COUPLING PERCEPTION AND REASONING IN FEDERATED MULTIMODAL GRAPH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated multimodal graph foundation models (GFMs) aim to adapt pretrained multimodal models to decentralized graph data, where each client owns a private multimodal graph and cannot share raw information. These models typically combine a multimodal Encoder that extracts semantic evidence from heterogeneous modalities and a graph neural network (GNN) that performs relational reasoning over graph structures. However, existing federated GFM adaptation methods mainly update graph-side modules while keeping the multimodal Encoder frozen, limiting adaptation to *how information is propagated* while fixing *what information is extracted*. Through empirical studies, we reveal that Encoder and GNN adaptations are not independent: Encoder adaptation is affected by graph relations, while cross-client module swapping reveals substantial pairing sensitivity between separately parameterized Encoder and GNN updates. Motivated by this observation, we propose **FedCORE**, a federated adaptation framework that represents Encoder and GNN updates through a shared low-dimensional latent state. FedCORE jointly optimizes this core from multimodal and structural signals and performs federated evolution directly in the shared state space, preserving compatibility between perception and reasoning adaptations. Extensive experiments demonstrate that FedCORE reduces the Encoder–GNN pairing gap from 30.6 to 5.9, corresponding to an 80.7% reduction over independent joint adaptation.

1 INTRODUCTION

Multimodal graph foundation models (GFMs) have emerged as a paradigm for jointly modeling multimodal semantics and graph relations. Combining pretrained multimodal encoders with graph neural networks (GNNs), multimodal GFMs enable unified perception and relational reasoning across diverse graph-based applications (He et al., 2025; Liu et al., 2026), including e-commerce recommendation (Liu et al., 2021), social and content platforms (Wei et al., 2019). However, in practice, these graphs are distributed across organizations or platforms and cannot be centralized because of privacy, ownership, and communication constraints. Federated learning therefore provides a practical paradigm for collaboratively adapting multimodal GFMs without sharing raw data.

A natural solution jointly adapts the Encoder and GNN on each client, allowing multimodal perception and graph reasoning to improve jointly on local tasks. During training, each client updates both modules using its multimodal graph and downstream objective, while federated optimization typically aggregates their updates independently. Such module-wise aggregation fails to preserve the relationship between Encoder and GNN updates established during training, weakening coordination between multimodal perception and graph reasoning after federation. Thus, the key challenge is not only enabling Encoder–GNN joint adaptation but also preserving their coordination during federated optimization and aggregation.

To understand this coordination, our empirical studies in Sec. 3 examine three questions: ❶ **Encoder Adaptation.** We first test whether stronger graph reasoning can compensate for a fixed multimodal Encoder. Figure 1 shows that increasing GNN capacity cannot consistently compensate for the absence of Encoder adaptation, demonstrating that graph reasoning alone cannot recover the missing task-relevant multimodal information and that perception adaptation is necessary. ❷ **Graph Dependency.** Given its necessity, we further study whether Encoder adaptation can be optimized independently from graph structures. Figure 2 shows that controlled changes in graph relations systemati-

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.

cally alter Encoder updates even when multimodal contents and labels remain unchanged, revealing that Encoder adaptation depends on the relational environment where representations are utilized. ❸ **Update Coupling.** Since Encoder adaptation is affected by graph relations, we finally examine whether locally co-adapted Encoder and GNN updates are freely interchangeable across clients. Figure 3 shows that locally matched Encoder–GNN pairs outperform cross-client mismatched combinations on average, revealing substantial pairing sensitivity between the two modules. These observations motivate explicitly coupling Encoder and GNN adaptations through a shared state during local optimization, rather than parameterizing their updates separately.

Based on these observations, we propose FedCORE, a federated multimodal graph learning framework that represents coupled multimodal perception and graph reasoning adaptation through a shared latent state. Rather than independently maintaining Encoder and GNN adapters, FedCORE learns a compact core jointly optimized on each client and decoded into module-specific low-rank updates for multimodal Encoders, fusion layers, and GNNs. The shared core preserves the relationship between extracted multimodal evidence and its propagation through graph structures. During federation, only the shared core is communicated and aggregated, maintaining learned perception–reasoning coordination across clients while retaining communication efficiency.

Our Contributions. (1) ***New Insight.*** We reveal that Encoder and GNN adaptations in federated multimodal GFMs form a coupled optimization process rather than independent modules, motivating coordinated perception–reasoning adaptation. (2) ***New Framework.*** We propose FedCORE, which represents Encoder and GNN adaptations through a shared low-dimensional latent state, enabling coordinated optimization and aggregation-consistent federation with low communication overhead. (3) ***SOTA Performance.*** Experiments across multimodal graph benchmarks and downstream tasks show that FedCORE consistently outperforms strong federated baselines, achieving an average improvement of **2.27%** over the strongest non-FedCORE baseline.

2 Related Work

2.1 Multimodal Graph Foundation Models

Multimodal graph foundation models combine pretrained multimodal encoders with graph neural networks to jointly capture semantic information and relational structures (He et al., 2025; Fang et al., 2025). Recent studies explore cross-modal interaction, topology-aware representation learning, and structural grounding in multimodal graphs (Ning et al., 2025; Liu et al., 2026). These methods validate combining multimodal perception with graph reasoning but primarily consider centralized training, where components are jointly optimized with complete data. We instead study adapting and coordinating such multimodal graph models under federated settings.

2.2 Federated Adaptation of Graph Foundation Models

Federated graph foundation models transfer pretrained graph learning capabilities across decentralized clients while preserving data locality. Recent studies explore parameter-efficient adaptation and shared knowledge modeling for improved cross-client transferability (Zhu et al., 2026a; Wu et al., 2025; Zhu et al., 2026b; Chen et al., 2026b). Meanwhile, federated multimodal graph learning considers challenges from heterogeneous modalities and decentralized graph structures, with recent methods developing adaptive strategies for multimodal federated environments (Li et al., 2026a; Zhang et al., 2026; Chen et al., 2026a;c; Zhu et al., 2026c; Li et al., 2026b). However, existing approaches generally treat multimodal representation learning and graph reasoning as independent adaptation components, overlooking that updates can become coupled during local optimization and lose coordination after federated aggregation. FedCORE addresses this limitation by modeling Encoder and GNN adaptations through a shared latent state that preserves their perception–reasoning coordination throughout federated learning.

3 Empirical Study

We first diagnose why independently adapting and aggregating the Encoder and GNN may be insufficient in federated multimodal GFMs. Our studies answer three questions: whether Encoder

adaptation is necessary beyond graph-side optimization, whether Encoder adaptation depends on graph relations, and whether Encoder–GNN updates should be coordinated during federation. Detailed protocols and metric definitions are provided in Appendix A.3.

3.1 Can Stronger GNNs Replace Encoder Adaptation?

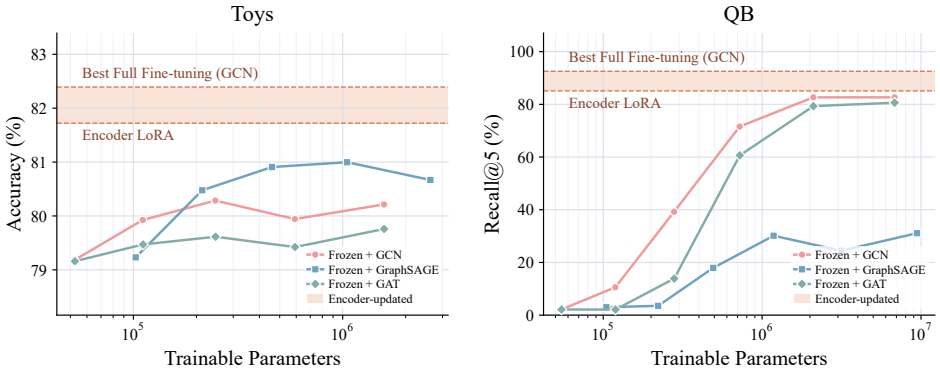


Figure 1: Observation 1: Stronger GNNs cannot replace Encoder adaptation. Performance of graph-only adaptation with increasing GNN capacity compared with Encoder-adapted models.

Observation 1: *Graph-side adaptation improves relational reasoning but cannot replace task-specific Encoder adaptation.* We first examine whether stronger graph reasoning can compensate for a frozen multimodal Encoder. In Fig. 1, we gradually increase the trainable capacity of GCN, GraphSAGE, and GAT while keeping the Encoder fixed, and compare them with Encoder LoRA adaptation and full fine-tuning. Although larger GNN capacity consistently improves graph-only adaptation, it remains inferior to Encoder-adapted models. This indicates that graph reasoning cannot recover multimodal information that is not sufficiently captured by the frozen Encoder.

3.2 Does Encoder Adaptation Depend on Relational Context?

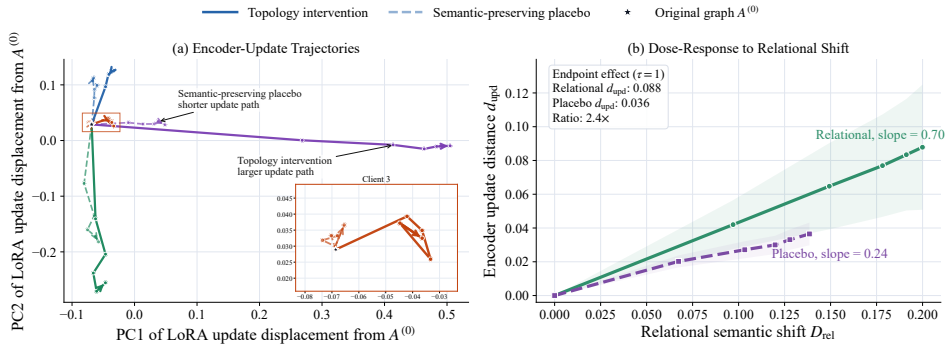


Figure 2: Observation 2: Relational context conditions multimodal representation adaptation. Encoder-LoRA update trajectories under graph interventions and semantic-preserving perturbations.

Observation 2: *Multimodal representation adaptation is relationally conditioned rather than determined solely by modality contents.* Given the necessity of Encoder adaptation, we further investigate whether it can be optimized independently from graph structures. We keep node features, labels, and splits unchanged while intervening only on graph topology. As shown in Fig. 2, stronger relational interventions induce increasingly different Encoder-LoRA update trajectories, whereas semantic-preserving perturbations introduce much smaller changes. The mean update distance reaches 0.088 under the strongest intervention compared with 0.036 under the low-semantic-shift control, demonstrating that Encoder adaptation is conditioned by the relational environment where representations are utilized.

3.3 Are Encoder and GNN Adaptations Independent?

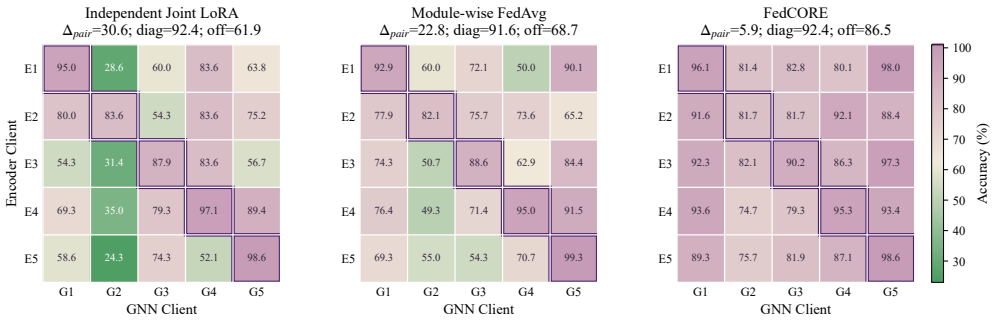


Figure 3: Observation 3: Independently learned module updates can exhibit cross-client combination incompatibility. Cross-client Encoder–GNN pairing results on Toys.

Observation 3. *Independently learned Encoder and GNN updates can exhibit cross-client combination incompatibility.* We examine whether independently learned Encoder and GNN updates remain compatible across clients. We train local Encoder–GNN pairs and evaluate cross-client combinations by replacing one module while fixing the other. As shown in Fig. 3, matched Encoder–GNN pairs consistently outperform mismatched combinations. Independent Joint LoRA suffers a pairing gap of 30.6, while module-wise aggregation reduces it to 22.8. FedCORE exhibits a substantially smaller pairing gap, demonstrating the importance of preserving Encoder–GNN coordination during federation.

4 Methodology

4.1 Overview

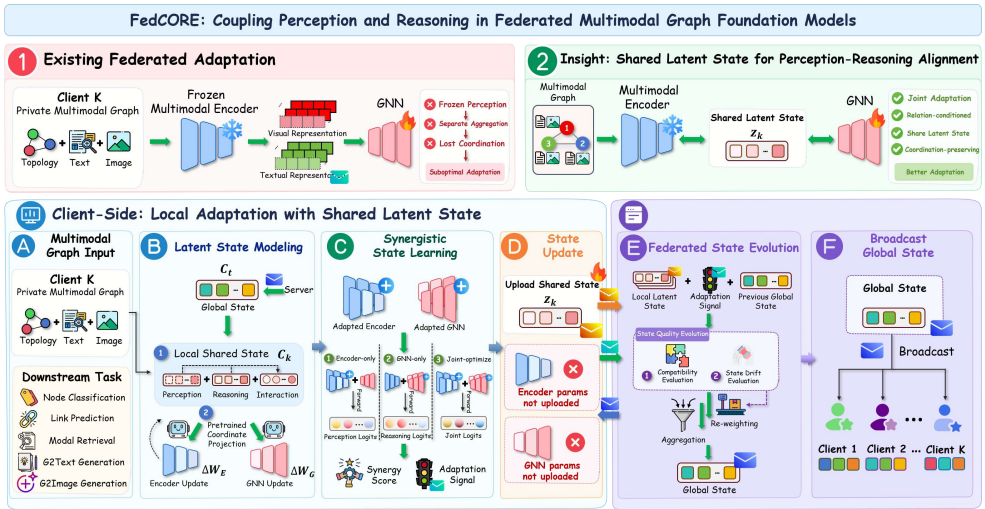


Figure 4: **Overview of FedCORE.**

We present FedCORE, a latent-state adaptation framework for federated multimodal graph foundation models. As shown in Fig. 4, FedCORE has three components: (1) *Latent State Modeling*, representing heterogeneous Encoder–GNN adaptations via a shared state; (2) *Synergistic State Learning*, optimizing this state with task supervision and perception–reasoning synergy; and (3) *Federated State Evolution*, evolving local states across clients while preserving their coordination. Problem formulation appears in Appendix A.1.

4.2 LATENT STATE MODELING

Motivation. Obs. 2 and Obs. 3 show that Encoder and GNN adaptations are coupled: multimodal adaptation depends on relational context, while independently learned module updates may become incompatible when combined across clients. FedCORE therefore models both through a shared latent state.

Shared Latent State. Let $\mathcal{J} = \mathcal{J}_P \cup \mathcal{J}_R$ denote the adaptable parameter matrices, where $\mathcal{J}_P$ contains Encoder and fusion layers and $\mathcal{J}_R$ contains GNN layers. Each client k maintains a matrix-valued state $\mathbf{C}_k \in \mathbb{R}^{r\times r}$, where r is the latent rank. Rather than assigning independent variables to different modules, $\mathbf{C}_k$ provides their common adaptation coordinate, implicitly encoding perception, reasoning, and their interaction.

Pretrained Coordinate Construction. Because modules in $\mathcal{J}$ have heterogeneous dimensions, FedCORE constructs a fixed coordinate frame for each pretrained matrix $\mathbf{W}_j^0 \in \mathbb{R}^{d_j^{\text{out}}\times d_j^{\text{in}}}$ using its rank-r truncated singular value decomposition:

$$\mathbf{W}_j^0 \approx \mathbf{U}_j\mathbf{\Sigma}_j\mathbf{V}_j^\top, \quad \mathbf{S}_j = \left(\frac{\mathbf{\Sigma}_j + \epsilon\mathbf{I}_r}{\text{tr}(\mathbf{\Sigma}_j)/r + \epsilon}\right)^{1/4}, \quad \mathbf{U}_j^\top\mathbf{U}_j = \mathbf{V}_j^\top\mathbf{V}_j = \mathbf{I}_r. \tag{1}$$

Here, $\mathbf{U}_j \in \mathbb{R}^{d_j^{\text{out}}\times r}$, $\mathbf{V}_j \in \mathbb{R}^{d_j^{\text{in}}\times r}$, and $\mathbf{\Sigma}_j, \mathbf{S}_j \in \mathbb{R}^{r\times r}$. The constant $\epsilon > 0$ ensures numerical stability. These coordinates are computed once from the pretrained model and remain frozen, leaving $\mathbf{C}_k$ as the client-specific adaptation variable.

State-conditioned Adaptation Realization. FedCORE maps the shared state to each module through a fixed linear decoder $\mathcal{D}_j : \mathbb{R}^{r\times r} \to \mathbb{R}^{d_j^{\text{out}}\times d_j^{\text{in}}}$:

$$\Delta\mathbf{W}_{k,j} = \mathcal{D}_j(\mathbf{C}_k) = \alpha_j\mathbf{U}_j\mathbf{S}_j\mathbf{C}_k\mathbf{S}_j\mathbf{V}_j^\top, \quad \alpha_j = \frac{\|\mathbf{W}_j^0\|_F}{\sqrt{d_j^{\text{out}}d_j^{\text{in}}} + \epsilon}. \tag{2}$$

Here, α_j compensates for scale differences across layers, and the adapted parameter is $\mathbf{W}_{k,j} = \mathbf{W}_j^0 + \Delta\mathbf{W}_{k,j}$. Thus, heterogeneous Encoder and GNN updates retain their native parameter forms while sharing the same latent origin.

Canonical State Recovery. The linear decoder admits an exact projection from each module space back to the shared state:

$$\mathcal{P}_j(\Delta\mathbf{W}) = \alpha_j^{-1}\mathbf{S}_j^{-1}\mathbf{U}_j^\top\Delta\mathbf{W}\mathbf{V}_j\mathbf{S}_j^{-1}, \quad \mathcal{P}_j(\mathcal{D}_j(\mathbf{C}_k)) = \mathbf{C}_k, \quad \forall j \in \mathcal{J}. \tag{3}$$

Hence, $\mathcal{P}_i(\Delta\mathbf{W}_{k,i}) = \mathcal{P}_j(\Delta\mathbf{W}_{k,j}) = \mathbf{C}_k$ for any $i, j \in \mathcal{J}$, ensuring a common latent origin by construction.

4.3 SYNERGISTIC STATE LEARNING

Motivation. The shared state defines how heterogeneous adaptations are represented, but must be optimized for the local task. FedCORE therefore learns $\mathbf{C}_k$ using downstream supervision together with perception–reasoning synergy.

State-guided Local Adaptation. At round t, the server broadcasts the global state $\mathbf{C}^t$, and each participating client initializes $\mathbf{C}_k^{t,0} = \mathbf{C}^t$. Let $\mathcal{D}_P(\mathbf{C}_k)$ and $\mathcal{D}_R(\mathbf{C}_k)$ denote the decoded updates for modules in $\mathcal{J}_P$ and $\mathcal{J}_R$, respectively. Given local features $\mathbf{X}_k$, graph structure $\mathbf{A}_k$, labels $\mathbf{Y}_k$, and task head H_{Ω_k}, the joint prediction is

$$\widehat{\mathbf{Y}}_k^{\text{joint}} = H_{\Omega_k}\left(G_{\Phi_0+\mathcal{D}_R(\mathbf{C}_k)}\left(E_{\Theta_0+\mathcal{D}_P(\mathbf{C}_k)}(\mathbf{X}_k), \mathbf{A}_k\right)\right), \tag{4}$$

where Θ_0 and Φ_0 denote the pretrained Encoder-side and GNN-side parameters. Local adaptation is therefore entirely governed by the shared state rather than independent module adapters.

Perception–Reasoning Synergy Learning. To evaluate whether joint adaptation provides additional benefit, we construct perception-only and reasoning-only counterfactual forward passes:

$$\widehat{\mathbf{Y}}_k^{\,p} = H_{\Omega_k}\left(G_{\Phi_0}\left(E_{\Theta_0+\mathcal{D}_P(\mathbf{C}_k)}(\mathbf{X}_k), \mathbf{A}_k\right)\right), \widehat{\mathbf{Y}}_k^{\,r} = H_{\Omega_k}\left(G_{\Phi_0+\mathcal{D}_R(\mathbf{C}_k)}\left(E_{\Theta_0}(\mathbf{X}_k), \mathbf{A}_k\right)\right). \quad (5)$$

Let $\ell(\widehat{\mathbf{Y}}, \mathbf{Y})$ denote the task loss, yielding ℓ_k^{joint}, ℓ_k^p, and ℓ_k^r for the three predictions. To prevent synergy from being artificially increased by degrading isolated branches, FedCORE treats their losses as detached references. With $\text{sg}(\cdot)$ denoting stop-gradient and $[x]_+ = \max(x, 0)$,

$$b_k = \text{sg}\left(\min\{\ell_k^p, \ell_k^r\}\right), \quad \mathcal{L}_k^{\text{syn}} = \left[m + \ell_k^{\text{joint}} - b_k\right]_+, \quad \mathcal{L}_k = \ell_k^{\text{joint}} + \lambda_c \mathcal{L}_k^{\text{syn}} + \frac{\lambda_r}{2}\|\mathbf{C}_k\|_F^2, \quad (6)$$

where $m \geq 0$ is the synergy margin and $\lambda_c, \lambda_r \geq 0$ control synergy and state regularization. This objective encourages joint adaptation to outperform the stronger isolated branch without degrading either reference branch.

Latent Signal Integration. Since module gradients lie in heterogeneous parameter spaces, FedCORE maps them into the shared state space. For module j, let $\mathbf{G}_{k,j}^W = \nabla_{\mathbf{W}_{k,j}} \mathcal{L}_k$. The adjoint of the decoder gives

$$\mathbf{g}_{k,j} = \mathcal{D}_j^*\left(\mathbf{G}_{k,j}^W\right) = \alpha_j \mathbf{S}_j \mathbf{U}_j^\top \mathbf{G}_{k,j}^W \mathbf{V}_j \mathbf{S}_j, \quad \mathbf{g}_k^p = \sum_{j \in \mathcal{J}_P} \mathbf{g}_{k,j}, \quad \mathbf{g}_k^r = \sum_{j \in \mathcal{J}_R} \mathbf{g}_{k,j}. \quad (7)$$

The resulting state gradient and local update are

$$\mathbf{g}_k^e = \mathbf{g}_k^{p,e} + \mathbf{g}_k^{r,e} + \lambda_r \mathbf{C}_k^e, \qquad \mathbf{C}_k^{t,e+1} = \mathbf{C}_k^{t,e} - \eta \mathbf{P}_{L,k}^e \mathbf{g}_k^e \mathbf{P}_{R,k}^e. \quad (8)$$

where $e \in \{0, \ldots, E-1\}$ indexes the E local steps and $\eta > 0$ is the learning rate. Perception and reasoning signals thus jointly update a single client state. The preconditioners and their local second-moment statistics are defined in Appendix A.4.

4.4 Federated State Evolution

Motivation. Local training produces client-specific states encoding coordinated perception–reasoning adaptation. FedCORE evolves the shared states directly, thereby enforcing a common parameterization for the decoded Encoder and GNN updates.

State Quality-aware Evolution. After E local steps, client k obtains $\mathbf{C}_k^{t,E}$. We summarize its perception–reasoning coordination by averaging the latent signals along the local trajectory:

$$\overline{\mathbf{g}}_k^p = \frac{1}{E}\sum_{e=0}^{E-1} \mathbf{g}_{k,e}^p, \quad \overline{\mathbf{g}}_k^r = \frac{1}{E}\sum_{e=0}^{E-1} \mathbf{g}_{k,e}^r, \quad \Gamma_k = \frac{\langle \overline{\mathbf{g}}_k^p, \overline{\mathbf{g}}_k^r \rangle_F}{\|\overline{\mathbf{g}}_k^p\|_F \|\overline{\mathbf{g}}_k^r\|_F + \epsilon}. \quad (8)$$

Here, $\langle \mathbf{A}, \mathbf{B} \rangle_F = \text{tr}(\mathbf{A}^\top \mathbf{B})$ is the Frobenius inner product, and larger Γ_k indicates stronger perception–reasoning agreement. Each client uploads only $\mathbf{C}_k^{t,E}$ and the scalar Γ_k; module parameters and gradients remain local.

The server combines this coordination score with normalized state drift to determine the evolution weight:

$$D_k^t = \frac{\|\mathbf{C}_k^{t,E} - \mathbf{C}^t\|_F^2}{\|\mathbf{C}^t\|_F^2 + \epsilon}, \qquad \omega_k^t = \frac{p_k^t \exp(\tau\Gamma_k - \xi D_k^t)}{\sum_{i \in \mathcal{S}_t} p_i^t \exp(\tau\Gamma_i - \xi D_i^t)}, \quad p_k^t = \frac{n_k}{\sum_{i \in \mathcal{S}_t} n_i}, \quad (9)$$

where $\mathcal{S}_t$ denotes participating clients, n_k is the number of local training instances, and $\tau, \xi \geq 0$ control coordination and drift. This weighting favors coordinated states while suppressing excessive deviation from the previous global state.

Latent State Evolution. FedCORE combines the quality-aware client states with the previous global state:

$$\mathbf{C}^{t+1} = (1-\mu)\sum_{k\in\mathcal{S}_t}\omega_k^t\mathbf{C}_k^{t,E}+\mu\mathbf{C}^t, \qquad \mathcal{D}_j(\mathbf{C}^{t+1}) = (1-\mu)\sum_{k\in\mathcal{S}_t}\omega_k^t\mathcal{D}_j(\mathbf{C}_k^{t,E})+\mu\mathcal{D}_j(\mathbf{C}^t), \quad (10)$$

where $\mu \in [0, 1)$ controls temporal persistence. By linearity of $\mathcal{D}_j$, state evolution remains closed under the coupled adaptation family, so all global module updates continue to originate from the same $\mathbf{C}^{t+1}$. The updated state is then broadcast for the next communication round.

5 Experiments

We conduct comprehensive experiments to evaluate FedCORE and address four research questions. **Q1: Overall Performance.** Does FedCORE consistently outperform existing federated multimodal graph learning methods across downstream tasks? **Q2: Ablation Study.** How does each design contribute to FedCORE's effectiveness? **Q3: Robustness.** Does FedCORE remain effective under stronger client heterogeneity and structural perturbations? **Q4: Efficiency and Scalability.** Can FedCORE achieve these gains with reasonable communication, computation, and memory overhead?

5.1 Experimental Setup

We evaluate FedCORE on multimodal graph benchmarks spanning node classification, link prediction, multimodal retrieval, graph-to-text generation, and graph-to-image generation. Toys and Grocery derive from Amazon review data, whereas Bili Music, DY, QB, and Bili Cartoon come from NineRec (Zhang et al., 2024). Following MM-OpenFGL (Li et al., 2026a), we compare FedCORE with representative federated, federated graph foundation, and multimodal federated graph methods, including FedAvg (McMahan et al., 2017), FedGFM+ (Zhu et al., 2026a), STAGE (Chen et al., 2026a), FedGAMMA (Li et al., 2026b), and federated PLANET (Liu et al., 2026), plus two adaptation controls: Frozen Encoder + GNN and Independent Joint LoRA (Hu et al., 2021). All methods share client partitions, pretrained backbones, and evaluation protocols. Detailed settings appear in Appendix A.5.

5.2 Overall Performance (Q1)

Table 1: Overall performance comparison. Best results are in **bold** and second-best are underlined.

Method	Node Classification (Acc)		Link Prediction (AUC)		Modal Retrieval (R@5)		G2Text (ROUGE-L)	G2Image (CLIP-S)
	Toys	Grocery	Bili Music	DY	QB	Bili Cartoon	Flickr30k	SemArt
FedGAMMA	$76.37_{\pm 0.65}$	$73.73_{\pm 0.54}$	$75.87_{\pm 0.91}$	$72.25_{\pm 0.26}$	$86.52_{\pm 2.44}$	$80.22_{\pm 0.78}$	$31.29_{\pm 0.09}$	$70.87_{\pm 0.23}$
STAGE	$79.19_{\pm 0.02}$	$81.04_{\pm 0.42}$	$69.41_{\pm 1.74}$	$69.19_{\pm 0.49}$	$\underline{94.91}_{\pm 3.65}$	$79.77_{\pm 1.39}$	$30.87_{\pm 0.47}$	$\underline{71.86}_{\pm 0.53}$
FedGFM+	$80.21_{\pm 0.28}$	$83.76_{\pm 0.15}$	$\underline{76.24}_{\pm 0.02}$	$74.30_{\pm 0.11}$	$92.06_{\pm 0.83}$	$\underline{84.57}_{\pm 4.38}$	$31.09_{\pm 0.31}$	$71.58_{\pm 0.26}$
Fed-PLANET	$\underline{80.58}_{\pm 0.07}$	$\underline{83.93}_{\pm 0.48}$	$65.29_{\pm 0.90}$	$64.08_{\pm 0.70}$	$89.89_{\pm 2.64}$	$81.20_{\pm 5.54}$	$\underline{31.35}_{\pm 0.23}$	$71.38_{\pm 0.21}$
Frozen Enc-GNN	$78.93_{\pm 0.14}$	$80.63_{\pm 0.08}$	$74.50_{\pm 0.70}$	$72.60_{\pm 0.21}$	$86.45_{\pm 2.56}$	$76.66_{\pm 5.00}$	$31.07_{\pm 0.35}$	$70.55_{\pm 0.09}$
Joint LoRA	$78.72_{\pm 0.04}$	$80.59_{\pm 0.55}$	$73.64_{\pm 0.69}$	$72.41_{\pm 1.64}$	$80.49_{\pm 1.95}$	$81.49_{\pm 2.27}$	$30.53_{\pm 0.07}$	$70.21_{\pm 0.12}$
Ours	$\mathbf{81.72}_{\pm 0.10}$	$\mathbf{84.68}_{\pm 0.18}$	$\mathbf{79.50}_{\pm 0.94}$	$\mathbf{77.69}_{\pm 0.32}$	$\mathbf{95.05}_{\pm 1.61}$	$\mathbf{90.73}_{\pm 2.34}$	$\mathbf{33.31}_{\pm 0.59}$	$\mathbf{73.24}_{\pm 0.27}$

For Q1, we compare FedCORE with representative federated graph, multimodal graph, and GFM baselines across diverse tasks. Table 1 shows that FedCORE consistently performs best, exceeding the strongest non-FedCORE baseline by 2.27% on average and up to 6.16% on Bili Cartoon. FedCORE also surpasses Independent Joint LoRA by 5.98% on average. These results show that the complete FedCORE parameterization outperforms independently maintained Encoder and GNN adapters in our implementation.

5.3 Ablation Study (Q2)

To answer Q2, we ablate FedCORE's key designs in Table 2. We replace the shared latent state with independent Encoder–GNN adapters, remove the fixed SVD bases, replace the geometry-balanced

Table 2: Ablation study. Best results are in **bold** and second-best results are underlined.

Variant	Node Classification (Acc)	Link Prediction (AUC)	Modal Retrieval (R@5)	G2Text (ROUGE-L)	G2Image (CLIP-S)
	Toys	DY	QB	Flickr30k	SemArt
Full FedCORE	**81.67**$_{\pm 0.05}$	**77.72**$_{\pm 0.32}$	**95.28**$_{\pm 0.61}$	**33.21**$_{\pm 0.76}$	**73.26**$_{\pm 0.21}$
w/o Shared Latent State	79.96$_{\pm 0.21}$	74.86$_{\pm 0.43}$	92.94$_{\pm 0.46}$	30.71$_{\pm 0.94}$	71.75$_{\pm 0.36}$
w/o SVD Bases	80.87$_{\pm 0.12}$	75.93$_{\pm 0.32}$	94.18$_{\pm 0.73}$	32.48$_{\pm 0.85}$	72.59$_{\pm 0.29}$
w/o Geometry-Balanced Update	81.24$_{\pm 0.08}$	76.31$_{\pm 0.29}$	94.76$_{\pm 0.66}$	32.91$_{\pm 0.75}$	72.94$_{\pm 0.24}$
Indep. Adapters + FedAvg	80.51$_{\pm 0.06}$	76.66$_{\pm 0.27}$	95.05$_{\pm 0.15}$	32.19$_{\pm 0.40}$	72.24$_{\pm 0.21}$

update with ordinary gradient descent, and replace shared core aggregation with independent adapter aggregation. These variants assess the contributions of shared adaptation representation, coordinate transport, geometry-balanced optimization, and aggregation-consistent federation.

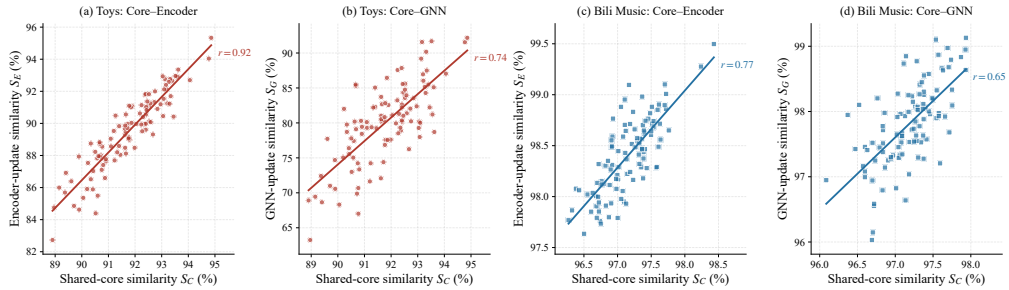
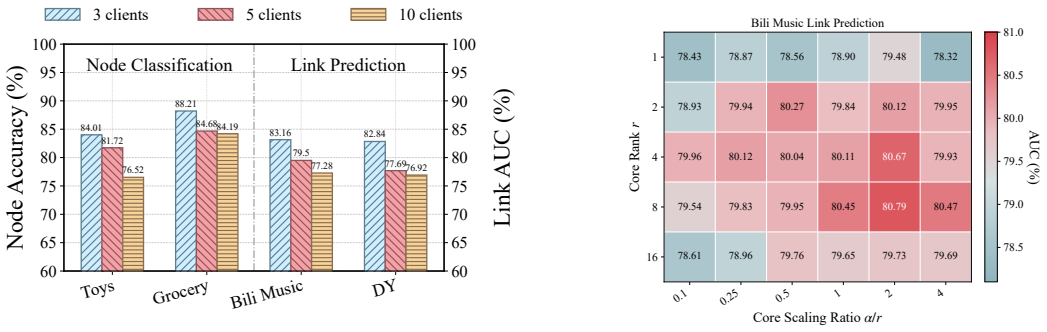


Figure 5: **Relational core–update correspondence.** Each point is one client pair. The x-axis shows shared-core similarity, and the y-axis shows Encoder/GNN update similarity.

Building on these ablations, we further test whether the learned shared core captures a common perception–reasoning adaptation state. Fig. 5 measures the relationship between pairwise shared-core similarity and decoded Encoder and GNN update similarity on Toys and Bili Music. Positive correlations indicate that clients with similar shared cores tend to receive similar perception and reasoning adaptations.

5.4 Robustness Analysis (Q3)

To answer Q3, we evaluate FedCORE under stronger client heterogeneity, graph fragmentation, hyperparameter variations, and imperfect multimodal inputs.



(a) Client fragmentation.

(b) Hyperparameter sensitivity.

Figure 6: **Robustness to client fragmentation and hyperparameter choices.**

Scalability to Client Fragmentation. We vary the number of clients as $K \in \{3, 5, 10\}$ while keeping the complete dataset fixed. Larger K leads to smaller and more fragmented local graphs, making local adaptation and global coordination more difficult. Fig. 6(a) reports the performance of FedCORE on representative node classification and link prediction tasks.

Hyperparameter Sensitivity. We study the latent state rank r and the core scaling ratio α/r, which control the capacity and magnitude of the shared latent state. Fig. 6(b) presents the sensitivity heatmap on Bili Music link prediction. The broad high-performing region indicates that FedCORE is not sensitive to precise hyperparameter choices.

Robustness to Modality Corruption and Missing Modalities. We further evaluate robustness to degraded modality information on Grocery node classification. Fig. 7 reports performance as the corruption and missing-modality ratios increase, examining whether coupled Encoder–GNN adaptation can better exploit the remaining semantic and relational information.

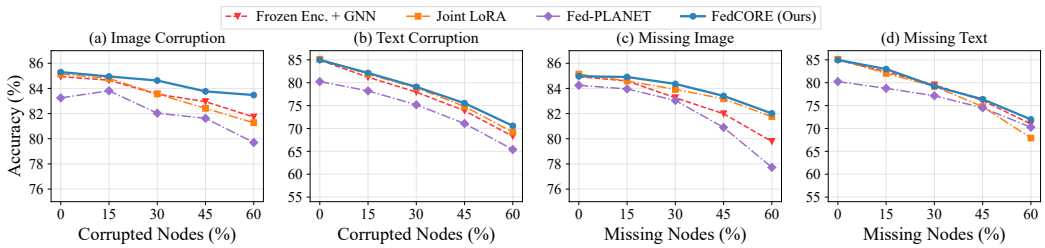


Figure 7: **Robustness to modality corruption and missing modalities.**

We also evaluate FedCORE under imperfect multimodal inputs and client heterogeneity. Additional results on controlled feature and topology heterogeneity are provided in Appendix C.

5.5 Efficiency and Scalability Analysis (Q4)

To answer Q4, we evaluate whether FedCORE achieves coupled adaptation with reasonable system overhead and remains scalable as more model components are adapted.

Communication and Resource Efficiency. Table 3 compares FedCORE with representative baselines in trainable parameters, communication payload, FLOPs, GPU memory, and training time. We count one upload and one download of federated adaptation parameters per participating client per round. FedCORE communicates 5.12×10^2 scalars per round, compared with 4.64×10^5 scalars for Independent Joint LoRA, a 99.89% reduction in per-round communication. Appendix B reports additional convergence and adaptation-scale analyses.

Table 3: Communication and resource efficiency of FedCORE and baseline methods.

Method	Trainable Params.	Comm. Payload (Scalars/Round)	Total Ops (FLOPs)	Space Cost (MB)	Time (s/round)
FedGAMMA	2.32×10^5	4.64×10^5	2.69×10^{11}	**242.0**	2.933
STAGE	2.49×10^5	4.98×10^5	$\mathbf{2.69 \times 10^{10}}$	469.6	1.795
Fed-PLANET	8.43×10^5	1.686×10^6	2.73×10^{11}	528.0	2.088
Frozen Encoder + Trainable GNN + FedAvg	2.49×10^5	4.98×10^5	$\mathbf{2.69 \times 10^{10}}$	443.6	**0.676**
Independent Joint LoRA + FedAvg	2.32×10^5	4.64×10^5	2.90×10^{10}	438.8	0.752
FedCORE (Ours)	$\mathbf{2.56 \times 10^2}$	$\mathbf{5.12 \times 10^2}$	3.12×10^{10}	441.3	1.155

6 Conclusion

In this work, we revisit federated adaptation of multimodal graph foundation models and show that Encoder adaptation is necessary, sensitive to graph relations, and coupled with GNN adaptation. Based on these findings, we propose FedCORE, which models perception and graph reasoning through a shared latent state and preserves their coordination during local optimization and federated aggregation. Experiments across diverse multimodal graph tasks demonstrate that FedCORE consistently outperforms frozen-Encoder and independent adaptation strategies, highlighting the importance of coordinated perception–reasoning adaptation in federated multimodal GFMs.

AI USE STATEMENT

Generative AI tools were used during the preparation of this manuscript for limited purposes, including language refinement, literature retrieval and discovery, and research ideation support. Specifically, AI tools assisted with improving writing clarity, organizing related work, exploring potential research directions, and discussing possible methodological designs and experimental considerations.

REPRODUCIBILITY STATEMENT

We provide an anonymized reference implementation, experiment configurations, and reproducibility checks at `https://anonymous.4open.science/r/FedCORE-A34E/`. The main paper describes the empirical diagnostic protocols, the FedCORE parameterization and optimization procedure, and the evaluation settings and results. The appendix provides additional protocol details, theoretical derivations, robustness analyses, and implementation settings. The released code excludes datasets and pretrained checkpoints, but documents the required data format, external model dependencies, and instructions for reproducing the reported experimental settings.

REFERENCES

Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In *International conference on machine learning*, pp. 1396–1415. PMLR, 2023.

Zekai Chen, Xun Wu, Xunkai Li, Yihan Sun, Rong-Hua Li, and Guoren Wang. Stage: Tackling semantic drift in multimodal federated graph learning. *arXiv preprint arXiv:2605.11919*, 2026a.

Zekai Chen, Kairui Yang, Xuaner Chen, Xunkai Li, Xun Wu, Rong-Hua Li, and Guoren Wang. Fedlab: Traceable semantic codebooks for federated multimodal graph foundation learning. *arXiv preprint arXiv:2606.32016*, 2026b.

Zekai Chen, Miao Zhang, Jiayang Xing, Xunkai Li, Xun Wu, Rong-Hua Li, and Guoren Wang. Prism: Topology-aware cross-modal imputation for modality-deficient federated graph learning. *arXiv preprint arXiv:2606.09301*, 2026c.

Yi Fang, Bowen Jin, Jiacheng Shen, Sirui Ding, Qiaoyu Tan, and Jiawei Han. Graphgpt-o: Synergistic multimodal comprehension and generation on graphs. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19467–19476. IEEE, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Yufei He, Yuan Sui, Xiaoxin He, Yue Liu, Yifei Sun, and Bryan Hooi. Unigraph2: Learning a unified embedding space to bind multimodal graphs. In *Proceedings of the ACM on Web Conference 2025*, pp. 1759–1770, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Xunkai Li, Yuming Ai, Yinlin Zhu, Haodong Lu, Yi Zhang, Guohao Fu, Bowen Fan, Qiangqiang Dai, Rong-Hua Li, and Guoren Wang. Mm-openfgl: A comprehensive benchmark for multimodal federated graph learning. *arXiv preprint arXiv:2601.22416*, 2026a.

Xunkai Li, Guohao Fu, Yuming Ai, Zhengyu Wu, Hongchao Qin, Rong-Hua Li, and Guoren Wang. Toward federated multimodal graph foundation models: A topology-aware multimodal alignment framework. *arXiv preprint arXiv:2607.15687*, 2026b.

Sicheng Liu, Xunkai Li, Daohan Su, Ru Zhang, Hongchao Qin, Ronghua Li, and Guoren Wang. Toward effective multimodal graph foundation model: A divide-and-conquer based approach. *arXiv preprint arXiv:2602.04116*, 2026.

Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. Pre-training graph transformer with multimodal side information for recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2853–2861, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Xuying Ning, Dongqi Fu, Tianxin Wei, Wujiang Xu, and Jingrui He. Graph4mm: Weaving multimodal learning with structural information. *arXiv preprint arXiv:2510.16990*, 2025.

Zhen Wang, Weirui Kuang, Yuexiang Xie, Liuyi Yao, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-gnn: Towards a unified, comprehensive and efficient package for federated graph learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4110–4120, 2022.

Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 1437–1445, 2019. doi: 10.1145/3343031.3351034.

Zhengyu Wu, Yinlin Zhu, Xunkai Li, Ziang Qiu, Rong-Hua Li, Guoren Wang, and Chenghu Zhou. Fedbook: A unified federated graph foundation codebook with intra-domain and inter-domain knowledge modeling. *arXiv preprint arXiv:2510.07755*, 2025.

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-iid graphs. *Advances in neural information processing systems*, 34:18839–18852, 2021.

Jiaqi Zhang, Yu Cheng, Yongxin Ni, Yunzhu Pan, Zheng Yuan, Junchen Fu, Youhua Li, Jie Wang, and Fajie Yuan. Ninerec: A benchmark dataset suite for evaluating transferable recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5256–5267, 2024.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *Advances in neural information processing systems*, 34:6671–6682, 2021.

Sirui Zhang, Haonan Wang, Xunkai Li, Zekai Chen, Shumeng Li, Hongchao Qin, Rong-Hua Li, and Guoren Wang. Towards robust federated multimodal graph learning under modality heterogeneity. *arXiv preprint arXiv:2605.12584*, 2026.

Yinlin Zhu, Xunkai Li, Jishuo Jia, Miao Hu, Di Wu, and Meikang Qiu. Towards effective federated graph foundation model via mitigating knowledge entanglement. *Advances in Neural Information Processing Systems*, 38:64599–64628, 2026a.

Yinlin Zhu, Di Wu, Xianzhi Zhang, Yuming Ai, Xunkai Li, Miao Hu, and Guocong Quan. Rethinking federated graph foundation models: A graph-language alignment-based approach. *arXiv preprint arXiv:2601.21369*, 2026b.

Yinlin Zhu, Di Wu, Yi Zhang, Xunkai Li, Wang Luo, Wei-Jin Huang, Miao Hu, and Guocong Quan. Towards effective federated multimodal graph learning via navigating multifaceted heterogeneity. *arXiv preprint arXiv:2608.00623*, 2026c.

A Appendix

A.1 Problem Formulation

We consider federated multimodal graph foundation models consisting of a pretrained multimodal Encoder E_Θ and a GNN G_Φ. Each client k owns a private multimodal graph $\mathcal{G}_k = (\{\mathbf{X}_k^m\}_{m\in\mathcal{M}}, \mathbf{A}_k, \mathbf{Y}_k)$, where $\mathcal{M}$ denotes the modality set, $\mathbf{A}_k$ the graph structure, and $\mathbf{Y}_k$ the task labels. Raw data remain local, and downstream prediction is performed as

$$\widehat{\mathbf{Y}}_k = H_{\Omega_k}(G_\Phi(E_\Theta(\{\mathbf{X}_k^m\}_{m\in\mathcal{M}}), \mathbf{A}_k)) \,. \tag{11}$$

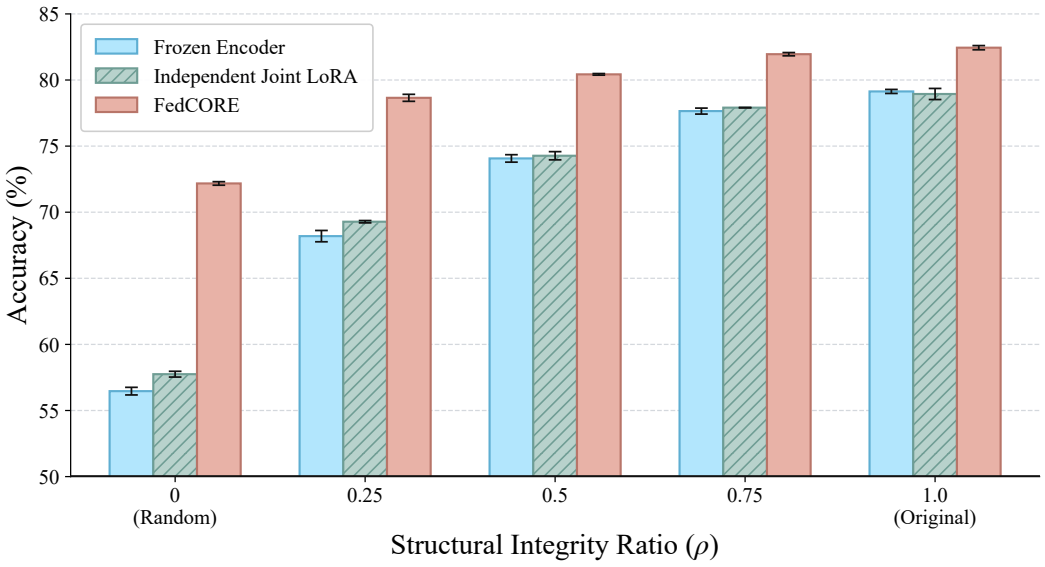


Figure 8: **Performance under progressive structural randomization.** Absolute accuracy of Frozen Encoder, Independent Joint LoRA, and FedCORE as the structural integrity ratio ρ increases from a randomized graph to the original graph.

Conventional federated adaptation aggregates Encoder and GNN updates separately across clients, without explicitly preserving their locally learned coordination. We therefore study how to jointly represent and federatively evolve perception–reasoning adaptations while keeping client privacy.

A.2 Structural Randomization Analysis

To complement the parameter-level analysis in Sec. 3, we further study how the methods behave when graph relations are gradually corrupted. We construct graphs with different structural integrity ratios ρ, ranging from a fully randomized graph ($\rho = 0$) to the original graph ($\rho = 1$), while keeping node contents and supervision unchanged. This type of community-based subgraph construction is consistent with standard subgraph federated learning protocols (Wang et al., 2022; Zhang et al., 2021).

As shown in Fig. 8, all methods improve as more graph structure is preserved, and FedCORE consistently achieves higher accuracy across all integrity levels. This result complements Observation 2 by showing the performance-level effect of structural corruption, whereas the main empirical study examines how relational changes directly alter Encoder adaptation.

A.3 Controlled Protocols for Empirical Study

This section details the controlled protocols underlying the empirical observations in Sec. 3. The studies are designed as a progressive diagnostic chain rather than independent performance comparisons. This diagnostic setting is related to prior work on subgraph federated learning, which studies missing cross-client neighbors and structural heterogeneity across local subgraphs (Zhang et al., 2021; Xie et al., 2021; Baek et al., 2023). Specifically, we examine whether graph-side capacity can substitute for Encoder adaptation, whether graph-relational interventions alter the preferred Encoder update, and whether the resulting Encoder and GNN adaptations remain independently composable. Unless otherwise specified, all experiments use identical node contents, labels, data splits, pretrained backbones, and optimization budgets, with only the investigated factor being changed.

A.3.1 Diagnosing the Limit of Graph-Side Adaptation

The first study examines whether the benefit of Encoder adaptation can be reproduced simply by increasing GNN capacity. We conduct the analysis on Toys and QB and keep the pretrained multimodal Encoder completely frozen. For each GNN family $g \in \{\mathrm{GCN}, \mathrm{GraphSAGE}, \mathrm{GAT}\}$, we

construct a sequence of models with increasing hidden capacity and record their trainable parameter budgets P_g. Their performance is compared with Encoder-LoRA and full fine-tuning under the same data partition and training protocol.

Let $\mathcal{S}_g(P)$ denote the performance of a frozen-Encoder model with GNN family g and approximately P trainable parameters, and let $\mathcal{S}_E$ denote the performance obtained when the Encoder is adapted. We characterize the residual representation gap at budget P as

$$\Delta_{\text{cap}}(P) = \mathcal{S}_E - \max_{g \in \mathcal{G}} \mathcal{S}_g(P), \quad \mathcal{G} = \{\text{GCN}, \text{GraphSAGE}, \text{GAT}\}. \tag{12}$$

A positive $\Delta_{\text{cap}}(P)$ across sufficiently expressive GNN configurations indicates that additional graph-side capacity cannot fully substitute for changing the multimodal representation itself. Importantly, this study does not require GNN performance to strictly saturate. The diagnostic criterion is whether stronger GNNs can *consistently close* the gap to Encoder-adapted models. As observed in Fig. 1, increasing GNN capacity improves performance, particularly on QB, yet the frozen-Encoder configurations remain consistently constrained relative to Encoder adaptation. This supports the premise that federated multimodal GFMs should expose adaptation capacity on both perception and graph-reasoning sides.

A.3.2 Controlled Relational Intervention

The second study examines whether Encoder adaptation is affected by the graph on which the representations are consumed, beyond node-level multimodal content. To examine this effect, we construct a family of controlled graphs

$$\mathcal{G}^{(\tau)} = \left(\mathcal{V}, \mathcal{A}^{(\tau)}, \mathbf{X}, \mathbf{Y}\right), \quad \tau \in [0, 1], \tag{13}$$

where node identities, multimodal contents $\mathbf{X}$, labels $\mathbf{Y}$, and train/validation/test splits remain fixed, while only the relational structure $\mathcal{A}^{(\tau)}$ is changed. Relational interventions are generated through constrained edge rewiring with approximately preserved node-degree profiles,

$$\left\| \mathbf{d}^{(\tau)} - \mathbf{d}^{(0)} \right\|_1 \approx 0, \tag{14}$$

so that the intervention primarily changes *who is connected to whom* rather than trivially changing graph density or node exposure.

We quantify the semantic consequence of a structural intervention using the change of the neighborhood-conditioned semantic profile. Let $\mathbf{z}_i$ denote the fixed multimodal representation of node i and

$$\mathbf{h}_i^{(\tau)} = \frac{\sum_{j \in \mathcal{N}_i^{(\tau)}} \mathbf{z}_j}{|\mathcal{N}_i^{(\tau)}| + \epsilon}. \tag{15}$$

The relational semantic shift is measured as

$$D_{\text{rel}}^{(\tau)} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left[1 - \cos\left(\mathbf{h}_i^{(\tau)}, \mathbf{h}_i^{(0)} \right) \right]. \tag{16}$$

This quantity measures how strongly the intervention changes the semantic context delivered by graph relations while leaving the node content itself unchanged.

For each intervened graph, we keep the GNN architecture fixed and optimize only Encoder-LoRA parameters together with the task head. Rather than comparing raw LoRA factors, whose factorization is non-unique, we analyze the effective update of each adapted layer,

$$\Delta \mathbf{W}_\ell^{(\tau)} = \mathbf{B}_\ell^{(\tau)} \mathbf{A}_\ell^{(\tau)}. \tag{17}$$

We concatenate the normalized effective updates across adapted Encoder layers,

$$\mathbf{u}_E^{(\tau)} = \mathrm{vec}\left(\left\{\frac{\Delta \mathbf{W}_\ell^{(\tau)}}{\|\Delta \mathbf{W}_\ell^{(\tau)}\|_F + \epsilon}\right\}_{\ell \in \mathcal{L}_E}\right), \tag{18}$$

and measure its deviation from the original relational environment as

$$d_{\mathrm{upd}}^{(\tau)} = 1 - \cos\left(\mathbf{u}_E^{(\tau)}, \mathbf{u}_E^{(0)}\right). \tag{19}$$

To compare the relational intervention with a lower-semantic-shift control, we construct perturbations that produce smaller observed changes in neighborhood semantic profiles. We refer to these as low-semantic-shift perturbation controls rather than semantic-preserving placebo interventions, because neighborhood label homophily is not explicitly constrained. The diagnostic therefore compares the observed Encoder-update displacement under the relational interventions and the low-semantic-shift controls.

$$\mathcal{R} = \frac{d_{\mathrm{upd}}^{\mathrm{rel}}}{d_{\mathrm{upd}}^{\mathrm{plc}} + \epsilon}, \tag{20}$$

together with the dose–response coefficient between relational shift and Encoder-update movement,

$$\widehat{\beta} = \frac{\sum_\tau \left(D_{\mathrm{rel}}^{(\tau)} - \overline{D}_{\mathrm{rel}}\right)\left(d_{\mathrm{upd}}^{(\tau)} - \overline{d}_{\mathrm{upd}}\right)}{\sum_\tau \left(D_{\mathrm{rel}}^{(\tau)} - \overline{D}_{\mathrm{rel}}\right)^2 + \epsilon}. \tag{21}$$

As reported in Sec. 3, the strongest relational intervention produces an update distance of 0.088, compared with 0.036 under the low-semantic-shift control, yielding a $2.4\times$ difference. The corresponding dose–response slopes are 0.70 and 0.24. These results do not imply that topology heterogeneity is necessarily stronger than feature heterogeneity. They provide controlled evidence that graph-relational perturbations are associated with systematic changes in the preferred Encoder adaptation. Because neighborhood label homophily is not explicitly controlled, the result does not establish a pure topology-only causal effect. This motivates representing perception adaptation as a relation-sensitive state rather than as an isolated multimodal adapter.

A.3.3 Cross-Client Encoder–GNN Compatibility

The preceding study shows that the Encoder update depends on the local relational environment. We therefore examine whether Encoder and GNN adaptations learned within the same client remain independently interchangeable. For each client k, we obtain a locally adapted pair (E_k, G_k) and evaluate every cross-client composition (E_i, G_j). This produces the compatibility matrix

$$\mathbf{M}_{ij} = \mathcal{S}\left(E_i, G_j; \mathcal{D}\right), \quad i, j \in \{1, \ldots, K\}, \tag{22}$$

where $\mathcal{S}(\cdot)$ denotes downstream performance under a common evaluation protocol. Diagonal entries correspond to Encoder–GNN adaptations formed under the same client environment, whereas off-diagonal entries deliberately break this locally learned correspondence.

We quantify this effect through the pairing gap

$$\Delta_{\mathrm{pair}} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{M}_{ii} - \frac{1}{K(K-1)}\sum_{i \neq j} \mathbf{M}_{ij}. \tag{23}$$

A large Δ_{pair} indicates that the Encoder and GNN do not behave as independently exchangeable adapters: their effectiveness depends on being paired with the complementary state learned under

the same relational environment. We apply the same evaluation protocol to the three parameterizations. Because the baselines and FedCORE also differ in their adapter coordinates, this diagnostic compares end-to-end compatibility and does not by itself isolate the effect of core sharing from coordinate anchoring.

The observed pairing gaps are 30.6 for Independent Joint LoRA, 22.8 after module-wise federated aggregation, and 5.9 for FedCORE. Independent Joint LoRA therefore demonstrates that simply optimizing both modules with the same local objective is insufficient to make their adaptations independently composable. Module-wise aggregation partially reduces the discrepancy but still aggregates two marginal parameter states rather than the state that generates their compatibility. FedCORE instead constrains both module updates to be realizations of the same client-specific core, substantially reducing the mismatch while preserving strong matched-pair performance.

These three diagnostics form the empirical basis of FedCORE. The capacity study establishes the need to adapt multimodal perception, the intervention study shows that such adaptation is conditioned by graph relations, and the pairing study shows that perception and graph reasoning form a compatible client-specific state. The resulting design principle is therefore not simply to make both modules trainable, but to *jointly represent* their relation-conditioned adaptation through a common state that can survive federated aggregation.

A.4 Theoretical Properties of Shared-Core Adaptation

We provide several theoretical properties that connect the empirical observations in Sec. 3 with the design of FedCORE. Rather than establishing a generic convergence guarantee for federated optimization, our analysis focuses on four questions specific to our setting: why graph relations can affect Encoder adaptation, how the shared core constrains Encoder–GNN compatibility, why such compatibility is preserved after federated aggregation, and under what conditions a local geometry-balanced update decreases the objective. These propositions characterize the shared linear parameterization and a single preconditioned local step. They do not constitute a convergence guarantee for multi-step federated optimization with client drift or state-quality weighting.

Relation-conditioned Encoder adaptation. Consider the local prediction function

$$\widehat{\mathbf{Y}}_k = H_{\omega_k}\left(G_{\Phi(\mathbf{C}_k)}\left(E_{\Theta(\mathbf{C}_k)}(\mathbf{X}_k), \mathbf{A}_k\right)\right), \tag{24}$$

where both Encoder and GNN parameters are induced by the client core $\mathbf{C}_k$. Let ℓ_k denote the task loss.

Proposition A.1 (Relation-conditioned Encoder gradient)**.** *Let*

$$\mathbf{H}_k = E_{\Theta(\mathbf{C}_k)}(\mathbf{X}_k)$$

and assume that the GNN, task head, and task loss are differentiable. The gradient received by the Encoder satisfies

$$\nabla_{\mathbf{H}_k}\ell_k = \mathbf{J}_G^\top(\mathbf{H}_k, \mathbf{A}_k)\mathbf{J}_H^\top \nabla_{\widehat{\mathbf{Y}}_k}\ell_k, \tag{25}$$

where $\mathbf{J}_G(\mathbf{H}_k, \mathbf{A}_k)$ is the Jacobian of graph propagation with respect to its input representations and $\mathbf{J}_H$ is the Jacobian of the task head. Therefore, even when $\mathbf{X}_k$ and the labels remain unchanged, modifying $\mathbf{A}_k$ can alter the gradient applied to the multimodal Encoder whenever

$$\mathbf{J}_G(\mathbf{H}_k, \mathbf{A}_k) \neq \mathbf{J}_G(\mathbf{H}_k, \mathbf{A}'_k). \tag{26}$$

Proof. By the chain rule,

$$\nabla_{\mathbf{H}_k}\ell_k = \left(\frac{\partial G_{\Phi(\mathbf{C}_k)}(\mathbf{H}_k, \mathbf{A}_k)}{\partial \mathbf{H}_k}\right)^\top \left(\frac{\partial H_{\omega_k}}{\partial G_{\Phi(\mathbf{C}_k)}}\right)^\top \nabla_{\widehat{\mathbf{Y}}_k}\ell_k, \tag{27}$$

which gives the stated expression. Since message passing explicitly depends on $\mathbf{A}_k$, its Jacobian generally changes when the relational structure changes.

For example, consider a message-passing layer

$$\mathbf{Z} = \sigma\left(\widetilde{\mathbf{A}}_k \mathbf{H}_k \mathbf{W}_G\right). \tag{28}$$

Let

$$\mathbf{Q} = \nabla_{\mathbf{Z}} \ell_k \odot \sigma' \left(\widetilde{\mathbf{A}}_k \mathbf{H}_k \mathbf{W}_G \right).$$

Then

$$\nabla_{\mathbf{H}_k} \ell_k = \widetilde{\mathbf{A}}_k^\top \mathbf{Q} \mathbf{W}_G^\top, \tag{29}$$

which explicitly depends on the graph relations. Hence, changing only $\mathbf{A}_k$ is sufficient to change the learning signal received by the Encoder, except in degenerate cases where the relevant Jacobians or downstream gradients coincide. □

Proposition A.1 provides a mechanistic explanation for Observation 2. Graph relations do not merely determine how fixed representations are propagated; through back-propagation, they also condition how the upstream multimodal representation should be adapted.

A shared compatibility manifold. For an adapted module j, FedCORE decodes the shared core as

$$\mathcal{D}_j(\mathbf{C}) = \alpha_j \mathbf{U}_j \mathbf{S}_j \mathbf{C} \mathbf{S}_j \mathbf{V}_j^\top, \tag{30}$$

where $\mathbf{U}_j$ and $\mathbf{V}_j$ are fixed rank-r coordinate bases satisfying

$$\mathbf{U}_j^\top \mathbf{U}_j = \mathbf{V}_j^\top \mathbf{V}_j = \mathbf{I}_r,$$

$\mathbf{S}_j$ is a fixed positive diagonal scaling matrix, and $\alpha_j > 0$ is a fixed module-specific scale. Its canonical projection is

$$\mathcal{P}_j(\Delta \mathbf{W}) = \alpha_j^{-1} \mathbf{S}_j^{-1} \mathbf{U}_j^\top \Delta \mathbf{W} \mathbf{V}_j \mathbf{S}_j^{-1}. \tag{31}$$

Proposition A.2 (Cross-module compatibility)**.** *For every adapted module j,*

$$\mathcal{P}_j \left(\mathcal{D}_j(\mathbf{C}) \right) = \mathbf{C}. \tag{32}$$

Consequently, all module updates generated by FedCORE belong to the coupled adaptation space

$$\mathcal{M} = \left\{ (\mathcal{D}_1(\mathbf{C}), \ldots, \mathcal{D}_J(\mathbf{C})) : \mathbf{C} \in \mathbb{R}^{r \times r} \right\}, \tag{33}$$

which has dimension r^2.

Proof. Substituting $\mathcal{D}_j(\mathbf{C})$ into $\mathcal{P}_j$ gives

$$\mathcal{P}_j(\mathcal{D}_j(\mathbf{C})) = \mathbf{S}_j^{-1} \mathbf{U}_j^\top \mathbf{U}_j \mathbf{S}_j \mathbf{C} \mathbf{S}_j \mathbf{V}_j^\top \mathbf{V}_j \mathbf{S}_j^{-1} \tag{34}$$

$$= \mathbf{C}. \tag{35}$$

Hence $\mathcal{D}_j$ is injective for every j. The joint decoding

$$\mathcal{D}(\mathbf{C}) = (\mathcal{D}_1(\mathbf{C}), \ldots, \mathcal{D}_J(\mathbf{C})) \tag{36}$$

is therefore also injective. Since its domain is $\mathbb{R}^{r \times r}$ and all $\mathcal{D}_j$ are linear, $\mathcal{M}$ is a linear subspace of dimension r^2. □

This property gives a precise interpretation of a *shared adaptation state*: although different modules receive heterogeneous parameter updates, every update can be uniquely projected back to the same core.

To quantify deviations from this condition, for arbitrary independently learned module updates $\{\Delta \mathbf{W}_j\}_{j=1}^J$, define the compatibility defect

$$\mathcal{I}\left(\{\Delta \mathbf{W}_j\}_{j=1}^J \right) = \min_{\mathbf{C}} \sum_{j=1}^J \omega_j \left\| \mathcal{P}_j(\Delta \mathbf{W}_j) - \mathbf{C} \right\|_F^2, \qquad \omega_j > 0. \tag{37}$$

Its minimizer has the closed form

$$\mathbf{C}^\star = \frac{\sum_{j=1}^J \omega_j \mathcal{P}_j(\Delta \mathbf{W}_j)}{\sum_{j=1}^J \omega_j}. \tag{38}$$

For FedCORE,

$$\mathcal{I}\left(\{\mathcal{D}_j(\mathbf{C})\}_{j=1}^J \right) = 0, \tag{39}$$

whereas independently parameterized Encoder and GNN adapters generally have a positive defect whenever their projected latent states disagree. This provides a parameter-space interpretation of the cross-module incompatibility observed in Observation 3.

Aggregation closure of the coupled state.

Proposition A.3 (Aggregation consistency)**.** *Let $\mathbf{C}_k$ denote the locally evolved core of client k and let*

$$\overline{\mathbf{C}} = \sum_{k=1}^{K} p_k \mathbf{C}_k, \qquad \sum_{k=1}^{K} p_k = 1. \tag{40}$$

Then, for every module j,

$$\mathcal{D}_j(\overline{\mathbf{C}}) = \sum_{k=1}^{K} p_k \mathcal{D}_j(\mathbf{C}_k), \tag{41}$$

and the resulting global module updates remain in $\mathcal{M}$ with zero compatibility defect.

Proof. Since $\mathcal{D}_j$ is linear,

$$\mathcal{D}_j\left(\sum_{k=1}^{K} p_k \mathbf{C}_k\right) = \alpha_j \mathbf{U}_j \mathbf{S}_j \left(\sum_{k=1}^{K} p_k \mathbf{C}_k\right) \mathbf{S}_j \mathbf{V}_j^\top \tag{42}$$

$$= \sum_{k=1}^{K} p_k \mathcal{D}_j(\mathbf{C}_k). \tag{43}$$

Moreover,

$$\mathcal{P}_j\left(\mathcal{D}_j(\overline{\mathbf{C}})\right) = \overline{\mathbf{C}} \tag{44}$$

for every j by Proposition A.2. Therefore all aggregated module updates recover the same global state and remain in $\mathcal{M}$. □

Under the same linear decoder and the same aggregation weights, aggregating the shared cores before decoding is equivalent to decoding each local core first and then averaging the resulting module updates. Therefore, a module-wise aggregation variant is different from FedCORE only when it changes the local parameterization, uses module-specific aggregation weights, or introduces additional module-specific operations.

Proposition A.3 shows that FedCORE restricts local updates to the shared compatibility family $\mathcal{M}$, which is closed under linear federated averaging. Independently learned Encoder and GNN updates need not share a common projected state, leading to cross-client combination incompatibility. Under fixed linear decoders and common aggregation weights, core-space aggregation and decoded module-wise averaging are algebraically equivalent; FedCORE's structural benefit is the shared parameterization and explicit compatibility constraint.

Exact joint gradient pullback. The shared core is not optimized by heuristically averaging Encoder- and GNN-side gradients. Its gradient follows exactly from the joint objective. Assume that the core regularization term in the local objective is

$$\frac{\lambda_r}{2}\|\mathbf{C}_k\|_F^2.$$

Proposition A.4 (Exact cross-module pullback)**.** *Let*

$$\mathbf{G}_{k,j}^{W} = \nabla_{\mathbf{W}_{k,j}} \mathcal{L}_k. \tag{45}$$

Under the Frobenius inner product, the adjoint of $\mathcal{D}_j$ is

$$\mathcal{D}_j^*(\mathbf{G}) = \alpha_j \mathbf{S}_j \mathbf{U}_j^\top \mathbf{G} \mathbf{V}_j \mathbf{S}_j. \tag{46}$$

Therefore,

$$\nabla_{\mathbf{C}_k} \mathcal{L}_k = \sum_{j \in \mathcal{J}} \mathcal{D}_j^*\left(\mathbf{G}_{k,j}^{W}\right) + \lambda_r \mathbf{C}_k. \tag{47}$$

Proof. For an arbitrary perturbation $\mathbf{Z}$,

$$\langle \mathbf{G}, \mathcal{D}_j(\mathbf{Z}) \rangle_F = \alpha_j \operatorname{tr}\left(\mathbf{G}^\top \mathbf{U}_j \mathbf{S}_j \mathbf{Z} \mathbf{S}_j \mathbf{V}_j^\top\right) \tag{48}$$

$$= \left\langle \alpha_j \mathbf{S}_j \mathbf{U}_j^\top \mathbf{G} \mathbf{V}_j \mathbf{S}_j, \mathbf{Z} \right\rangle_F . \tag{49}$$

Hence the stated expression is the adjoint mapping. Applying the chain rule over all decoded modules and differentiating $\lambda_r \|\mathbf{C}_k\|_F^2/2$ gives the result. □

This shows that Encoder- and GNN-side learning signals are exactly transported into the same coordinate system and jointly determine the evolution of $\mathbf{C}_k$. In particular, by Proposition A.1, the Encoder-side component already contains information about the local graph relations through backpropagation.

Geometry-balanced core optimization. The second-moment statistics and preconditioned update below are the same quantities used in the local optimization procedure in Sec. 5.3. The sufficient descent result concerns one local update with fixed preconditioners; it does not establish convergence of the complete federated procedure. Let

$$\mathbf{G}_k^e = \nabla_{\mathbf{C}_k^e} \mathcal{L}_k^e$$

denote the core gradient at local step e. The client maintains the left and right second-moment statistics

$$\mathbf{L}_k^e = \beta \mathbf{L}_k^{e-1} + (1-\beta) \mathbf{G}_k^e (\mathbf{G}_k^e)^\top, \tag{50}$$

$$\mathbf{R}_k^e = \beta \mathbf{R}_k^{e-1} + (1-\beta) (\mathbf{G}_k^e)^\top \mathbf{G}_k^e, \tag{51}$$

where $\beta \in [0, 1)$ is a decay coefficient. These statistics are maintained locally and are not communicated to the server. The resulting update follows the general family of matrix preconditioned optimization methods that maintain left and right gradient statistics (Gupta et al., 2018).

For readability, we omit the local-step superscript below and define

$$\mathbf{P}_L = (\mathbf{L}_k + \epsilon \mathbf{I})^{-1/4}, \qquad \mathbf{P}_R = (\mathbf{R}_k + \epsilon \mathbf{I})^{-1/4}, \tag{52}$$

where $\epsilon > 0$ ensures positive definiteness. The geometry-balanced core update is

$$\mathbf{C}^+ = \mathbf{C} - \eta \mathbf{P}_L \nabla_{\mathbf{C}} \mathcal{L} \mathbf{P}_R. \tag{53}$$

Proposition A.5 (Sufficient descent under fixed preconditioning)**.** *Assume $\mathcal{L}(\mathbf{C})$ is L_C-smooth and $\mathbf{P}_L, \mathbf{P}_R$ are symmetric positive definite and fixed during one update. Let*

$$\mathbf{M} = \mathbf{P}_R \otimes \mathbf{P}_L. \tag{54}$$

If

$$0 < \eta < \frac{2\lambda_{\min}(\mathbf{M})}{L_C \lambda_{\max}^2(\mathbf{M})}, \tag{55}$$

then every non-stationary update yields

$$\mathcal{L}(\mathbf{C}^+) < \mathcal{L}(\mathbf{C}). \tag{56}$$

Proof. Let

$$\mathbf{g} = \operatorname{vec}\left(\nabla_{\mathbf{C}} \mathcal{L}\right).$$

Vectorizing the update gives

$$\operatorname{vec}(\mathbf{C}^+ - \mathbf{C}) = -\eta \mathbf{M} \mathbf{g}. \tag{57}$$

By L_C-smoothness,

$$\mathcal{L}(\mathbf{C}^+) \le \mathcal{L}(\mathbf{C}) - \eta \mathbf{g}^\top \mathbf{M} \mathbf{g} + \frac{L_C \eta^2}{2} \|\mathbf{M}\mathbf{g}\|_2^2 \tag{58}$$

$$\le \mathcal{L}(\mathbf{C}) - \eta \left[\lambda_{\min}(\mathbf{M}) - \frac{L_C \eta}{2} \lambda_{\max}^2(\mathbf{M})\right] \|\mathbf{g}\|_2^2. \tag{59}$$

The stated step-size condition makes the term in brackets strictly positive, yielding strict descent whenever $\mathbf{g} \neq 0$. □

The analysis therefore establishes a direct connection between our empirical findings and method design. Graph relations can systematically alter the Encoder learning signal; Encoder and GNN updates generated from the same core lie on a common r^2-dimensional compatibility manifold; this manifold is preserved by linear federated aggregation; and a single local update with a fixed positive-definite preconditioner satisfies a sufficient descent condition.

These results explain the structural motivation of FedCORE, but they do not provide a convergence guarantee for the complete federated optimization process. In particular, they do not analyze client drift over multiple local steps, partial client participation, state-quality weighting, or the interaction between local preconditioner statistics and server-side state evolution.

A.5 Experimental Setup

A.5.1 Datasets and Tasks

We evaluate FedCORE on multimodal graph benchmarks covering diverse downstream tasks, including node classification, link prediction, multimodal retrieval, graph-to-text generation, and graph-to-image generation. All datasets follow the federated multimodal graph learning protocol introduced by MM-OpenFGL (Li et al., 2026a), where each client owns a local multimodal graph and raw data remain decentralized.

Amazon product graphs. Toys and Grocery are multimodal Amazon product graphs. Each node represents an individual product with textual and visual attributes, while product–product relations, such as co-purchase or complementary-product relations, define the graph edges. Toys contains 20,695 nodes and 18 product categories, whereas Grocery contains 17,074 nodes and 20 product categories. We evaluate node classification on both datasets and report classification accuracy.

NineRec-based datasets. Bili Music, DY, QB, and Bili Cartoon are derived from the NineRec benchmark (Zhang et al., 2024). These datasets contain multimodal user–item interaction graphs with heterogeneous semantic information. Following MM-OpenFGL (Li et al., 2026a), we evaluate link prediction on Bili Music and DY using AUC. For QB and Bili Cartoon, we evaluate multimodal retrieval using Recall@5, measuring whether the relevant item appears among the top-five retrieved candidates.

Flickr30k graph-to-text generation. For graph-to-text generation, we use the preprocessed Flickr30k graph with 31,783 nodes and partition it into five clients using `subgraph_fl_louvain`. Visual features are extracted using CLIP ViT-L/14 with dimension 768, while textual features are extracted using RoBERTa-base. We use the trainable `flan-t5-base-VG-factual-sg` decoder with `freeze_lm=false` and no PEFT adapter. Neighbor information is constructed using the embedding-based neighbor mode, with at most four text neighbors, four image neighbors, and one neighbor layer. The maximum input and output lengths are 1024 and 128 tokens, respectively. The model is trained for five epochs with a learning rate of 1×10^{-3}, a batch size of 4, weight decay of 1×10^{-2}, and gradient accumulation over four steps. Generation quality is evaluated using ROUGE-L.

SemArt graph-to-image generation. For graph-to-image generation, we use the preprocessed SemArt graph with 21,382 nodes and partition it into five clients using `subgraph_fl_louvain`. The graph-conditioned generator is based on Stable Diffusion v1.5 with a resolution of 256×256. Graph information is incorporated through a GNN, a GraphAdapter, and a Q-Former. We use four graph neighbors, a cross-attention frequency of 2, and four additional CLIP context tokens. The model is trained for 10 epochs with a learning rate of 1×10^{-4}, a batch size of 16, and Adam weight decay of 1×10^{-2}. Generation quality is evaluated using CLIP-Score.

The datasets and their downstream tasks are described above, and all experiments follow the federated multimodal graph learning protocol of MM-OpenFGL.

A.5.2 Evaluation Metrics

For node classification tasks, we report Accuracy, defined as the proportion of correctly classified nodes:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i). \tag{60}$$

For link prediction tasks, we use Area Under the ROC Curve (AUC), which measures the probability that a randomly sampled positive edge receives a higher score than a negative edge.

For multimodal retrieval tasks, we adopt Recall@5 (R@5), which evaluates the percentage of queries whose corresponding targets appear in the top-5 retrieved candidates.

For graph-to-text generation, we report ROUGE-L, which measures the longest-common-subsequence overlap between the generated text and its reference text.

For graph-to-image generation, we report CLIP-Score, computed as the cosine similarity between the CLIP embeddings of the generated image and its corresponding target image. We use the pre-trained CLIP ViT-L/14 model with its standard image and text preprocessing.

A.5.3 Baselines

We compare FedCORE with representative baselines from three categories.

Federated Learning Baseline. FedAvg (McMahan et al., 2017) is the standard federated optimization method that independently averages client model parameters.

Federated Graph Foundation Model Baselines. FedGFM+ (**?**), STAGE (Chen et al., 2026a), and FedGAMMA (Li et al., 2026b) are representative approaches for adapting pretrained graph foundation models under federated settings. These methods investigate parameter-efficient adaptation, knowledge sharing, and cross-client transferability.

Federated Multimodal Graph Baseline. Federated PLANET (Liu et al., 2026) extends multimodal graph foundation models to decentralized environments and serves as a strong comparison for multimodal graph adaptation.

Adaptation controls. To analyze the contribution of Encoder–GNN coordination, we include two adaptation controls.

Frozen Encoder + GNN updates only graph-side parameters while keeping the pretrained multimodal Encoder fixed. This baseline evaluates whether graph reasoning alone can achieve effective adaptation.

Independent Joint LoRA + FedAvg is implemented as an isolated matched-backbone control. The common backbone uses the frozen visual, textual, and fusion projections from FedCOREv2, with input features extracted by CLIP ViT-L/14. Independent LoRA adapters are applied to the visual projection, textual projection, fusion projection, and the three custom residual GNN transformation matrices. The GNN contains three message-passing layers with hidden dimension 256. For each adapted matrix, the update is parameterized as

$$\Delta\mathbf{W}_j = \frac{\alpha}{r}\mathbf{B}_j\mathbf{A}_j,$$

where $\alpha = 4r$ and no LoRA dropout is used. We evaluate ranks $r \in \{1, 2, 4, 8, 16, 32\}$ in the rank-sensitivity experiments; the adaptation-scale comparison uses $r = 24$.

The LoRA factors are optimized independently on each client and aggregated using FedAvg. The same five clients, full client participation, warm-up schedule, local epochs, optimizer, and data splits are used as for FedCORE. The local task classifier is trainable but remains client-specific: parameters whose names start with `classifier.` are excluded from the federated payload and are not averaged by the server. Thus, the reported trainable parameter count includes both LoRA factors and the local task head, whereas the communication payload counts only the LoRA $\mathbf{A}_j$ and $\mathbf{B}_j$ tensors.

A.5.4 Implementation Details

All methods are evaluated under identical data splits, pretrained backbones, and downstream settings. Following MM-OpenFGL (Li et al., 2026a), each dataset is partitioned into five clients unless

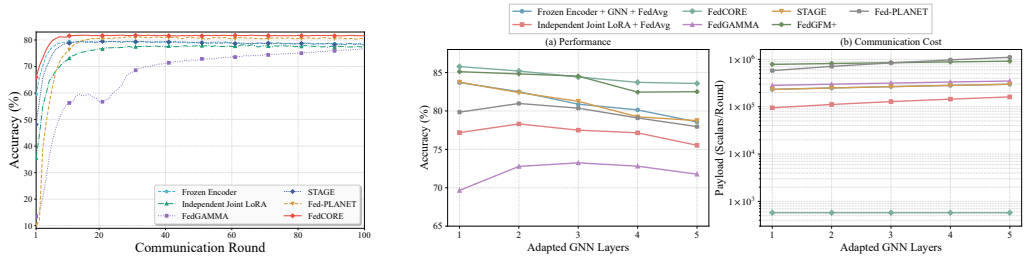
otherwise specified. For node classification, multimodal retrieval, graph-to-text generation, and graph-to-image generation, we use Louvain community detection to construct the client subgraphs. For link prediction, we use Metis graph partitioning(Wang et al., 2022). All compared methods use the same client partitions.

For FedCORE, only the upper layers of the multimodal Encoder, fusion layers, and primary GNN transformations are adapted, while lower backbone layers remain frozen. The latent state dimension and method-specific optimization hyperparameters are selected using the validation set and fixed before evaluation. For the Independent Joint LoRA control, clients use two local epochs with Adam optimization, learning rate 0.004, weight decay 0.0012, batch size 512, and dropout rate 0.38. The warm-up stage lasts 10 rounds, followed by 45 adaptation rounds. All five clients participate in every communication round.

B Additional Efficiency and Scalability Analysis

Convergence Efficiency. Figure 9(a) compares the performance of FedCORE and representative baselines across communication rounds on Toys node classification. FedCORE reaches a strong performance level within fewer rounds and remains stable afterwards, providing additional evidence that preserving Encoder–GNN coordination can improve federated optimization.

Scalability to Adaptation Scale. Figure 9(b) examines how the methods behave as the adaptation scale increases while the core dimension of FedCORE remains fixed. Independent Joint LoRA communicates more module-specific parameters as additional layers are adapted, whereas FedCORE continues to transmit a compact $r \times r$ core. The results provide additional evidence of the performance–communication trade-off associated with increasing the adaptation scale.



(a) Convergence efficiency. (b) Adaptation scale.

Figure 9: **Additional convergence and scalability analysis.** (a) Test accuracy on Toys node classification across communication rounds. (b) Performance–communication trade-off on Grocery node classification as the adaptation scale increases.

C Additional Robustness Analysis

Robustness to Client Heterogeneity. We evaluate FedCORE on Toys node classification under client-specific feature drift and topology perturbation. We keep the node set, labels, data splits, and training settings fixed. For feature drift, we add a shared offset $b_k \sim \mathcal{N}(0, \alpha^2 I)$ to all nodes of client k, with $\alpha \in \{0, 0.15, 0.30, 0.45, 0.60\}$. For topology perturbation, we inject undirected edges between nodes from different Louvain communities before the final five-client partition, using $\rho \in \{0, 0.0625, 0.125, 0.1875, 0.25\}$. As shown in Figure 10, FedCORE degrades more slowly than the baselines as either perturbation increases.

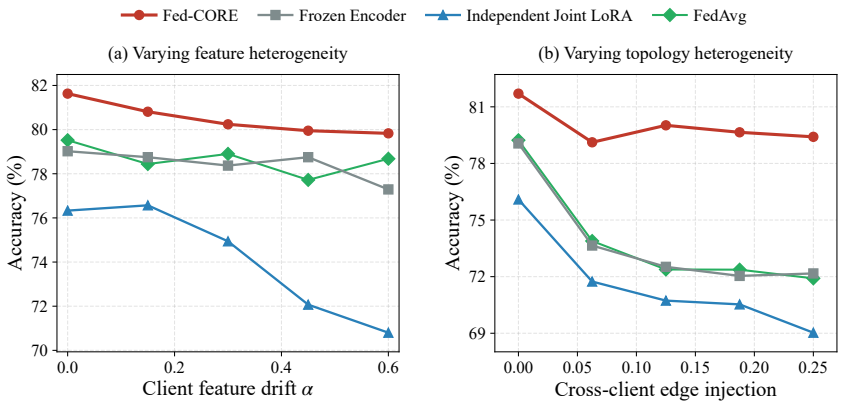


Figure 10: **Robustness to client heterogeneity.** The left panel varies client feature drift α, while the right panel varies the cross-community edge-injection ratio ρ before the final client partition. The node set, labels, and data splits remain fixed.