parameter generation provides a favorable accuracy-efficiency trade-off under joint modality and architecture heterogeneity.

The main contributions of this paper are as follows:

- We introduce FedMAGE, a unified framework for task- and architecture-heterogeneous multimodal federated learning that represents client models built from a predefined operator vocabulary as computational graphs and enables implicit sharing through a graph hypernetwork.
- We propose Type-aware Decoding, which conditions the decoding stage explicitly on operator type, improving parameter generation quality across heterogeneous operator semantics.
- Extensive experiments demonstrate that FedMAGE consistently improves both predictive performance and communication efficiency over existing FL baselines, indicating that architecture-aware graph-based parameter generation is a promising path toward truly heterogeneous MFL.

2 Preliminaries and Problem Setup

We consider a federated system with clients $\mathcal{C} = \{1, \ldots, K\}$ and a central server. Let $\mathcal{M} = \{1, \ldots, M\}$ be the set of all modality types. Client k observes a modality subset $\mathcal{M}_k \subseteq \mathcal{M}$, solves a task τ_k with label space $\mathcal{Y}_{\tau_k}$, and owns private data $\mathcal{D}_k = \{(x_{k,i}^{\Omega_{k,i}}, y_{k,i})\}_{i=1}^{n_k}$, where $\Omega_{k,i} \subseteq \mathcal{M}_k$ is the set of observed modalities for sample i.

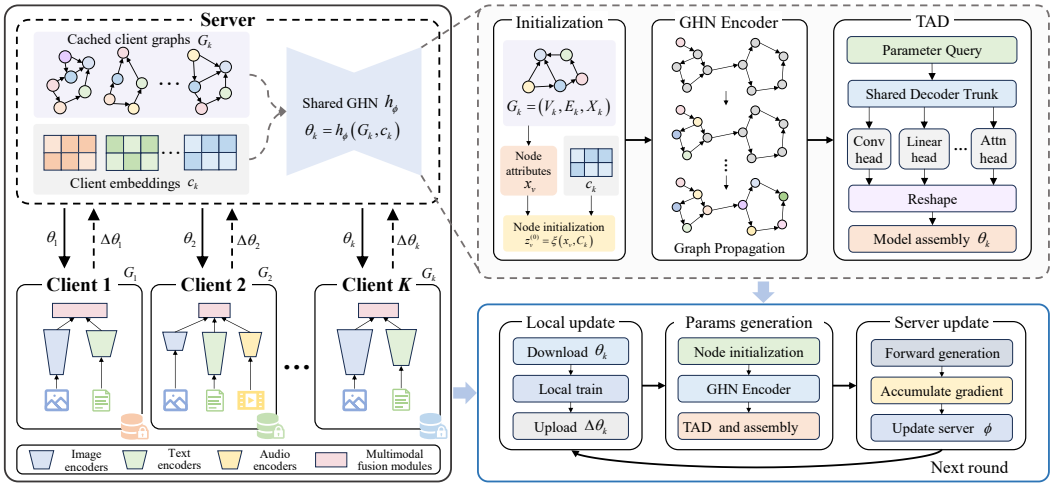


Figure 1: Overview of FedMAGE. The server-side hypernetwork generates parameters tailored to heterogeneous client architectures, while client updates optimize the shared generator without directly averaging incompatible weights. The lower-right section illustrates the training and optimization process of the overall framework.

Each client is sampled from a client-specific distribution $\mathcal{P}_k$ over $\mathcal{X}_{\mathcal{M}_k} \times \mathcal{Y}_{\tau_k}$, where $\mathcal{X}_{\mathcal{M}_k} = \prod_{m \in \mathcal{M}_k} \mathcal{X}^{(m)}$. The distributions may differ in data statistics, available modalities, label spaces, and task semantics. We also allow each client to use its own architecture $a_k \in \mathcal{A}$ with parameter space $\Theta(a_k) \subseteq \mathbf{R}^{d_k}$. The local predictor is $f_k(\cdot; \theta_k, a_k) : \mathcal{X}_{\mathcal{M}_k} \to \mathcal{Y}_{\tau_k}$, where $\theta_k \in \Theta(a_k)$.

Given a task-specific loss ℓ_{τ_k}, the empirical risk of client k is

$$\widehat{F}_k(\theta_k; a_k, \tau_k) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell_{\tau_k}\big(f_k(x_{k,i}^{\Omega_{k,i}}; \theta_k, a_k), y_{k,i}\big). \tag{1}$$

If client k is trained independently, its local solution is

$$\theta_k^{\mathrm{loc}} \in \arg\min_{\theta_k \in \Theta(a_k)} \widehat{F}_k(\theta_k; a_k, \tau_k). \tag{2}$$

Independent training, however, cannot exploit complementary knowledge across clients, especially when local multimodal data are scarce or incomplete.

Classical FL assumes a shared parameter space and optimizes a single model. In our setting, this assumption no longer holds because $\mathcal{M}_k$, τ_k, a_k, and $\Theta(a_k)$ may all vary across clients. We therefore formulate heterogeneous multimodal FL as personalized federated optimization over $\Theta = \{\theta_1, \ldots, \theta_K\}$:

$$F(\Theta) = \sum_{k=1}^{K} p_k \widehat{F}_k(\theta_k; a_k, \tau_k), \quad p_k = \frac{n_k}{\sum_{j=1}^{K} n_j}. \tag{3}$$

The objective is to minimize $F(\Theta)$ without exchanging raw data or assuming aligned client parameter spaces.

Let Π denote a federated training protocol that returns personalized parameters Θ_Π^T after T communication rounds, and let $C_T(\Pi)$ be its communication cost. Under a budget B, the system-level problem is

$$\Pi^\star \in \arg\min_{\Pi} F(\Theta_\Pi^T) \quad \text{s.t.} \quad C_T(\Pi) \le B. \tag{4}$$

Thus, this paper studies how to learn accurate personalized multimodal models under simultaneous data, modality, task, and architecture heterogeneity, while respecting the federated constraint that all raw data remain local.

3 METHODOLOGY

FedMAGE replaces explicit weight aggregation with architecture-conditioned parameter generation. An overview of the proposed framework is presented in Figure 1. Its core component is a server-side graph hypernetwork h_ϕ that maps a client architecture graph G_k and a learnable client embedding c_k to personalized model parameters. At each communication round, the server generates model initializations for the selected clients. Each client then performs local optimization and returns its parameter displacement, which is used to update the shared hypernetwork and the corresponding client embedding. We next describe the graph representation, Type-aware Decoding, and federated optimization procedure. The complete algorithm is provided in Algorithm 1.

3.1 GRAPH HYPERNETWORK

Typed DAG representation of client architectures. For each client k, we convert its neural architecture a_k into a typed directed acyclic graph (DAG)

$$G_k = (V_k, E_k, X_k), \tag{5}$$

where V_k is the set of operator nodes, E_k denotes directed computational dependencies, and $X_k = \{x_v : v \in V_k\}$ contains node attributes. Each node v has an operator type $o_v \in \mathcal{O}$ and, if the operator is parametric, a target parameter shape s_v. The operator vocabulary $\mathcal{O}$ is fixed in advance and includes common primitives such as convolution, linear projection, normalization, embedding, attention, pooling, activation, addition, concatenation, and MLP/FFN blocks. Therefore, any client model that can be decomposed into these primitives can be represented in the same graph space, regardless of its modality configuration, depth, model family, or fusion strategy.

Node initialization and GHN encoder. We initialize each node by embedding its operator type, shape, modality role, and topological position. The initial node representation is: $z_v^{(0)} = \xi(x_v, c_k)$, where c_k is a learnable client embedding that summarizes persistent client-specific factors such as modality composition, task identity, and resource profile. The graph encoder then propagates information over the typed DAG:

$$\begin{aligned} m_v^{(\ell)} &= \sum_{u:(u,v)\in E_k} M_\phi^{(\ell)}(z_u^{(\ell)}, z_v^{(\ell)}, e_{uv}), \\ z_v^{(\ell+1)} &= U_\phi^{(\ell)}(z_v^{(\ell)}, m_v^{(\ell)}), \quad \ell = 0, \dots, L-1. \end{aligned} \tag{6}$$

where $M_\phi^{(\ell)}$ and $U_\phi^{(\ell)}$ denote the message and update functions of the GHN encoder, respectively. The edge feature e_{uv} encodes the edge direction and the connection type. After L propagation layers, $z_v^{(L)}$ captures both the local operator semantics and its architectural context.

Basic weight decoding. For each parametric node v, a standard GHN decodes its final node representation into a vectorized parameter prediction and reshapes it into the tensor required by the corresponding operator:

$$\begin{aligned} \hat{\theta}_{k,v} &= \mathrm{Reshape}_{s_v}\left(D_\phi(z_v^{(L)})\right), \\ \hat{\theta}_k &= \{\hat{\theta}_{k,v} : v \in V_k^p\}. \end{aligned} \tag{7}$$

where $V_k^p \subseteq V_k$ is the set of parametric nodes and D_ϕ is the decoder. This design allows the same hypernetwork to generate weights for clients with different depths, widths, and topologies. However, using an operator-agnostic decoder for all operator types forces convolutional kernels, linear projections, attention matrices, fusion blocks, and task heads to share one decoding rule, which can weaken generation quality.

3.2 TYPE-AWARE DECODING

Motivation. The parameter tensors associated with different operator types have different statistical structures and functional roles. For example, convolutional kernels capture local spatial patterns, attention projections model token interactions, and fusion modules encode cross-modal dependencies. A shared MLP decoder must map all these heterogeneous parameter distributions from the same latent space, which makes the generation problem unnecessarily entangled. We therefore introduce

Algorithm 1 FedMAGE

Input: client datasets $\{\mathcal{D}_k\}_{k=1}^K$, architecture graphs $\{G_k\}_{k=1}^K$, rounds T, local steps E, client fraction ρ, local learning rate α, server learning rates η, η_c.
Initialize: GHN parameters ϕ^0 and client embeddings $\{c_k^0\}_{k=1}^K$.
for $t = 0, 1, \ldots, T-1$ **do**
  Server samples participants $\mathcal{S}_t \subseteq \mathcal{C}$ with $|\mathcal{S}_t| = \max(\rho K, 1)$.
  Server generation and download:
  for each client $k \in \mathcal{S}_t$ in parallel **do**
    Generate $\theta_k^t = h_{\phi^t}(G_k, c_k^t)$.
    Send θ_k^t to client k.
  end for
  Client local training and upload:
  for each client $k \in \mathcal{S}_t$ in parallel **do**
    Set $\tilde{\theta}_k^{t,0} \leftarrow \theta_k^t$.
    for $e = 0, 1, \ldots, E-1$ **do**
      Update $\tilde{\theta}_k^{t,e+1} \leftarrow \tilde{\theta}_k^{t,e} - \alpha \nabla_\theta \widehat{F}_k(\tilde{\theta}_k^{t,e}; a_k, \tau_k)$.
    end for
    Compute $\Delta\theta_k^t \leftarrow \tilde{\theta}_k^{t,E} - \theta_k^t$.
    Upload $\Delta\theta_k^t$ to the server; transmit G_k only if the architecture has changed.
  end for
  Server GHN update:
  Update $\phi^{t+1} \leftarrow \phi^t + \eta \sum_{k \in \mathcal{S}_t} p_k J_{\phi,k}^{t\top} \Delta\theta_k^t$.
  for each client $k \in \mathcal{S}_t$ **do**
    Update $c_k^{t+1} \leftarrow c_k^t + \eta_c J_{c,k}^{t\top} \Delta\theta_k^t$.
  end for
end for
Output: trained GHN h_{ϕ^T} and personalized client models $\theta_k^T = h_{\phi^T}(G_k, c_k^T)$.

Type-aware Decoding (TAD), which retains a shared decoding trunk for cross-architecture transfer and uses type-specific heads for specialized parameter generation.

Parameter query construction. For each parametric node, we first build a query by combining its graph representation with operator-type, parameter-shape, and shape information:

$$q_v = Q_\psi(z_v^{(L)}, e_{o_v}, e_{s_v}, c_k), \tag{8}$$

where e_{o_v} and e_{s_v} are embeddings of the operator type and parameter shape, respectively. And Q_ψ is a query builder. The client embedding c_k conditions the generated weights on the local task and modality context.

Shared trunk with type-specific heads. The query is first passed through a shared decoder trunk and then through a head selected by the operator type:

$$\hat{\theta}_{k,v} = \mathrm{Reshape}_{s_v}\left(N_{o_v}\left(H_{o_v}(D_0(q_v))\right)\right). \tag{9}$$

where D_0 is the shared trunk, H_{o_v} is the head associated with operator type o_v, and N_{o_v} denotes type-specific normalization. The shared trunk transfers knowledge across architectures, while the specialized heads adapt the final mapping to operator-specific parameter statistics. This design is particularly suitable for multimodal models, whose modality encoders, fusion modules, and task-specific heads often differ substantially in semantic roles and tensor structures.

3.3 Federated Optimization

Model generation and local training. At communication round t, the server selects a subset of clients $\mathcal{S}_t$ and generates an initial model for each participating client:

$$\theta_k^t = h_{\phi^t}(G_k, c_k^t), \qquad k \in \mathcal{S}_t. \tag{10}$$

The server sends θ_k^t to client k. The client then performs E local optimization steps on its private data, starting from $\tilde{\theta}_k^{t,0} = \theta_k^t$:

$$\tilde{\theta}_k^{t,e+1} = \tilde{\theta}_k^{t,e} - \alpha \nabla_\theta \widehat{F}_k(\tilde{\theta}_k^{t,e}; a_k, \tau_k), \quad e = 0, \ldots, E-1. \tag{11}$$

After local training, the client uploads the update direction $\Delta\theta_k^t = \tilde{\theta}_k^{t,E} - \theta_k^t$, together with its architecture graph G_k and client embedding information. If the architecture does not change, G_k can be cached by the server and need not be resent.

Hypernetwork update. The server treats the locally trained parameters $\theta_k^t + \Delta\theta_k^t$ as a first-order target for the generator. This corresponds to minimizing the surrogate objective

$$\begin{aligned} \mathcal{L}_{\mathrm{srv}}(\phi, \{c_k\}) &= \frac{1}{2} \sum_{k \in \mathcal{S}_t} p_k \left\| r_k(\phi) \right\|_2^2, \\ r_k(\phi) &= h_\phi(G_k, c_k) - \mathrm{sg}(\theta_k^t + \Delta\theta_k^t). \end{aligned} \tag{12}$$

where $\mathrm{sg}(\cdot)$ stops gradients through the local target. Evaluated at $\theta_k^t = h_{\phi^t}(G_k, c_k^t)$, the first-order update can be written as a Jacobian-vector product:

$$\phi^{t+1} = \phi^t + \eta \sum_{k \in \mathcal{S}_t} p_k J_{\phi,k}^{t\top} \Delta\theta_k^t, \quad J_{\phi,k}^t = \frac{\partial h_{\phi^t}(G_k, c_k^t)}{\partial \phi}. \tag{13}$$

The same principle updates the client embedding:

$$c_k^{t+1} = c_k^t + \eta_c J_{c,k}^{t\top} \Delta\theta_k^t, \quad J_{c,k}^t = \frac{\partial h_{\phi^t}(G_k, c_k^t)}{\partial c_k}. \tag{14}$$

Thus, the framework avoids averaging incompatible client parameters. Instead, it updates a shared parameter generator so that future generated parameters follow the directions favored by client-side optimization.

Inference and architectural generalization. After training, FedMAGE can generate a personalized model for each existing client from its architecture graph and learned embedding. Because the generator is defined on typed computation graphs rather than a fixed parameter layout, it can also be applied to compatible unseen architectures composed of the predefined operator vocabulary, enabling architecture-aware parameter generation beyond the architectures observed in a single communication round.

4 Experiments

We evaluate FedMAGE on three representative multimodal tasks and compare it against naive FL, personalized FL, and multimodal FL baselines. The experiments are designed to answer three questions: (i) whether FedMAGE improves predictive performance under heterogeneous multimodal clients; (ii) whether Type-aware Decoding contributes to the improvement; and (iii) whether the graph hypernetwork remains robust under stronger architecture heterogeneity and unseen architectures.

4.1 Experimental Setup

Datasets and tasks. We consider three multimodal tasks. Task1 is image-text retrieval on Flickr30K (Plummer et al., 2015), where the system evaluates both image-to-text and text-to-image retrieval. Task2 is multimodal sentiment recognition on CMU-MOSEI (Bagher Zadeh et al., 2018), which contains textual, visual, and acoustic modalities. Task3 is RGB-D scene recognition on NYU-Depth-v2 (Silberman et al., 2012), where RGB and depth inputs provide complementary scene information. These tasks cover cross-modal matching, multimodal affect understanding, and multimodal visual recognition.

Baselines. We compare FedMAGE with three groups of baselines: naive FL methods, namely FedAvg (McMahan et al., 2017), MOON (Li et al., 2021a), and FedDC (Gao et al., 2022); personalized FL methods, covering FedKDC (Qiu et al., 2025b), FedPer (Arivazhagan et al., 2019), pFedMe (Dinh et al., 2020), and Ditto (Li et al., 2021b); and multimodal FL methods, including CreamFL (Yu et al., 2023), CAR-MFL (Poudel et al., 2024), and PmcmFL (Bao et al., 2026).

Metrics. For Task1, we report the mean recall (mR), computed by averaging Recall@1, Recall@5, and Recall@10 over both retrieval directions. For Task2, we report Acc-7, Acc-2, and F1. For Task3, we report Top-1 accuracy and F1. We also report CoT, the average communication time per round

Table 1: Main comparison on three heterogeneous multimodal FL tasks. Higher performance metrics are better, and lower CoT indicates lower communication time.

Type	Method	Task1		Task2				Task3		
		mR↑	CoT↓	Acc-7↑	Acc-2↑	F1↑	CoT↓	Top-1↑	F1↑	CoT↓
Naive FL	FedAvg	66.9	0.2	45.0	76.9	77.0	0.3	70.0	69.3	0.4
	MOON	72.2	2.5	46.8	77.8	78.4	5.4	72.4	71.5	5.6
	FedDC	72.8	3.2	47.1	79.3	79.5	6.7	73.7	73.3	6.1
Personalized FL	FedKDC	72.7	14.2	49.6	79.8	80.6	28.4	74.2	75.6	25.3
	FedPer	72.8	15.9	48.3	82.0	82.1	31.5	76.3	75.6	28.7
	pFedMe	74.2	18.4	50.0	79.1	79.5	25.0	77.4	78.0	30.6
	Ditto	74.8	16.8	50.2	82.5	82.1	23.6	76.9	77.1	29.5
Multimodal FL	CreamFL	75.3	30.4	51.3	84.4	84.3	48.6	79.2	78.8	58.3
	CAR-MFL	72.4	52.6	51.6	84.2	84.5	85.3	78.5	77.3	94.5
	PmcmFL	73.8	35.7	54.6	87.4	86.7	56.4	78.0	78.1	67.4
Ours	FedMAGE	**81.4**	16.3	**56.8**	**89.5**	**90.2**	25.5	**82.4**	**81.0**	30.2

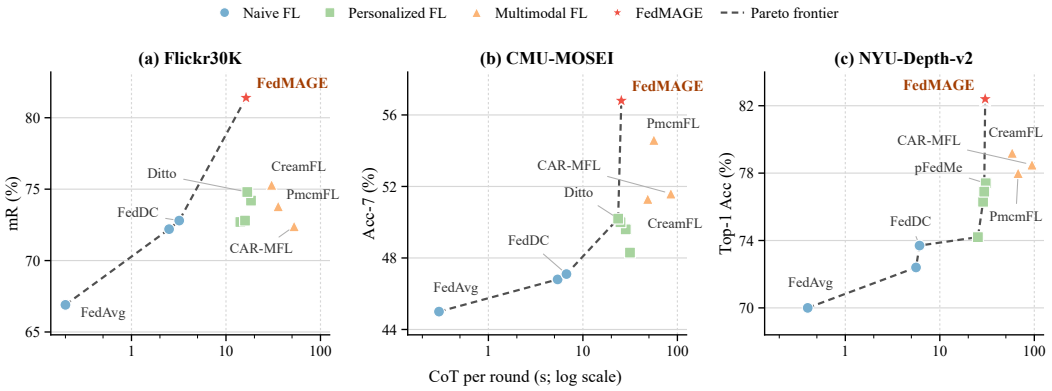


Figure 2: Performance–communication trade-off on three tasks. Dashed curves mark the Pareto frontier.

in seconds, to evaluate communication efficiency. We report CoT primarily for comparison with existing multimodal FL methods, as FedMAGE targets the high communication overhead incurred by methods using knowledge distillation or prototype learning.

Implementation details. Unless stated otherwise, we use heterogeneous client architectures composed of modality-specific encoders, fusion modules, and task heads. The architecture pool includes ResNet and ViT visual encoders; MLP-, GRU-, and Transformer-based modality encoders; and concatenation, element-wise product, and cross-attention fusion modules. All experiments are conducted on NVIDIA RTX 4090 GPUs.

4.2 Main Results

Image-Text Retrieval. As shown in Table 1 and Figure 2, FedMAGE achieves an mR of 81.4%, outperforming the strongest baseline, CreamFL, by 6.1%. This gain is notable because retrieval requires heterogeneous clients to maintain a shared image-text semantic space despite different encoders and fusion paths. Naive and personalized FL methods improve over FedAvg but remain behind the strongest multimodal FL baseline, indicating that local adaptation alone is insufficient for effective cross-modal alignment. By contrast, FedMAGE transfers knowledge through architecture-conditioned parameter generation and reaches this accuracy with only 16.3 seconds of CoT, much lower than CreamFL, CAR-MFL, and PmcmFL. This supports a more efficient route to cross-client sharing than representation-level distillation.

Sentiment Recognition. On CMU-MOSEI, FedMAGE achieves 56.8% Acc-7, 89.5% Acc-2, and 90.2% F1. Compared with PmcmFL, the strongest baseline on this task, FedMAGE improves these metrics by 2.2%, 2.1%, and 3.5%, respectively. The simultaneous gains on fine-grained Acc-7 and binary Acc-2/F1 indicate that FedMAGE improves both sentiment intensity modeling and coarse polarity recognition. This is nontrivial because textual, visual, and acoustic cues contribute unevenly across clients, and heterogeneous architectures can further disturb their fusion. FedMAGE also keeps CoT at 25.5 seconds, clearly below multimodal FL baselines that exchange intermediate representations or prototypes. The result suggests that generated personalized models can preserve useful cross-modal dependencies without requiring extensive knowledge exchange.

Scene Recognition. On NYU-Depth-v2, our approach achieves 82.4% Top-1 accuracy and 81.0% F1, improving over the best baseline by 3.2% and 2.2%, respectively. Unlike the preceding language-related tasks, this setting evaluates the fusion of RGB appearance and depth geometry. The gains therefore demonstrate that FedMAGE is applicable beyond a specific modality pair. It also achieves the best recognition accuracy with 30.2 seconds of CoT, whereas multimodal FL baselines pay a much larger communication cost for richer information exchange. This confirms that graph-based parameter generation can capture transferable multimodal regularities with a better accuracy-efficiency trade-off.

4.3 Ablation Study

Table 2: Effectiveness of Type-aware Decoding.

Variant	T1 mR↑	T2 Acc↑	T3 Acc↑
Shared MLP	77.4	51.0	77.8
Query + Decoder	79.5	53.4	79.8
Separate Decoders	79.1	53.0	79.5
Type-specific Heads	78.0	52.4	79.1
Query + Heads, w/o Norm	79.6	54.2	80.0
Full TAD	**81.4**	**56.8**	**82.4**

Effectiveness of Type-aware Decoding. Table 2 evaluates the contribution of Type-aware Decoding (TAD). Compared with the shared MLP decoder, the complete TAD module improves the primary metric on the three tasks by 4.0%, 5.8%, and 4.6%, respectively. This result shows that a single operator-agnostic decoder limits parameter generation when convolutional layers, projection layers, fusion modules, and task-specific heads share the same latent-to-parameter mapping. The intermediate variants clarify the source of these improvements. Adding a parameter query already improves performance on all three tasks, indicating that explicit operator-type and shape information better conditions the decoding problem. In contrast, using separate decoders or type-specific heads alone gives only partial gains, suggesting that naive specialization can distinguish operator categories but does not fully preserve shared cross-architecture knowledge. The variant without normalization remains below full TAD, especially on Task2, which implies that operator-specific scaling is important when multimodal branches have different activation and parameter statistics. Overall, the best performance of full TAD supports our design principle: effective GHN decoding should be both shared enough to transfer across architectures and conditional enough to respect operator semantics.

4.4 Further Analysis

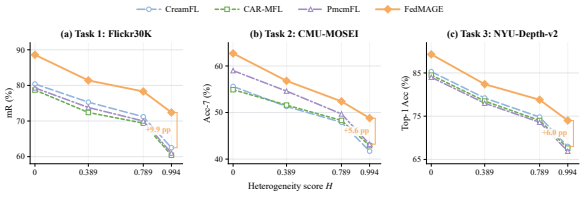


Figure 3: The visualization of heterogeneity stress test. FedMAGE degrades more gracefully and retains clear margins over the strongest baselines at $H = 0.994$.

Heterogeneity stress test. We further evaluate robustness under increasing architecture and modality heterogeneity. Figure 3 visualizes the robustness trends summarized in Table 3. All methods degrade as H increases. The decline is particularly pronounced for multimodal FL baselines whose representation-exchange mechanisms rely on sufficiently compatible local representation spaces. FedMAGE achieves the highest primary-metric scores at the reported heterogeneity level. Even at $H = 0.994$, where heterogeneity is close to maximal under our category design, it exceeds the strongest baseline by 9.9%, 5.6%, and 6.0% on the primary metrics of the three tasks, respectively. This trend suggests that typed graph-based parameter generation is not merely effective in the average case; it is particularly valuable when client architectures become structurally diverse and explicit representation alignment becomes fragile.

Table 3: Robustness under increasing architecture and modality heterogeneity.

H	Method	T1 mR↑	T2 Acc↑	T3 Acc↑
0	CreamFL	80.4	55.6	85.3
	CAR-MFL	78.7	54.9	84.5
	PmcmFL	79.3	59.0	84.0
	FedMAGE	**88.6**	**62.7**	**89.3**
0.389	CreamFL	75.3	51.3	79.2
	CAR-MFL	72.4	51.6	78.5
	PmcmFL	73.8	54.6	78.0
	FedMAGE	**81.4**	**56.8**	**82.4**
0.789	CreamFL	71.2	47.9	74.8
	CAR-MFL	69.4	48.3	74.0
	PmcmFL	70.0	49.7	73.6
	FedMAGE	**78.3**	**52.4**	**78.8**
0.994	CreamFL	62.5	41.7	68.0
	CAR-MFL	60.4	42.9	67.6
	PmcmFL	61.0	43.2	66.9
	FedMAGE	**72.4**	**48.8**	**74.0**

Zero-shot architecture generalization. We further evaluate whether the trained hypernetwork has learned reusable architectural priors rather than merely memorizing the architectures observed during training. To this end, we change the test-time architecture from a 3-layer to a 5-layer CNN or ViT while keeping the trained generator fixed. The "*No switch*" setting serves as an in-distribution reference, whereas Xavier initialization measures how well an unseen architecture performs without knowledge transferred from the hypernetwork.

Table 4: Zero-shot generalization to unseen deeper client architectures.

Scenario	Initialization	Task1 ↑	Task2 ↑	Task3 ↑
No switch	FedMAGE	81.4	56.8	82.4
CNN-3 → CNN-5	Xavier init	26.8	18.4	29.0
	FedMAGE	**59.5**	**35.6**	**60.2**
ViT-3 → ViT-5	Xavier init	25.3	19.5	27.9
	FedMAGE	**56.4**	**34.0**	**64.3**

As shown in Table 4, FedMAGE substantially outperforms Xavier initialization under both unseen architecture shifts. For the CNN-3→CNN-5 switch, our method improves the three task metrics by 32.7%, 17.2%, and 31.2%, respectively; for the ViT-3→ViT-5 switch, the gains are 31.1%, 14.5%, and 36.4%. These gains suggest that the GHN captures transferable relationships among graph topology, operator type, and parameter structure. At the same time, the zero-shot results remain below the no-switch setting, which is expected because deeper architectures introduce new computation paths that are not directly optimized during training. Thus, the experiment provides a conservative but important signal: FedMAGE does not fully solve arbitrary architecture extrapolation, but it can generate meaningful starting parameters for compatible unseen architectures.

5 Conclusion

In this work, we proposed FedMAGE, a graph-hypernetwork framework for heterogeneous multimodal federated learning. By representing client architectures as typed computational graphs, FedMAGE replaces the aggregation of incompatible weights with architecture-conditioned parameter generation. We introduced Type-aware Decoding to account for differences in the semantic roles and parameter structures of heterogeneous operators. Across three multimodal tasks, we achieve consistent gains in both accuracy and communication efficiency, while ablations and stress tests highlight the effectiveness of type-aware generation under strong architecture heterogeneity.

AI USE STATEMENT

In this work, we used GPT-6 to assist with polish writing and the review of related literature. GPT-6 also assisted with research development and implementation, as well as experimental execution.

The authors take responsibility for the final manuscript, including its text, methodology, results, and conclusions.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 describes the architecture representation, graph hypernetwork, type-aware decoding, and federated optimization procedure, with the training workflow summarized in Algorithm 1. Section 4.1 summarizes the experimental tasks, baseline methods, client architecture families, and hardware setup. Appendix B provides additional details on architecture graphs and the server–client communication protocol. Appendix C describes the datasets, data splits, task definitions, and evaluation metrics. Appendix D specifies the factors varied in the heterogeneity stress test and defines the heterogeneity measure.

The supplementary material provides the core components of an anonymized implementation, along with the experimental configurations and scripts for reproducing the results reported in the paper.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated Learning Based on Dynamic Regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

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)*, pp. 2236–2246. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1208. URL https://aclanthology.org/P18-1208/.

Guangyin Bao, Qi Zhang, Duoqian Miao, Zixuan Gong, Chaochao Chen, Liang Hu, and Longbing Cao. Incomplete multimodal federated learning via masking and contrasting prototypes. *IEEE transactions on neural networks and learning systems*, 2026.

Liwei Che, Jiaqi Wang, Yao Zhou, and Fenglong Ma. Multimodal Federated Learning: A Survey. *Sensors*, 23(15):6986, 2023. doi: 10.3390/s23156986.

Jiayi Chen and Aidong Zhang. FedMBridge: Bridgeable Multimodal Federated Learning. In *Forty-First International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=jrHUbftLd6.

Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous Ensemble Knowledge Transfer for Training Large Models in Federated Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 2881–2887. International Joint Conferences on Artificial Intelligence Organization, 2022. doi: 10.24963/ijcai.2022/399.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

Enmao Diao, Jie Ding, and Vahid Tarokh. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TNkPBBYFkXg.

Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

Tiantian Feng, Digbalay Bose, Tuo Zhang, Rajat Hebbar, Anil Ramakrishna, Rahul Gupta, Mi Zhang, Salman Avestimehr, and Shrikanth Narayanan. FedMultimodal: A Benchmark for Multimodal Federated Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4035–4045. ACM, 2023. doi: 10.1145/3580305.3599825.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10112–10121, 2022.

Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 564–571, 2013. URL https://openaccess.thecvf.com/content_cvpr_2013/html/Gupta_Perceptual_Organization_and_2013_CVPR_paper.html.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a1d4c20b182ad7137ab3606f0e3fc8a4-Abstract.html.

Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos I. Venieris, and Nicholas D. Lane. FjORD: Fair and Accurate Federated Learning under Heterogeneous Targets with Ordered Dropout. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12876–12889, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6aed000af86a084f9cb0264161e29dd3-Abstract.html.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021. doi: 10.1561/2200000083.

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*, pp. 5132–5143. PMLR, 2020.

Boris Knyazev, Michal Drozdzal, Graham W. Taylor, and Adriana Romero-Soriano. Parameter Prediction for Unseen Deep Architectures. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29433–29448, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f6185f0ef02dcaec414a3171cd01c697-Abstract.html.

Huy Q. Le, Minh N. H. Nguyen, Chu Myaet Thwal, Yu Qiao, Chaoning Zhang, and Choong Seon Hong. FedMEKT: Distillation-based Embedding Knowledge Transfer for Multimodal Federated Learning, 2023. URL https://arxiv.org/abs/2307.13214.

Huy Q. Le, Chu Myaet Thwal, Yu Qiao, Ye Lin Tun, Minh N. H. Nguyen, and Choong Seon Hong. Cross-Modal Prototype based Multimodal Federated Learning under Severely Missing Modality, 2024. URL https://arxiv.org/abs/2401.13898.

Daliang Li and Junpu Wang. FedMD: Heterogenous Federated Learning via Model Distillation, 2019. URL https://arxiv.org/abs/1910.03581.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021a.

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.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pp. 6357–6368. PMLR, 2021b.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *International Conference on Learning Representations*, 2021c. URL https://openreview.net/forum?id=6YEQUn0QICG.

Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble Distillation for Robust Model Fusion in Federated Learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/18df51b97ccd68128e994804f3eccc87-Abstract.html.

Yi-Ming Lin, Yuan Gao, Mao-Guo Gong, Si-Jia Zhang, Yuan-Qiao Zhang, and Zhi-Yuan Li. Federated Learning on Multimodal Data: A Comprehensive Survey. *Machine Intelligence Research*, 20(4):539–553, 2023. doi: 10.1007/s11633-022-1398-0.

Or Litany, Haggai Maron, David Acuna, Jan Kautz, Gal Chechik, and Sanja Fidler. Federated Learning with Heterogeneous Architectures using Graph HyperNetworks, 2022. URL https://arxiv.org/abs/2201.08459.

Fenglin Liu, Xian Wu, Shen Ge, Wei Fan, and Yuexian Zou. Federated learning for vision-and-language grounding problems. *Proceedings of the AAAI conference on artificial intelligence*, 34 (07):11572–11579, 2020.

Junming Liu, Guosun Zeng, Ding Wang, Yanting Gao, and Yufei Jin. FedRecon: Missing Modality Reconstruction in Distributed Heterogeneous Environments, 2025a. URL https://arxiv.org/abs/2504.09941.

Yi Liu, Cong Wang, and Xingliang Yuan. FedMobile: Enabling Knowledge Contribution-aware Multi-modal Federated Learning with Incomplete Modalities. In *Proceedings of the ACM on Web Conference 2025*, pp. 2775–2786. ACM, 2025b. doi: 10.1145/3696410.3714623.

Huan Ma, Zongbo Han, Changqing Zhang, Huazhu Fu, Joey Tianyi Zhou, and Qinghua Hu. Trustworthy multimodal regression with mixture of normal-inverse gamma distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6881–6893, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/371bce7dc83817b7893bcdeed13799b5-Abstract.html.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

Manh Duong Nguyen, Trung Thanh Nguyen, Huy Hieu Pham, Trong Nghia Hoang, Phi Le Nguyen, and Thanh Trung Huynh. FedMAC: Tackling Partial-Modality Missing in Federated Learning with Cross-Modal Aggregation and Contrastive Regularization, 2024. URL https://arxiv.org/abs/2410.03070.

Manh Duong Nguyen, Trong Nghia Hoang, Thanh Trung Huynh, Quoc Viet Hung Nguyen, and Phi Le Nguyen. Learning Reconfigurable Representations for Multimodal Federated Learning with Missing Data. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rMqQdJJz5r.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30K Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2641–2649, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Plummer_Flickr30k_Entities_Collecting_ICCV_2015_paper.html.

Pranav Poudel, Prashant Shrestha, Sanskar Amgain, Yash Raj Shrestha, Prashnna Gyawali, and Binod Bhattarai. Car-mfl: Cross-modal augmentation by retrieval for multimodal federated learning with missing modalities. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 102–112. Springer, 2024.

Fan Qi and Shuai Li. Adaptive Hyper-graph Aggregation for Modality-Agnostic Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12312–12321, 2024.

Xihang Qiu, Jiarong Cheng, Yuhao Fang, Wanpeng Zhang, Yao Lu, Ye Zhang, and Chun Li. Federated Dialogue-Semantic Diffusion for Emotion Recognition under Incomplete Modalities. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=mI9HqgVuTS.

Xihang Qiu, Wanyong Qiu, Ye Zhang, Kun Qian, Chun Li, Bin Hu, Björn W Schuller, and Yoshiharu Yamamoto. Fedkdc: Consensus-driven knowledge distillation for personalized federated learning in eeg-based emotion recognition. *IEEE Journal of Biomedical and Health Informatics*, 2025b.

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized Federated Learning using Hypernetworks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9489–9502. PMLR, 2021. URL https://proceedings.mlr.press/v139/shamsian21a.html.

Jiangming Shi, Shanshan Zheng, Xiangbo Yin, Yang Lu, Yuan Xie, and Yanyun Qu. CLIP-Guided Federated Learning on Heterogeneity and Long-Tailed Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14955–14963, 2024. doi: 10.1609/aaai.v38i13.29416.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Computer Vision – ECCV 2012*, pp. 746–760. Springer, 2012. URL https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5005–5013, 2016. doi: 10.1109/CVPR.2016.541. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Wang_Learning_Deep_Structure-Preserving_CVPR_2016_paper.html.

Shu Wang, Zhe Qu, Yuan Liu, Shichao Kan, Yixiong Liang, and Jianxin Wang. FedMMR: Multi-Modal Federated Learning via Missing Modality Reconstruction. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024. doi: 10.1109/ICME57554.2024.10688162.

Baochen Xiong, Xiaoshan Yang, Fan Qi, and Changsheng Xu. A Unified Framework for Multi-Modal Federated Learning. *Neurocomputing*, 480:110–118, 2022. doi: 10.1016/j.neucom.2022.01.063.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019. doi: 10.1145/3298981.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006/.

Qiying Yu, Yang Liu, Yimu Wang, Ke Xu, and Jingjing Liu. Multimodal Federated Learning via Contrastive Representation Ensemble. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Hnk1WRMAYqg.

Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph HyperNetworks for Neural Architecture Search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkgW0oA9FX.

Yuchen Zhao, Payam Barnaghi, and Hamed Haddadi. Multimodal Federated Learning on IoT Data. In *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 43–54. IEEE, 2022. doi: 10.1109/IOTDI54339.2022.00011.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12878–12889. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhu21b.html.

A Related Work

Federated learning with heterogeneous clients. Existing studies address statistical and system heterogeneity, as well as client personalization through regularization, variance correction, local normalization, and personalized representations (Yang et al., 2019; Li et al., 2020; Wang et al., 2020; Li et al., 2021b; Collins et al., 2021). Model-heterogeneous FL further supports clients with different capacities via width scaling, sub-model extraction, depth adaptation, or hypernetworks (Diao et al., 2021; Horvath et al., 2021; Shamsian et al., 2021). However, these methods often assume a shared model family, super-network, or partially aligned parameter space, making them insufficient for clients with distinct topologies, task heads, and multimodal fusion paths.

Multimodal federated learning. Multimodal federated learning (MFL) has emerged to exploit decentralized image, text, audio, and sensor data under privacy constraints (Liu et al., 2020; Zhao et al., 2022; Che et al., 2023). Existing methods primarily address modality heterogeneity, task heterogeneity, cross-modal alignment, or missing modalities. For instance, CreamFL (Yu et al., 2023) performs representation-level distillation and contrastive aggregation on public data; HAMFL (Qi & Li, 2024) models inter-client relationships via adaptive hypergraphs; FedRecon (Liu et al., 2025a), PEPSY (Nguyen et al., 2025), and FedDISC (Qiu et al., 2025a) address incomplete modalities through reconstruction, reconfigurable representations, and generative recovery, respectively. Despite these advances, most methods focus on data-level modality heterogeneity rather than parameter sharing under joint task and arbitrary architecture heterogeneity. Moreover, these MFL methods typically require public data and communicates logits, representations, or prototypes, increasing communication overhead.

Hypernetworks and graph-based parameter generation. Hypernetworks generate target-model parameters conditionally, offering an implicit alternative to direct weight aggregation (Ha et al., 2017; Shamsian et al., 2021). Graph hypernetworks (GHN) further encode neural architectures as computational graphs and generate parameters for diverse topologies (Zhang et al., 2019; Knyazev et al., 2021). HAFL-GHN applies this idea to architecture-heterogeneous FL, while FedMBridge (Chen & Zhang, 2024) extends hypernetwork-based bridging to architecture-personalized MFL. In contrast, FedMAGE studies task- and architecture-heterogeneous MFL, avoids public-data-dependent distillation, and introduces Type-aware Decoding to improve operator-conditioned parameter generation.

B Additional Methodological Details

B.1 Architecture Graph Representation

For each client k, the server represents its architecture a_k as a typed directed acyclic graph $G_k = (V_k, E_k, X_k)$. Nodes represent neural operators, directed edges encode computational dependencies, and node attributes describe properties such as operator type, parameter shape, modality role, and topological position. The predefined operator vocabulary includes common primitives such as convolution, linear projection, normalization, embedding, attention, pooling, activation, addition, concatenation, and MLP/FFN blocks. Consequently, FedMAGE supports any architecture that can be decomposed into operators in this vocabulary.

B.2 Server–Client Communication Protocol

The architecture graph G_k is registered with the server before federated training and retransmitted only if the client architecture changes. At communication round t, the server generates

$$\theta_k^t = h_{\phi^t}(G_k, c_k^t) \tag{15}$$

for each selected client k. Starting from θ_k^t, client k performs local optimization and returns only the resulting parameter displacement

$$\Delta\theta_k^t = \widetilde{\theta}_k^{t,E} - \theta_k^t. \tag{16}$$

The learnable embedding c_k is maintained and updated by the server; it is not uploaded by the client.

C Datasets and Task Definitions

We evaluate FedMAGE on three benchmarks spanning cross-modal semantic matching, multimodal sentiment recognition, and appearance–geometry fusion.

C.1 Flickr30K: Bidirectional Image–Text Retrieval

Dataset. Flickr30K comprises 31,783 photographs of everyday activities, each paired with five independently crowdsourced English captions (158,915 in total) (Young et al., 2014; Plummer et al., 2015). We use the widely adopted split of 29,783 training, 1,000 validation, and 1,000 test images (Plummer et al., 2015). Accordingly, the test pool contains 1,000 images and 5,000 corresponding captions.

Task and evaluation. The task evaluates bidirectional image–text semantic alignment (Wang et al., 2016). In image-to-text retrieval, an image is used to rank all candidate captions, and a query is successful at rank K if at least one of its five reference captions appears among the top-K results. In text-to-image retrieval, a caption is used to rank all candidate images, and a query is successful when its paired image appears in the top-K list. Recall@K (R@K) is the fraction of queries that are successful at cutoff K, where $K \in \{1, 5, 10\}$. We report mean recall (mR), defined as the average of the six direction-specific recall values:

$$\mathrm{mR} = \frac{1}{6} \sum_{d \in \{\mathrm{I2T}, \mathrm{T2I}\}} \sum_{K \in \{1,5,10\}} \mathrm{R@}K_d, \tag{17}$$

Higher mR indicates better bidirectional retrieval performance.

C.2 CMU-MOSEI: Multimodal Sentiment Recognition

Dataset. CMU-MOSEI is collected from 3,228 YouTube videos and contains 23,453 opinion-bearing utterance segments from 1,000 speakers covering 250 topics (Bagher Zadeh et al., 2018). Each segment contains three coordinated temporal streams: a language transcript, visual cues such as facial expressions and gestures, and acoustic cues such as prosody and voice quality. Sentiment intensity is annotated on a seven-point scale ranging from -3 (strongly negative) to $+3$ (strongly positive). The corpus includes intensity annotations for six emotions, which are not used in this task. Following the commonly used processed benchmark protocol, we use 16,326 segments for training, 1,871 for validation, and 4,659 for testing, for a total of 22,856 segments in the processed release (Ma et al., 2021).

Task and evaluation. Given an utterance's language, visual, and acoustic streams, the model predicts its sentiment intensity. Acc-7 is the fraction of samples for which the predicted sentiment level matches the ground truth over the label set $\{-3, -2, -1, 0, +1, +2, +3\}$. Acc-2 and F1 evaluate negative-versus-positive sentiment classification after excluding neutral samples; F1 summarizes precision and recall under the same binary protocol. Acc-7 is the primary metric because it retains the strength as well as the polarity of the expressed sentiment.

C.3 NYU-Depth-v2: RGB-D Scene Recognition

Dataset. NYU-Depth-v2 consists of indoor video sequences captured using the RGB and depth cameras of a Microsoft Kinect. Its labeled subset contains 1,449 densely annotated, spatially aligned RGB–depth pairs sampled from 464 scenes across three cities. The full release also includes 407,024 unlabeled frames (Silberman et al., 2012). RGB images provide color, texture, and appearance cues, whereas metric depth maps provide complementary geometric and spatial information. We use the labeled RGB–depth pairs for scene recognition.

Task and evaluation. Following the NYU-Depth-v2 scene-recognition protocol (Gupta et al., 2013), we group its 27 scene labels into 10 categories by retaining the nine most frequent labels and merging the remainder into an *other* class. The fixed split contains 795 training images and 654 test images. Given an aligned RGB–depth pair, the model predicts one of the 10 scene categories.

Top-1 accuracy is the fraction of test samples assigned the correct label, while F1 complements it by measuring class-wise precision and recall under category imbalance. Top-1 accuracy is treated as the primary metric.

D HETEROGENEITY STRESS TEST SETUP

We vary five discrete sources of heterogeneity: the visual encoder, selected from ResNet and ViT; the text encoder, chosen from GRU and Transformer; the fusion operator, selected from concatenation and cross-attention; the architecture depth, selected from $\{3, 5, 7\}$; and the client modality-composition pattern. For each factor r, let $\mathcal{C}_r$ denote its category set and $p_{r,c}$ the fraction of clients assigned to category $c \in \mathcal{C}_r$. We quantify the diversity of factor r using a normalized discrete entropy:

$$H_r = -\frac{1}{\log |\mathcal{C}_r|} \sum_{c \in \mathcal{C}_r} p_{r,c} \log p_{r,c}. \tag{18}$$

By construction, $H_r \in [0, 1]$, with $H_r = 0$ when all clients fall into a single category and H_r approaching 1 when clients are uniformly distributed across all categories. The overall heterogeneity score is:

$$H = \frac{H_{\text{visual}} + H_{\text{text}} + H_{\text{fusion}} + H_{\text{depth}} + H_{\text{modality}}}{5}. \tag{19}$$

Thus, larger H corresponds to a federation in which clients differ more simultaneously in architecture components, depth, and modality composition, rather than in a single isolated attribute.