RECOG: RECIPROCAL CO-EVOLUTION FOR MULTIMODAL GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal graph learning requires jointly training over graph structure and heterogeneous node attributes, yet existing methods largely decouple these processes: prior multimodal graph neural networks (GNNs) focus on aligning modalities in a shared embedding space while operating on fixed or weakly adapted graph structures, and graph structure learning approaches infer topology from unimodal node representations without accounting for multimodal interactions. This separation fundamentally limits the ability of GNNs to capture semantically meaningful relationships in multimodal settings, where observed edges are often noisy, incomplete, or misaligned with underlying semantics. We propose **ReCoG** (Reciprocal Co-Evolution for Multimodal Graph Learning), a new learning paradigm that tightly couples graph structure learning and multimodal representation learning through **end-to-end reciprocal interaction**. Concretely, ReCoG integrates (i) **a multimodal graph refiner** that infers and corrects edges using cross-modal semantic evidence, and (ii) **a coupled cross-modal message passing mechanism** that performs joint intra- and inter-modality propagation over the refined graph. This unified design yields greater expressiveness than decoupled or two-stage formulations and allows dynamic interaction between topology and representation learning. Across diverse benchmarks for node classification and link prediction, ReCoG consistently outperforms strong multimodal graph structure learning baselines, including graph foundation models. Our results demonstrate that **reciprocal co-evolution of structure and semantics is important for effective multimodal graph learning**, challenging the prevailing separation between topology and representation learning. *Code will be released upon acceptance.*

1 INTRODUCTION

Graph neural networks (GNNs) have become a dominant paradigm for learning over relational data, achieving strong performance across applications such as recommendation systems, social networks, and scientific modeling (Tang et al., 2020; Sankar et al., 2021; Fout et al., 2017; Wu et al., 2022b; Zhang et al., 2024; Kipf & Welling, 2016; Gasteiger et al., 2019; Veličković et al., 2017; Wu et al., 2019; Xue et al., 2023a). While most existing GNNs assume *unimodal* node attributes, real-world graphs are inherently **multimodal**, where nodes are described by heterogeneous signals such as text, images, and metadata. These modalities provide complementary semantic views, making multimodal graph learning an important step toward more realistic and expressive representation learning.

Despite recent progress, existing approaches to multimodal graph learning remain fundamentally limited by a **decoupled design**. On one hand, multimodal GNNs typically focus on aligning representations across modalities—often through separate encoders or late fusion—while operating on **fixed or weakly adapted graph structures**. On the other hand, graph structure learning methods (Sec. 2) refine topology jointly with representations, but only from unimodal evidence and with single-stream propagation. As a result, these methods fail to capture the tight interplay between structure and semantics that is essential in multimodal settings, where observed edges are often noisy, incomplete, or misaligned with multimodal semantic relationships.

We argue that **graph structure and multimodal representations** should not be treated as independent components, but rather as **mutually dependent variables that must be learned jointly**. In multimodal graphs, better representations provide stronger semantic signals for identifying meaningful connections, while improved graph structure enables more effective cross-modal information

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.

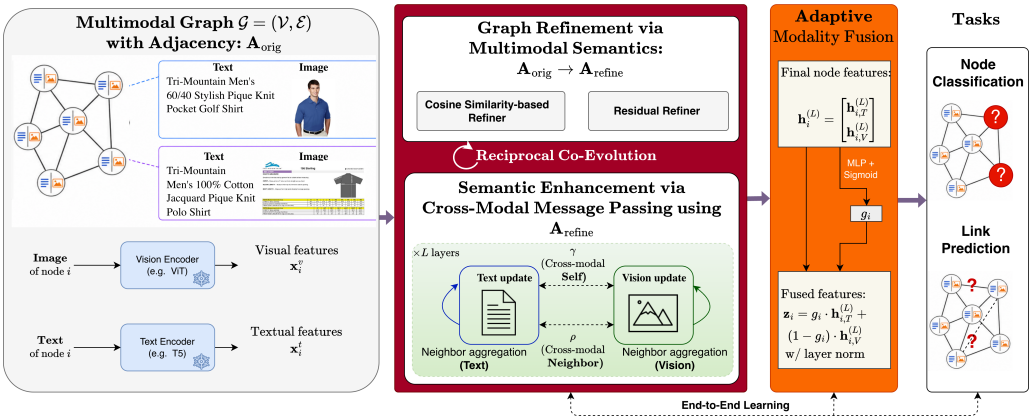


Figure 1: Overview of the proposed ReCoG framework. **Left:** The input multimodal graph with text and vision features. **Center-top:** The Structure Refiner module with two variants: a similarity-based version, and a residual MLP-based version. **Center-bottom:** Cross-modal coupled message passing over L layers, where each layer performs intra-modal neighbor aggregation and cross-modal injection. **Right:** Modality fusion followed by a task-specific head. The entire pipeline is trained end-to-end. *Best viewed in magnification.* See text for detail.

propagation. To realize this idea, we propose **ReCoG** (Reciprocal Co-Evolution for Multimodal Graph Learning, Fig. 1), a unified framework that tightly couples graph structure learning and multimodal representation learning through **end-to-end reciprocal interaction**. ReCoG consists of two interdependent components:

- **A multimodal graph refiner** leverages cross-modal semantic signals to denoise and augment the observed topology, correcting spurious edges and recovering missing but semantically meaningful connections (Fig. 2 and Appendix I).
- **A cross-modal message passing mechanism** performs joint intra- and inter-modality propagation over the evolving graph, enabling fine-grained interaction between modalities at every layer.

Importantly, these components are optimized jointly: updated representations iteratively guide topology refinement, while the refined graph directly shapes subsequent cross-modal propagation.

We evaluate ReCoG on nine multimodal graph benchmarks spanning node classification and link prediction tasks. Across diverse encoder configurations, the best ReCoG variant outperforms strong multimodal GNN and graph foundation model baselines on every benchmark, often by significant margins. These results demonstrate that **jointly evolving graph structure and multimodal semantics is important for effective multimodal graph learning.**

Our contributions. We position ReCoG against graph structure learning, multimodal graph learning, and the concurrent works in Sec. 2 (extended discussion in Appendix B). ReCoG makes three main contributions as follows:

- **A unified multimodal graph learning framework.** We propose ReCoG, which integrates a multimodal graph refiner with a coupled cross-modal message passing mechanism, enabling iterative interaction between structure and semantics throughout training, and challenging the conventional separation between topology and feature learning.
- **Spectral and contraction analysis of coupled propagation.** We give a supergraph reformulation of the coupled update that admits a closed-form spectral characterization of the modality coupling coefficients, and prove a depth-wise contraction bound that motivates our shared-weight design.
- **Strong empirical performance.** We demonstrate improvements over classical GNNs, multimodal GNNs, and recent multimodal graph models across nine benchmarks for node classification and link prediction.

2 Related Work

Graph structure learning (GSL). IDGL (Chen et al., 2020b), SLAPS (Fatemi et al., 2021), Pro-GNN (Jin et al., 2020), and NodeFormer (Wu et al., 2022a) learn the topology jointly with node

representations. They score node pairs in a *single* representation space, typically through (weighted) cosine similarity or attention, and propagate a single stream over the learned graph. Applied to multimodal graphs, they therefore see only concatenated features: modality-specific evidence is collapsed before refinement, and the modalities never exchange information during propagation. ReCoG instead scores each pair from modality-specific interaction features (Eq. 5) and propagates two coupled streams (Eq. 12).

Multimodal graph learning. MMGCN and MGAT (Zhu et al., 2025) run modality-specific GNNs with late fusion, while NSG-MoE (Zhang & Kuruoglu, 2026), LION (Li et al., 2026), PLANET (Liu et al., 2026), and UniGraph2 (He et al., 2025) improve modality interaction and alignment. All of these propagate over the *observed* topology. LLM-based approaches such as Graph4MM (Ning et al., 2025) and MLaGA (Fan et al., 2026) address a complementary setting that tunes large pretrained models, whereas ReCoG trains a lightweight GNN over frozen encoder features.

Concurrent work. TMTE (Zhu et al., 2026), a concurrent work released on arXiv in March 2026, similarly studies topology–representation co-evolution for multimodal graphs, but realizes it through substantially different mechanisms. TMTE first fuses the projected modalities into a single representation and propagates it over an evolved graph, whereas ReCoG preserves modality-specific streams and performs cross-modal self- and neighbor-level interaction at every layer. Moreover, TMTE updates topology through a non-negative similarity graph mixed with the original adjacency, while ReCoG learns signed, pair-specific residual corrections that can both suppress spurious edges and recover missing ones. The two methods also differ in scalability: TMTE uses node–anchor approximation, whereas ReCoG performs target-centric coarse-to-fine refinement within sampled mini-batches. Since the code linked by TMTE is not accessible, we report the results provided in the original paper. See Appendix B and K for details.

3 METHODOLOGY

We now describe the three components of ReCoG (Fig. 1): (1) a multimodal graph refiner that refines the observed topology (Sec. 3.1), (2) a cross-modal message passing module that propagates information over the refined graph (Sec. 3.2), and (3) an adaptive modality fusion layer that integrates modality-specific representations (Sec. 3.3). These components are optimized end-to-end, forming a closed loop between topology refinement and representation learning.

3.1 GRAPH REFINEMENT VIA MULTIMODAL SEMANTICS

We refine the observed topology using multimodal semantic evidence that adapts dynamically as representations evolve. We first state the refinement problem in general, then present two complementary instantiations: a *cosine similarity refiner* that is efficient and interpretable, and a *residual refiner* that provides greater expressive power.

Problem formulation. Given the multimodal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with adjacency $\mathbf{A}_{\text{orig}}$ and multimodal node features $\{\mathbf{x}_i^t, \mathbf{x}_i^v\}_{i\in\mathcal{V}}$, the refiner produces a refined adjacency $\mathbf{A}_{\text{refine}} \in [0,1]^{|\mathcal{V}|\times|\mathcal{V}|}$ by combining the original topology with a learned multimodal edge score.

Mini-batch target-centric candidates. Large-scale GNN training typically uses neighbor sampling. Let $\mathcal{V}_\mathcal{B}$ denote the sampled subgraph and $\mathcal{B} \subseteq \mathcal{V}_\mathcal{B}$ the set of target nodes, with $|\mathcal{B}| = B$ and $|\mathcal{V}_\mathcal{B}| = N_\mathcal{B}$. Since only target nodes receive direct supervision (Xue et al., 2023b), we restrict refinement to target-centric pairs:

$$\mathcal{P} = \big\{(i,j) : i \in \mathcal{B},\ j \in \mathcal{V}_\mathcal{B} \setminus \{i\}\big\} = \mathcal{S}_\mathcal{E} \cup \mathcal{S}_{\text{cand}}, \tag{1}$$

where $\mathcal{S}_\mathcal{E} \subset \mathcal{E}$ contains original edges to be re-scored (*denoising*) and $\mathcal{S}_{\text{cand}}$ ($\mathcal{S}_{\text{cand}} \cap \mathcal{E} = \emptyset$) contains candidate non-edges to be discovered (*recovery*). This reduces candidate complexity from $O(N_\mathcal{B}^2)$ to $O(BN_\mathcal{B})$. We provide a full analysis in Appendix J.

3.1.1 COSINE SIMILARITY REFINER

This variant provides a simple and interpretable mechanism based on pretrained feature geometry.

Multimodal similarity scoring. We compute modality-specific cosine similarities and combine them via a learnable convex combination to obtain $s(i,j)$.

$$s(i,j) = \max(0,\ \lambda S_V(i,j) + (1-\lambda) S_T(i,j))\,, \tag{2}$$

where $S_T(i,j) = \cos(\mathbf{x}_i^t, \mathbf{x}_j^t), S_V(i,j) = \cos(\mathbf{x}_i^v, \mathbf{x}_j^v)$, $\lambda = \sigma(\lambda_{\text{logit}}) \in (0,1)$ is learned, $\sigma(\cdot)$ is the sigmoid function. Textual similarity captures contextual alignment, visual similarity captures perceptual resemblance, and λ lets the model balance the two on a per-task basis. We mask self-loops and clamp negative similarities to zero to prevent anti-semantic edges.

Semantic neighbor selection. For each target node $i \in \mathcal{B}$, we select the top-k neighbors by similarity:

$$\mathcal{N}_{\text{sem}}(i) = \text{TopK}_{j \in \mathcal{V}_\mathcal{B}}\{s(i,j)\}, \qquad |\mathcal{N}_{\text{sem}}(i)| = k. \tag{3}$$

where k is a hyperparameter in training. This produces a set of candidate semantic edges, each associated with a similarity score. And we set $s(i,j) = 0, j \notin \mathcal{N}_{\text{sem}}(i)$.

Topology integration. We combine original and semantic edges via a convex combination:

$$\mathbf{A}_{\text{refine}}(i,j) = \text{Clamp}(\alpha \mathbf{A}_{\text{orig}}(i,j) + (1-\alpha)s(i,j)\mathbb{I}[j \in \mathcal{N}_{\text{sem}}(i)], 0, 1). \tag{4}$$

where $\alpha = \sigma(\alpha_{\text{logit}}) \in (0,1)$ is a learnable coefficient controlling the trade-off between observed topology and semantic evidence and $\text{Clamp}(x, 0, 1) = \min(1, \max(0, x))$. This formulation treats the original topology and semantic affinity as two competing sources of edge evidence: existing edges can be down-weighted when their semantic similarity is low, while candidate non-edges can be introduced when their multimodal similarity is high. However, it cannot learn flexible, pair-specific deviations from the observed topology when cosine similarity alone is insufficient to characterize the underlying relationship. This motivates learning an explicit residual correction to the original graph.

3.1.2 Residual Refiner

The cosine similarity refiner relies on a fixed functional form and collapses feature interactions into a scalar. To capture richer, task-dependent relationships, we introduce a more expressive *residual refinement* mechanism.

Pairwise feature and residual scoring. For each pair (i,j), we construct a pairwise feature that preserves dimension-wise interaction patterns:

$$\varphi(i,j) = [\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_i^v, \mathbf{x}_j^v, \mathbf{x}_i^t \odot \mathbf{x}_j^t, \mathbf{x}_i^v \odot \mathbf{x}_j^v, |\mathbf{x}_i^t - \mathbf{x}_j^t|, |\mathbf{x}_i^v - \mathbf{x}_j^v|], \tag{5}$$

combining individual node features, multiplicative interactions, and modality-specific differences rather than collapsing them into a scalar. We pass the pairwise feature through a learned MLP scorer f_θ to produce a *residual edge score* $r(i,j) \in (-1,1)$, which indicates the strength and direction of the predicted topological correction:

$$r(i,j) = \tanh\big(f_\theta(\varphi(i,j))\big). \tag{6}$$

A positive $r(i,j)$ encourages edge addition or strengthening, while a negative value signals edge removal or weakening.

Coarse-to-fine candidate selection. The complexity $O(BN_\mathcal{B})$ can be still expensive when $N_\mathcal{B}$ is large in conjuction with the high-dim of $\varphi(i,j)$. We therefore adopt a coarse-to-fine strategy: **(i) Coarse filtering.** We project each node into a low-dimensional embedding: $\mathbf{h}_i = \mathbf{W}_c[\mathbf{x}_i^t; \mathbf{x}_i^v]$, where $\mathbf{W}_c \in \mathbb{R}^{d_h \times (d_t + d_v)}$ and $d_h \ll d_t + d_v$. For each target node $i \in \mathcal{B}$, we compute the cosine similarity $\tilde{S}(i,j) = \cos(\mathbf{h}_i, \mathbf{h}_j)$ over all $j \in \mathcal{V}_\mathcal{B}$ and retain the top-c candidates, where $c = \eta k$ with overselection ratio $\eta > 1$. **(ii) Fine scoring.** We apply the MLP scorer to candidate pairs (i,j) with $j \in \mathcal{C}(i)$ and select the final top-k neighbors. Formally:

$$\textbf{Coarse:} \quad \mathcal{C}(i) = \text{TopK}_{j \in \mathcal{V}_\mathcal{B}} \tilde{S}(i,j), \quad \textbf{Fine:} \quad \mathcal{N}_{\text{sem}}(i) = \text{TopK}_{j \in \mathcal{C}(i)} r(i,j). \tag{7}$$

Topology integration. We then integrate the MLP-based scores into the original topology as a residual signal:

$$\mathbf{A}_{\text{refine}}(i,j) = \text{Clamp}\big(\mathbf{A}_{\text{orig}}(i,j) + \delta \cdot r(i,j),\ 0,\ 1\big), \tag{8}$$

where $\delta = \sigma(\delta_{\text{logit}}) \in (0,1)$ is a learnable scalar that controls the refinement magnitude. Unlike Eq. (4), Eq. (8) directly parameterizes the deviation from the observed graph. The signed residual $r(i,j)$ determines the direction and magnitude of this pair-specific correction, allowing the model to selectively suppress unreliable observed connections and recover missing semantic ones.

Effect of refinement. Fig. 2 compares multimodal similarity distributions on Reddit-M. In the original graph, connected and disconnected pairs overlap substantially, which indicates noisy and missing edges; the edges added by the refiner concentrate in the high-similarity region, with much less overlap. This demonstrates that the refiner selectively recovers semantically meaningful connections while avoiding spurious links, leading to a cleaner and more informative graph structure. Appendix I

gives more detailed analysis for the residual refiner (Fig. 3). Proposition 1 shows that the family of $\mathbf{A}_{\text{refine}}$ functions induced by Eq. 8 strictly contains those induced by Eq. 4.

Proposition 1 (**Strict Generalization of Residual Refinement**). *On a fixed candidate set $\mathcal{C} \subseteq V \times V$, let $\mathcal{F}_{\text{sim}}$ denote the family of edge-scoring functions induced by the similarity refiner (Eqn. 2), and let $\mathcal{F}_{\text{res}}$ denote the family induced by the residual refiner (Eq. 6). Assume the similarity refiner's correction magnitude is bounded by δ on $\mathcal{C}$. Then $\mathcal{F}_{\text{sim}} \subsetneq \mathcal{F}_{\text{res}}$.*

Remark 1. *Proposition 1 characterizes* expressiveness*, not generalization. The inclusion follows from a universal-approximation argument on the residual MLP, and strictness from dimension-wise interactions (e.g., $\mathbf{x}_i^t \odot \mathbf{x}_j^t$) that cosine similarity collapses to a scalar. Whether this additional capacity translates to better task performance depends on the learning context: Sec. 5.3 show that the residual refiner's advantage is not consistent without cross-modal propagation, and that the residual refiner gains more from coupling than the similarity refiner on Fashion and Reddit-S (Table 4).*

3.2 Semantic Alignment via Cross-Modal Message Passing

Given the refined graph $\mathbf{A}_{\text{refine}}$, we propagate information jointly across modalities and graph structure. We first present the propagation rule directly through its supergraph formulation, which exposes its spectral and alignment properties (Sec. 4), and then show that it admits an efficient per-modality implementation.

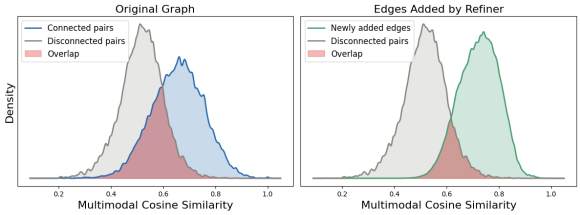


Figure 2: **Topology–semantics alignment before and after refinement. Left**: distribution of multimodal cosine similarity for connected and disconnected node pairs in the original graph (RedditM). **Right**: similarity distribution of edges added by the cosine refiner compared to disconnected pairs.

We first project modality-specific features into a common latent space of dimension d_h:

$$\mathbf{h}_i^{m,(0)} = \text{Proj}(\mathbf{x}_i^m; \phi^m), \quad m \in \{t, v\}, \tag{9}$$

where ϕ^m parametrizes the projection (a linear layer followed by layer normalization (LN) and the GELU nonlinearity with dropout). This ensures dimensional alignment and stabilizes subsequent cross-modal interaction. We then adopt symmetric normalization of $\mathbf{A}_{\text{refine}}$ with self-loops.

Coupled propagation on a modality-augmented supergraph. We define the stacked representation $\mathbf{H}^{(\ell)} = [\mathbf{H}^{t,(\ell)}; \mathbf{H}^{v,(\ell)}]$ and the supergraph adjacency

$$\mathbf{A}_{\text{sup}} = \begin{bmatrix} \mathbf{A}_{\text{refine}} & \gamma\mathbf{I} + \rho\mathbf{A}_{\text{refine}} \\ \gamma\mathbf{I} + \rho\mathbf{A}_{\text{refine}} & \mathbf{A}_{\text{refine}} \end{bmatrix}, \tag{10}$$

where $\gamma, \rho \in (0, 1)$ are **learnable** coupling scalars (parameterized via sigmoid). Propagation then takes the standard form

$$\mathbf{H}^{(\ell+1)} = \sigma\big(\mathbf{A}_{\text{sup}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\big), \tag{11}$$

with a *single shared* weight matrix $\mathbf{W}^{(\ell)}$ acting on both modality blocks. Two design choices are central: (1) **The symmetric block construction**(identical diagonals and off-diagonals) places the two modalities on equal footing and enables decomposition into symmetric and anti-symmetric eigenmodes (Corollary 1). (2) **Sharing $\mathbf{W}^{(\ell)}$ across modalities** is what enables the cross-modal contraction guarantee in Theorem 2; a separate-weight variant would introduce a residual term proportional to $\|\mathbf{W}_T^{(\ell)} - \mathbf{W}_V^{(\ell)}\|$ that does not vanish even when the two modalities have aligned.

Per-modality view. Expanding the block product in Eq. 11 recovers an explicit per-modality update. For the textual stream:

$$\mathbf{h}_i^{t,(\ell+1)} = \sigma\Bigg(\Big(\underbrace{\sum_{j\in\mathcal{N}(i)} A_{ij}\,\mathbf{h}_j^{t,(\ell)}}_{\text{intra-modal aggregation}} + \underbrace{\gamma\cdot\mathbf{h}_i^{v,(\ell)}}_{\text{cross-modal self-injection}} + \underbrace{\rho\cdot\sum_{j\in\mathcal{N}(i)} A_{ij}\,\mathbf{h}_j^{v,(\ell)}}_{\text{cross-modal neighbor aggregation}}\Big)\cdot\mathbf{W}^{(\ell)}\Bigg), \tag{12}$$

where $A_{ij} = \mathbf{A}_{\text{refine}}(i, j)$ (Eqn. 4 or 8) and $\mathcal{N}(i)$ is the neighborhood of i in the refined graph, i.e., its retained original neighbors together with its top-k semantic neighbors (Eqn. 3 or Eqn. 7). The visual update $\mathbf{h}_i^{v,(\ell+1)}$ is defined symmetrically. Each term has a distinct role: intra-modal aggregation propagates within a stream, self-injection mixes the node's own counterpart-modality representation,

and cross-modal neighbor aggregation propagates counterpart-modality signals over the refined graph. Importantly, this formulation reduces the number of nodes by half and decreases the number of edges from $4E + 2N$ in Eq. 11 to $2E$, making it more efficient.

Complexity optimization. The per-modality view above avoids materializing $\mathbf{A}_{\text{sup}}$, but still requires four GNN calls per layer (two per modality: one for intra-modal and one for cross-modal neighbor aggregation). By exploiting linearity of aggregation,

$$\sum_j A_{ij}(\mathbf{h}_j^t + \rho \mathbf{h}_j^v) = \sum_j A_{ij}\mathbf{h}_j^t + \rho \sum_j A_{ij}\mathbf{h}_j^v, \tag{13}$$

we merge intra-modal and cross-modal neighbor aggregation into a single GNN call per modality:

$$\tilde{\mathbf{h}}_i^{t,(\ell)} = \text{GNN}(\mathbf{h}_j^{t,(\ell)} + \rho\, \mathbf{h}_j^{v,(\ell)},\, \mathbf{A}_{\text{refine}}), \qquad \tilde{\mathbf{h}}_i^{v,(\ell)} = \text{GNN}(\mathbf{h}_j^{v,(\ell)} + \rho\, \mathbf{h}_j^{t,(\ell)},\, \mathbf{A}_{\text{refine}}), \tag{14}$$

followed by the self-injection term:

$$\mathbf{h}_i^{t,(\ell+1)} = \sigma\Big(\text{LN}\big(\tilde{\mathbf{h}}_i^{t,(\ell)} + \gamma\, \mathbf{h}_i^{v,(\ell)}\mathbf{W}^{(\ell)}\big)\Big), \quad \mathbf{h}_i^{v,(\ell+1)} = \sigma\Big(\text{LN}\big(\tilde{\mathbf{h}}_i^{v,(\ell)} + \gamma\, \mathbf{h}_i^{t,(\ell)}\mathbf{W}^{(\ell)}\big)\Big). \tag{15}$$

This reduces the cost from four to two GNN calls per layer. For any linear aggregation operator, the merged aggregation is exactly equivalent to Eq. 11; the implementation additionally applies LayerNorm for training stability, which is omitted from the analysis in Sec. 4. A detailed complexity optimization analysis is provided in Appendix D.

Closed loop with the refiner. Combined with the graph refiner of Sec. 3.1, the propagation step closes a loop: at each training step, refined representations sharpen the topology produced by Eq. 4/8, and the updated topology in turn shapes the next round of cross-modal propagation. This is the reciprocal co-evolution formalized in Proposition 2 in Appendix M: fixed refiners (i.e., not jointly optimized with the GNN) are recovered as special cases, while ReCoG additionally allows task gradients to update the topology refiner, strictly enlarging the feasible optimization space.

3.3 Adaptive Modality Fusion

After L layers of coupled message passing, we obtain modality-specific node representations $\mathbf{z}_i^t = \mathbf{h}_i^{t,(L)}$ and $\mathbf{z}_i^v = \mathbf{h}_i^{v,(L)}$. While the two modalities have been progressively aligned through shared propagation and cross-modal interaction, they may still carry complementary or unevenly informative signals. We therefore fuse them into a unified representation through a *node-adaptive gating mechanism*. Specifically, we compute a node-specific scalar gate:

$$g_i = \sigma\Big(\text{MLP}\big([\mathbf{z}_i^t\,;\, \mathbf{z}_i^v]\big)\Big) \in (0, 1), \tag{16}$$

where $[\cdot;\cdot]$ denotes concatenation. The final representation is:

$$\mathbf{z}_i = \text{LN}\Big(g_i \cdot \mathbf{z}_i^t + (1 - g_i) \cdot \mathbf{z}_i^v\Big). \tag{17}$$

Relation to coupled propagation. Unlike early or late fusion strategies, this design separates two roles: (i) *Cross-modal message passing* aligns and propagates multimodal information across the graph, and (ii) *Adaptive fusion* performs final integration based on node-specific relevance.

This separation allows ReCoG to first establish structurally grounded cross-modal representations and then perform flexible, task-adaptive combination. This final adaptive fusion ensures that the benefits of reciprocal co-evolution are translated into task-specific representations.

4 Theoretical Analysis

We provide formal justifications for the design choices in ReCoG from two complementary perspectives: the representational benefit of coupled cross-modal propagation, and the cross-modal alignment guarantee of shared-weight design. All proofs including propositions are deferred to Appendix L- P.

Theorem 1 (**Layerwise Cross-Modal Information Flow**). *Define a multimodal GNN as* decoupled *if each modality stream's update at layer ℓ depends only on that stream's representations from layer $\ell - 1$ and the graph A, with arbitrary per-modality weight matrices $W_T^{(\ell)}, W_V^{(\ell)}$. Then the cross-modal Jacobian satisfies:*

$$\frac{\partial\, H_T^{(\ell)}}{\partial\, H_V^{(0)}} = \begin{cases} 0 & \text{for any decoupled model, } \forall\, \ell \geq 0, \\ \neq 0 \text{ (generically)} & \text{for the coupled model with } \rho > 0 \text{ or } \gamma > 0,\ \forall\, \ell \geq 1. \end{cases} \tag{18}$$

Interpretation. Theorem 1 establishes a structural rather than parameter-count distinction: layerwise cross-modal information flow is impossible in decoupled or late-fusion architectures, regardless of per-modality parameterization or model capacity, because they cannot refine one modality using another's graph neighbors during propagation.

The supergraph interpretation (Eq. 10) enables an explicit spectral characterization of how the coupling coefficients reshape the effective frequency response.

Corollary 1 (**Spectral Characterization of Coupled Propagation**). *Let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_N$ be the eigenvalues of A_{refine}. Then the $2N$ eigenvalues of A_{sup} are*

$$\lambda_i^+ = (1+\rho)\,\mu_i + \gamma, \qquad \lambda_i^- = (1-\rho)\,\mu_i - \gamma, \qquad i = 1, \ldots, N, \tag{19}$$

where λ_i^+ correspond to eigenmodes in which text and vision components are aligned, and λ_i^- to eigenmodes in which they oppose. Intuitively, ρ and γ act as a learned spectral filter that amplify symmetric modes and attenuate anti-symmetric modes.

Theorem 2 (**Shared-Weight Cross-Modal Consistency**). *Consider one coupled layer with shared weight $W^{(\ell)}$ and 1-Lipschitz activation $\sigma(\cdot)$. Define the cross-modal discrepancy $\Delta^{(\ell)} = H_T^{(\ell)} - H_V^{(\ell)}$. Then*

$$\left\|\Delta^{(\ell+1)}\right\| \;\leq\; \left\|W^{(\ell)}\right\| \Big((1-\rho)\left\|A_{\text{refine}}\right\| + \gamma \Big) \left\|\Delta^{(\ell)}\right\|. \tag{20}$$

Consequently, after L layers:

$$\left\|\Delta^{(L)}\right\| \;\leq\; \prod_{\ell=0}^{L-1} \left\|W^{(\ell)}\right\| \Big((1-\rho)\left\|A_{\text{refine}}\right\| + \gamma \Big) \;\cdot\; \left\|\Delta^{(0)}\right\|. \tag{21}$$

When the per-layer factor $\|W^{(\ell)}\|\big((1-\rho)\|A_{\text{refine}}\|+\gamma\big) < 1$, the discrepancy contracts exponentially with depth, ensuring that text and vision embeddings converge to a shared representation space.

5 Experiments

Datasets. We evaluate ReCoG on two tasks: node classification (NC) and link prediction (LP) across **nine** multimodal graph datasets spanning diverse domains. For NC, we consider two product-review graphs, **Ele-Fashion** and **Goodreads-NC** (Zhu et al., 2025), and five datasets from the MAGB benchmark (Yan et al., 2025), including three e-commerce graphs (**Movies**, **Toys**, **Grocery**) and two social-interaction graphs (**Reddit-S**, **Reddit-M**). For LP, we use two Amazon co-purchase graphs, **Amazon-Sports** and **Amazon-Cloth** (Zhu et al., 2025). Each node is associated with a text description and an image, from which pretrained encoders extract modality-specific feature vectors. Detailed dataset statistics are provided in Table 12 in the Appendix. All experiments were conducted on a single NVIDIA A100 GPU.

Encoders. To ensure fair and comprehensive evaluation, we follow benchmark protocols and consider multiple encoder configurations. For Ele-Fashion, Goodreads-NC, and LP datasets, we use four combinations: *CLIP*, *T5+ViT*, *ImageBind*, and *T5+DINOv2*. CLIP and ImageBind produce aligned multimodal embeddings, while ViT/DINOv2 serve as vision encoders and T5 as the text encoder.

For MAGB datasets, we adopt three configurations: *L+C* (LLaMA-3.1-8B + CLIP), *QWen* (QWen2-VL-7B), and *LLaVL* (LLaMA-3.2-11B-Vision). A key distinction is that QWen and LLaVL baselines directly use *jointly pretrained multimodal embeddings*, where text and image representations are inherently aligned. In contrast, ReCoG treats text and image features as *separate streams* and learns cross-modal alignment during training. As shown in our results, this learned alignment consistently matches or surpasses encoder-level alignment, highlighting the effectiveness of our architecture.

Baselines. Our baselines fall into two groups: (i) multimodal GNNs that operate on fixed graphs (MMGCN and MGAT (Zhu et al., 2025) with late fusion, GCN (Kipf & Welling, 2016)/SAGE (Hamilton et al., 2017)/MLP on concatenated features), and (ii) recent multimodal graph models, including *NSG-MoE* (Zhang & Kuruoglu, 2026), *LION* (Li et al., 2026), *PLANET* (Liu et al., 2026), and *UniGraph2* (He et al., 2025). Note that, following the settings of major baselines and benchmarks (Zhu et al., 2025; Yan et al., 2025), we focus on multimodal-aware baselines, as most structure learning methods are designed to enhance single-modality graphs without semantic consideration and would require nontrivial adaptation to handle cross-modal evidence fairly.

Evaluation. We evaluate two variants of our model: **ReCoG-sim** (similarity refiner) and **ReCoG-res** (residual refiner), both using the same cross-modal message passing backbone. Each variant is tested

Table 1: Node classification accuracy (%) on the MM-Graph (Zhu et al., 2025). Best and runner-up are highlighted.

		Ele-fashion				Goodreads-NC			
	Model	CLIP	T5+ViT	IBind	T5+DINO	CLIP	T5+ViT	IBind	T5+DINO
	MMGCN	86.10 ± 0.50	82.39 ± 0.30	86.21 ± 0.94	85.53 ± 0.33	83.29 ± 0.20	81.85 ± 0.22	80.58 ± 1.08	82.44 ± 0.11
	MGAT	84.66 ± 0.29	84.01 ± 0.08	86.12 ± 0.08	84.54 ± 0.27	76.48 ± 0.59	75.43 ± 0.76	69.45 ± 6.25	74.98 ± 1.23
	GCN	79.83 ± 0.03	79.63 ± 0.07	80.35 ± 0.02	79.37 ± 0.04	81.61 ± 0.01	81.67 ± 0.03	78.91 ± 0.04	81.71 ± 0.03
	SAGE	87.10 ± 0.02	84.41 ± 0.09	87.71 ± 0.13	85.31 ± 0.09	83.30 ± 0.02	83.03 ± 0.04	80.39 ± 0.21	82.99 ± 0.08
	MLP	85.16 ± 0.03	84.98 ± 0.05	88.73 ± 0.01	84.87 ± 0.01	72.29 ± 0.02	67.82 ± 0.07	58.75 ± 0.05	68.83 ± 0.03
sim	ReCoG-GCN	88.56 ± 0.05	87.25 ± 0.08	89.01 ± 0.03	87.70 ± 0.06	90.43 ± 0.09	88.18 ± 0.10	89.26 ± 0.08	89.21 ± 0.12
sim	ReCoG-SAGE	89.01 ± 0.05	88.02 ± 0.07	89.31 ± 0.10	88.21 ± 0.10	90.51 ± 0.08	88.24 ± 0.11	89.31 ± 0.09	89.30 ± 0.10
res	ReCoG-GCN	88.60 ± 0.08	87.99 ± 0.09	89.06 ± 0.10	88.28 ± 0.08	90.68 ± 0.06	90.98 ± 0.03	90.05 ± 0.03	90.19 ± 0.06
res	ReCoG-SAGE	88.92 ± 0.07	88.04 ± 0.06	89.51 ± 0.03	88.34 ± 0.06	90.66 ± 0.11	90.07 ± 0.11	90.25 ± 0.03	90.08 ± 0.05

Table 2: Node classification accuracy (%) on the MAGB datasets (Yan et al., 2025).

		Movies			Toys			Grocery			Reddit-S			Reddit-M		
	Model	L+C	QWen	LLaVL	L+C	QWen	LLaVL	L+C	QWen	LLaVL	L+C	QWen	LLaVL	L+C	QWen	LLaVL
	GCN	53.98	55.19	54.92	81.59	80.95	81.57	84.64	82.96	84.17	89.77	90.88	91.13	71.42	72.96	75.64
	SAGE	56.08	58.01	57.39	82.35	81.45	82.35	86.65	85.55	86.81	91.08	92.57	92.94	80.84	81.23	85.61
sim	ReCoG-GCN	56.97	57.72	57.51	81.98	81.20	82.19	85.89	83.46	85.03	97.20	96.98	97.26	90.55	90.24	90.87
sim	ReCoG-SAGE	56.94	58.92	60.00	82.58	82.27	82.75	86.86	86.24	89.05	96.76	96.38	96.73	88.75	89.18	90.11
res	ReCoG-GCN	57.45	60.12	58.62	82.07	82.27	82.27	86.18	86.39	86.56	96.89	96.82	96.73	87.28	87.31	87.42
res	ReCoG-SAGE	57.93	60.84	60.45	82.68	82.36	83.18	86.92	86.56	87.53	96.89	96.48	97.17	89.18	89.71	89.86

† Following the MAGB benchmark protocol Yan et al. (2025), we report the average over 10 random seeds.

with both GCN-style and SAGE-style convolutions. We follow the same data splits and experimental settings as prior work, including multiple runs with different random seeds. For fair comparison, we reproduce results for *MMGCN*, *MGAT*, and *MLP*, while reporting official results for recent state-of-the-art methods. For models with substantially different architectures (e.g., *UniGraph2*), we use the best reported results from their original papers.

5.1 Node Classification Results

Tables 1 and 2 report node classification accuracy across all datasets and encoder configurations. Table 3 compares the best performance between our ReCoG and SOTA multimodal graph models.

Overall performance. ReCoG achieves consistent overall improvements across datasets and encoders, with the SAGE-style variants outperforming the corresponding baselines in all settings and only a few small exceptions for the GCN-style variants.

Key observations. (i) **Strong gains on large scale datasets.** On Goodreads-NC, ReCoG-res achieves 90.98%, outperforming the strongest recent baseline (PLANET, 84.16%) by a large margin. On Reddit-M, ReCoG improves over GCN by more than 15%, demonstrating substantial robustness. We attribute this to the fact that large-scale interaction graphs tend to contain a higher proportion of semantically irrelevant edges, and the refiner mitigates this issue by down-weighting noisy edges and introducing semantically coherent ones. (ii) **Consistent improvements across encoders.** The best ReCoG variant improves over all baselines under every encoder, including CLIP, ImageBind,

Table 3: Comparison with recent multimodal graph methods. R-S = Reddit-S, R-M = Reddit-M.

	Node Classification (Acc. %)							Link Pred. (MRR %)	
Model	Fashion	GoodReads	Movies	Toys	Grocery	R-S	R-M	Sports	Cloth
NSG-MoE	88.42	—	53.98	80.92	86.28	95.93	85.03	34.39	24.10
LION	—	78.54	58.61	—	88.30	94.80	—	—	—
PLANET	87.37	84.16	57.06	81.22	85.16	96.62	—	27.51	20.25
UniGraph2	87.91	81.97	*52.92	*79.49	*85.24	*93.50	*70.19	31.61	25.01
ReCoG-sim	89.31	90.51	60.00	82.75	89.05	97.26	90.87	38.53	28.60
ReCoG-res	89.51	90.98	60.84	83.18	87.53	97.17	89.86	38.57	28.73

* Results were not reported in the original paper, we use the numbers from (Zhang & Kuruoglu, 2026; Li et al., 2026).
"—" indicates the method was not evaluated on that dataset.

QWen, and LLaVL, so the gains are not tied to a specific feature extractor. (iii) **Graph-level alignment surpasses encoder-level alignment.** Even when using VLM embeddings that are already aligned (e.g., QWen, LLaVL), ReCoG still achieves significant gains (e.g., $+6.10\%$ on Reddit-S and $+17.28\%$ on Reddit-M).

Effect of refiner variants. The residual refiner generally achieves the best performance, suggesting that its expressive edge scoring better captures complex multimodal relationships. However, the similarity refiner remains competitive and sometimes preferable when pretrained embedding geometry is already informative, due to its strong inductive bias and lower risk of overfitting.

5.2 Link Prediction Results

Table 5 (Appendix C) reports results on Amazon-Sports and Amazon-Cloth. ReCoG consistently improves MRR across encoder configurations, with gains up to 4.74 points on Amazon-Sports and 3.53 points on Amazon-Cloth over the strongest baseline under the same features. Improvements are also pronounced in Hits@10, suggesting that topology refinement helps recover more high-quality candidate links. The two refiners perform similarly overall: ReCoG-sim is slightly stronger on Amazon-Sports, while ReCoG-res has a small advantage on Amazon-Cloth. Please refer to Appendix C for detailed analysis.

5.3 Ablation Studies

Synergistic Effect. We analyze the contribution of each component in ReCoG by comparing the full model with variants that remove (i) the graph refiner, (ii) cross-modal message passing, or (iii) both. The variant without both components reduces to independent modality-specific GNNs with late fusion. Experiments are conducted on three datasets using GCN as the backbone. Results are reported in Table 4.

Table 4: Ablation on node classification(%). † *MMGCN* uses separate GCNs for each modality with late fusion.

	Configuration	Fashion (Clip)	Movies (L+C)	Reddit-S (L+C)
	MMGCN† (w/o both)	86.10	54.18	91.13
sim	w/o Cross-Modal	86.83	55.16	94.28
sim	w/o Refiner	87.09	56.19	95.97
sim	Full Model	88.56	56.97	97.20
res	w/o Cross-Modal	86.41	55.74	92.74
res	w/o Refiner	87.09	56.19	95.97
res	Full Model	88.60	57.45	96.89

Overall effect of components. Removing either topology refinement or cross-modal propagation consistently degrades performance, while removing both yields the largest drop. Without coupling, the residual refiner shows no consistent advantage over the similarity refiner despite its more flexible pair-specific scoring. With cross-modal propagation, however, the residual refiner gains more from coupling on Fashion and Reddit-S, while the two refiners benefit similarly on Movies. This pattern suggests that cross-modal propagation provides richer task-dependent signals that help the more expressive residual refiner learn topology corrections better aligned with mixed-modality message passing. These results highlight the complementary roles of topology refinement and cross-modal propagation in ReCoG.

Other Ablations. Additional controlled studies of multimodal refinement and propagation, hyperparameter sensitivity, stronger GNN backbones, and computational efficiency are provided in Appendices G, E–F, H , and J– K, respectively.

6 Conclusion

We propose ReCoG, a co-evolution framework that jointly refines graph topology and learns cross-modal node representations end-to-end for multimodal attributed graphs. Unlike prior methods that operate on fixed topology or restrict modality interaction to late fusion, ReCoG enables graph structure and multimodal representations to mutually reinforce each other throughout training through a multimodal graph refiner and coupled cross-modal message passing. Extensive experiments on nine datasets spanning node classification and link prediction demonstrate consistent improvements over classical GNNs and recent multimodal graph models. Our results further show that cross-modal coupling during graph propagation complements the alignment provided by jointly pretrained VLMs.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735. PMLR, 2020a.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Dongzhe Fan, Jiajin Liu, Yi Fang, Djellel Difallah, and Qiaoyu Tan. MLaGA: Multimodal large language and graph assistant. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1027–1038, 2026.

Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. SLAPS: Self-supervision improves structure learning for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15180–15190, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

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.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pp. 2704–2710, 2020.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 66–74, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Xunkai Li, Zhengyu Wu, Zekai Chen, Henan Sun, Daohan Su, Guang Zeng, Hongchao Qin, Rong-Hua Li, and Guoren Wang. Lion: A clifford neural paradigm for multimodal-attributed graph learning. *arXiv preprint arXiv:2601.21453*, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

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.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. Graph neural networks for friend ranking in large-scale social platforms. In *Proceedings of the Web Conference 2021*, pp. 2535–2546, 2021.

Xianfeng Tang, Yozen Liu, Neil Shah, Xiaolin Shi, Prasenjit Mitra, and Suhang Wang. Knowing your fate: Friendship, action and temporal explanations for user engagement prediction on social apps. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2269–2279, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pp. 2022–2032, 2019.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

Qitian Wu, Wentao Zhao, Zenan Li, David P. Wipf, and Junchi Yan. NodeFormer: A scalable graph structure learning transformer for node classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022b.

Rui Xue and Tianfu Wu. H^3GNNs: Harmonizing heterophily and homophily in GNNs via joint structural node encoding and self-supervised learning. *arXiv preprint arXiv:2504.11699*, 2025.

Rui Xue, Haoyu Han, MohamadAli Torkamani, Jian Pei, and Xiaorui Liu. Lazygnn: Large-scale graph neural networks via lazy propagation. In *International Conference on Machine Learning*, pp. 38926–38937. PMLR, 2023a.

Rui Xue, Xipeng Shen, Ruozhou Yu, and Xiaorui Liu. Efficient large language models fine-tuning on graphs. *arXiv preprint arXiv:2312.04737*, 2023b.

Hao Yan, Chaozhuo Li, Jun Yin, Zhigang Yu, Weihao Han, Mingzheng Li, Zhengxin Zeng, Hao Sun, and Senzhang Wang. When graph meets multimodal: benchmarking and meditating on multimodal attributed graph learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5842–5853, 2025.

Jiahao Zhang, Rui Xue, Wenqi Fan, Xin Xu, Qing Li, Jian Pei, and Xiaorui Liu. Linear-time graph neural networks for scalable recommendations. In *Proceedings of the ACM on Web Conference 2024*, pp. 3533–3544, 2024.

Yihan Zhang and Ercan E Kuruoglu. Modality as heterogeneity: Node splitting and graph rewiring for multimodal graph learning. *arXiv preprint arXiv:2602.00067*, 2026.

Jing Zhu, Yuhang Zhou, Shengyi Qian, Zhongmou He, Tong Zhao, Neil Shah, and Danai Koutra. Mosaic of modalities: A comprehensive benchmark for multimodal graph learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14215–14224, 2025.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

Yinlin Zhu, Xunkai Li, Di Wu, Wang Luo, Miao Hu, and Guocong Quan. TMTE: Effective multimodal graph learning with task-aware modality and topology co-evolution. *arXiv preprint arXiv:2603.27723*, 2026.

Reproducibility Statement

We provide detailed descriptions of the datasets, model architectures, training procedures, hyperparameters, and evaluation protocols in the main text and Appendix R. All experiments follow the corresponding benchmark splits and are conducted with multiple random seeds where applicable. We will release the implementation and configuration files upon acceptance to facilitate reproduction of our results.

AI Use Statement

Large language models were used only for language editing, including improving grammar, spelling, clarity, and word choice. They were not used to develop the research ideas, methodology, experimental design, or reported results.

A Broader Impacts

Our proposed ReCoG advances multimodal graph learning by enabling joint refinement of graph structure and cross-modal representations, which can improve performance in applications such as recommendation systems, social networks, and scientific discovery. By leveraging multimodal semantic signals to infer graph connectivity, the proposed approach can better capture meaningful relationships in complex data.

However, these capabilities also introduce risks. Since ReCoG actively modifies graph topology, it may introduce spurious or biased connections when multimodal features are noisy or misaligned, potentially amplifying existing biases in pretrained encoders or datasets. In applications involving user data, such as social or e-commerce graphs, this may lead to unfair recommendations or unintended profiling.

Additionally, the ability to infer latent relationships from multimodal signals raises privacy concerns, as the model may uncover sensitive or hidden connections between entities. Misuse in surveillance or user behavior modeling contexts is also possible.

Future work should explore bias mitigation, robustness to noisy modalities, and privacy-preserving mechanisms to ensure responsible deployment.

B Related Work

Graph Neural Networks. Graph Neural Networks (GNNs) have become a dominant paradigm for representation learning on graph-structured data. Early spectral and spatial methods, such as GCN (Kipf & Welling, 2016), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2017), and SGC (Wu et al., 2019), learn node representations by recursively aggregating information from local neighborhoods. These models have achieved strong performance on node classification, link prediction, and graph classification tasks by exploiting graph topology as an inductive bias. Subsequent studies further extend GNNs to heterogeneous/heterophilic graphs, graph transformers, and scalable graph learning settings (Hu et al., 2020; Wang et al., 2019; Rampášek et al., 2022; Zhu et al., 2020; Xue & Wu, 2025). Despite their success, most conventional GNNs assume that the observed topology is reliable and that node attributes are uni-modal. This assumption is often violated in practical graphs, where edges may be noisy or incomplete with respect to multimodal semantics, and node attributes come from various modalities such as text and vision. Directly applying standard GNNs to concatenated multimodal features or fixed graph structures may therefore propagate modality-specific noise and fail to capture cross-modal dependencies during message passing.

Graph Structure Learning. Graph structure learning (GSL) methods jointly adapt graph topology and node representations to mitigate noisy or incomplete observed structures. Representative approaches such as IDGL Chen et al. (2020b), SLAPS Fatemi et al. (2021), Pro-GNN Jin et al. (2020), and NodeFormer Wu et al. (2022a) learn or reconstruct graph connectivity from node representations before graph propagation. For example, IDGL iteratively refines the graph using similarities computed in a single node-representation space and subsequently performs standard single-stream GNN

propagation. However, these methods are primarily formulated for unimodal representation learning: even when multimodal features are provided through concatenation, modality-specific evidence is collapsed into a single representation space, without explicitly modeling cross-modal interactions during topology refinement or graph propagation. In contrast, ReCoG preserves modality-specific signals, learns signed pair-specific residual topology corrections from multimodal semantic interactions, and performs explicit cross-modal neighbor exchange throughout graph propagation.

Multimodal Representation Learning. Multimodal representation learning has been a central topic in recent years. It aims to encode signals from different modalities, such as text, images, audio, and video, into semantically aligned embedding spaces. Contrastive vision-language models, such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), learn transferable representations by aligning paired images and texts through large-scale contrastive pre-training. Later models further improve multimodal understanding and generation through stronger vision encoders, language models, cross-attention modules, and instruction tuning, including BLIP (Li et al., 2022; 2023), ImageBind (Girdhar et al., 2023), LLaVA (Liu et al., 2023), and Qwen-VL (Bai et al., 2023). These models provide powerful feature extractors and have been widely used to initialize multimodal node features. However, most multimodal representation learning methods focus on instance-level alignment and do not explicitly model graph topology. As a result, their pretrained embeddings are usually static, task-agnostic, and insensitive to downstream graph structures. In multimodal graphs, semantic relationships between nodes are jointly determined by modality content and graph connectivity. Therefore, simply using pretrained multimodal embeddings is insufficient: the model should further adapt modality alignment and information propagation according to graph topology and task supervision.

Multimodal Graph Learning. Multimodal graph learning extends graph representation learning to graphs whose nodes or edges are associated with multiple modalities. A straightforward solution is to concatenate multimodal features and feed them into conventional GNNs, or to encode each modality with separate GNN branches followed by late fusion. Representative multimodal graph methods, such as MMGCN and MGAT (Zhu et al., 2025), exploit modality-specific representations and attention-based fusion to combine heterogeneous signals. More recent studies attempt to build more general multimodal graph learning frameworks. UniGraph2 (He et al., 2025) builds a graph foundation modal to learn a unified embedding space for multimodal graphs using modality-specific encoders, GNNs, and mixture-of-experts alignment across domains and modalities. NSG-MoE (Zhang & Kuruoglu, 2026) treats modality as heterogeneity by splitting each node into modality-specific sub-nodes and routing heterogeneous message flows through structured experts. LION (Li et al., 2026) introduces a Clifford-algebra-based paradigm to perform modality alignment and adaptive holographic aggregation on multimodal-attributed graphs. PLANET (Liu et al., 2026) further studies multimodal graph foundation models and proposes topology-aware modality interaction and alignment through embedding-wise domain gating and discretized semantic representation spaces. Concurrent work TMTE (Zhu et al., 2026) also explores topology–representation co-evolution, but fuses modalities before graph propagation and restricts topology evolution to nonnegative similarity affinities, without explicit cross-modal neighbor exchange or signed residual edge correction, while also reporting lower performance on the overlapping benchmarks.

These methods demonstrate the importance of multimodal interaction and alignment in graph learning. However, many existing approaches either rely on fixed observed topology, perform modality interaction after independent propagation, or emphasize representation alignment without explicitly modeling the reciprocal relationship between graph structure and multimodal semantics, thereby limiting the effectiveness of graph-modality fusion and cross-modal alignment. Furthermore, recent LLM/VLM-based multimodal graph approaches such as Graph4MM (Ning et al., 2025) and MLaGA (Fan et al., 2026) incorporate large pretrained models or instruction-tuning into graph reasoning. These methods address a complementary setting to ReCoG, which focuses on graph-level topology refinement and cross-modal propagation over fixed pretrained node features.

Positioning of ReCoG. Different from prior work, our approach treats multimodal graph learning as a coupled topology-semantics co-evolution problem. First, we introduce a multimodal graph refiner that adaptively reweights original edges and adds semantic neighbors using joint textual and visual evidence, addressing the topology–semantics mismatch in the original graph. Second, we perform iterative cross-modal message passing over the refined topology, allowing textual and visual representations to align and exchange information during propagation rather than only at the final fusion stage. This design jointly improves graph structure and multimodal representations in

Table 5: Link prediction performance on Amazon-Sports and Amazon-Cloth. (Zhu et al., 2025). The Best and runner-up are highlighted.

	Encoder	Amazon-Sports			Amazon-Cloth		
		MRR	H@1	H@10	MRR	H@1	H@10
MMGCN	CLIP	31.96 ± 0.10	16.35 ± 0.11	68.46 ± 0.08	22.20 ± 0.05	10.76 ± 0.10	46.62 ± 0.12
	ViT+T5	30.33 ± 0.08	15.01 ± 0.05	66.41 ± 0.11	19.45 ± 0.34	9.22 ± 0.20	40.49 ± 0.61
	ImageBind	31.74 ± 0.21	16.45 ± 0.13	67.39 ± 0.74	24.72 ± 0.19	12.47 ± 0.09	51.32 ± 0.56
	DINO+T5	30.04 ± 0.27	14.98 ± 0.07	64.56 ± 0.56	21.77 ± 0.23	10.47 ± 0.12	45.81 ± 0.52
MGAT	CLIP	27.56 ± 0.30	13.55 ± 0.29	60.21 ± 0.21	21.38 ± 0.23	10.39 ± 0.22	44.60 ± 0.36
	ViT+T5	30.15 ± 0.34	15.28 ± 0.34	64.84 ± 0.41	20.59 ± 0.41	9.79 ± 0.30	43.44 ± 0.76
	ImageBind	30.15 ± 0.12	15.50 ± 0.05	64.20 ± 0.43	22.13 ± 0.27	10.96 ± 0.15	45.84 ± 0.57
	DINO+T5	28.91 ± 0.09	14.47 ± 0.18	62.11 ± 0.22	21.42 ± 0.13	10.38 ± 0.13	44.11 ± 0.50
GCN	CLIP	31.38 ± 0.08	16.58 ± 0.13	66.14 ± 0.08	22.28 ± 0.05	11.83 ± 0.04	43.52 ± 0.10
	ViT+T5	30.83 ± 0.07	16.31 ± 0.08	64.76 ± 0.15	21.60 ± 0.05	11.37 ± 0.05	42.29 ± 0.14
	ImageBind	31.67 ± 0.09	17.07 ± 0.14	65.61 ± 0.10	22.81 ± 0.02	12.27 ± 0.05	44.28 ± 0.09
	DINO+T5	30.42 ± 0.02	16.02 ± 0.03	64.02 ± 0.06	21.19 ± 0.04	11.09 ± 0.06	41.46 ± 0.16
SAGE	CLIP	33.83 ± 0.08	17.57 ± 0.14	71.90 ± 0.07	24.58 ± 0.13	12.16 ± 0.11	51.12 ± 0.09
	ViT+T5	32.01 ± 0.10	15.94 ± 0.17	69.84 ± 0.21	23.11 ± 0.05	11.10 ± 0.04	48.89 ± 0.09
	ImageBind	34.32 ± 0.11	17.87 ± 0.23	73.04 ± 0.15	25.20 ± 0.09	12.63 ± 0.05	52.53 ± 0.21
	DINO+T5	32.20 ± 0.12	16.19 ± 0.20	69.98 ± 0.32	22.98 ± 0.01	11.12 ± 0.04	48.28 ± 0.11
MLP	CLIP	28.22 ± 0.09	14.54 ± 0.16	59.40 ± 0.08	21.10 ± 0.04	10.70 ± 0.05	42.77 ± 0.05
	ViT+T5	24.81 ± 0.05	11.63 ± 0.05	54.78 ± 0.04	17.65 ± 0.06	8.14 ± 0.04	36.77 ± 0.06
	ImageBind	30.45 ± 0.14	15.91 ± 0.10	64.10 ± 0.07	22.18 ± 0.02	11.42 ± 0.04	44.86 ± 0.06
	DINO+T5	24.81 ± 0.16	11.62 ± 0.18	54.97 ± 0.22	17.53 ± 0.11	8.07 ± 0.09	36.53 ± 0.26
ReCoG-sim (GCN)	CLIP	38.53 ± 0.03	21.90 ± 0.05	75.11 ± 0.02	28.30 ± 0.03	15.64 ± 0.06	54.96 ± 0.05
	ViT+T5	36.82 ± 0.05	20.49 ± 0.03	73.36 ± 0.05	24.66 ± 0.05	12.94 ± 0.09	49.19 ± 0.05
	ImageBind	38.36 ± 0.02	21.95 ± 0.03	74.68 ± 0.09	28.60 ± 0.06	15.89 ± 0.08	55.71 ± 0.08
	DINO+T5	37.16 ± 0.05	21.29 ± 0.03	73.31 ± 0.02	27.10 ± 0.06	14.51 ± 0.06	54.02 ± 0.16
ReCoG-sim (SAGE)	CLIP	35.34 ± 0.03	18.66 ± 0.12	74.23 ± 0.08	27.02 ± 0.09	14.14 ± 0.18	54.38 ± 0.08
	ViT+T5	34.55 ± 0.18	18.31 ± 0.18	71.11 ± 0.10	25.38 ± 0.11	12.99 ± 0.09	51.78 ± 0.08
	ImageBind	36.90 ± 0.08	20.41 ± 0.06	75.03 ± 0.08	27.08 ± 0.06	14.48 ± 0.11	54.03 ± 0.12
	DINO+T5	35.36 ± 0.10	19.13 ± 0.11	72.40 ± 0.03	26.36 ± 0.03	14.00 ± 0.05	53.05 ± 0.03
ReCoG-res (GCN)	CLIP	38.57 ± 0.03	22.04 ± 0.05	75.37 ± 0.02	28.35 ± 0.06	15.64 ± 0.08	54.60 ± 0.05
	ViT+T5	37.28 ± 0.03	20.89 ± 0.03	74.06 ± 0.06	27.86 ± 0.05	15.32 ± 0.09	54.35 ± 0.05
	ImageBind	38.42 ± 0.09	21.73 ± 0.05	75.60 ± 0.03	28.73 ± 0.05	15.75 ± 0.08	55.85 ± 0.08
	DINO+T5	37.39 ± 0.06	21.29 ± 0.03	74.89 ± 0.01	27.78 ± 0.06	15.11 ± 0.05	54.54 ± 0.03
ReCoG-res (SAGE)	CLIP	36.22 ± 0.03	19.92 ± 0.12	73.74 ± 0.08	27.37 ± 0.09	14.49 ± 0.12	55.37 ± 0.02
	ViT+T5	35.07 ± 0.10	18.64 ± 0.18	73.40 ± 0.10	26.55 ± 0.10	13.72 ± 0.09	54.19 ± 0.05
	ImageBind	36.13 ± 0.08	19.86 ± 0.06	73.97 ± 0.12	27.62 ± 0.06	14.76 ± 0.11	55.37 ± 0.06
	DINO+T5	35.60 ± 0.05	18.81 ± 0.10	73.19 ± 0.05	25.95 ± 0.08	13.96 ± 0.05	54.08 ± 0.03

an end-to-end manner, providing a more task-adaptive alternative to fixed-topology GNNs, static multimodal encoders, and late-fusion multimodal graph models.

C Results for Link Prediction

We provide the full link prediction results in Table 5. A comprehensive analysis is presented below.

Key observations. *First*, ReCoG achieves substantial gains in MRR. On Amazon-Sports with CLIP features, ReCoG-res (GCN) reaches 38.57% MRR, compared to 33.83% from the strongest baseline under the same features (SAGE). On Amazon-Cloth, ReCoG-res (GCN) achieves 28.73% MRR with ImageBind features, outperforming SAGE (25.20%) by 3.53 points. *Second*, improvements are particularly pronounced in Hits@10. For example, on Amazon-Sports with ImageBind features, ReCoG-res (GCN) achieves 75.60% H@10 versus 65.61% for GCN. This suggests that the refined topology is especially effective at expanding the set of high-quality candidate links.

Variant comparison. The two refiner variants show similar trends to node classification but with smaller differences. ReCoG-sim performs slightly better on Amazon-Sports across most encoders, while ReCoG-res shows marginal advantages on Amazon-Cloth. This reduced gap indicates that, in link prediction, performance is more sensitive to the quality of the learned representations than to the specific refinement mechanism, since the decoder directly operates on embedding similarities.

D Complexity Optimization of Cross Modal Message Passing

We describe two successive optimizations that reduce computational cost without altering the output.

D.1 Optimization I: Supergraph Reformulation

Directly constructing $\mathbf{A}_{\text{sup}}$ requires $2N$ nodes and $4E + 2N$ edges. By computing each term in Eq. 12 independently, we avoid constructing the supergraph entirely. This decomposition requires four `conv` calls (two per modality), each operating on N nodes and E edges, plus two `lin` calls at $O(N \cdot d_h^2)$ each. The total per-layer cost is:

$$4 \times \underbrace{O(E \cdot d_h + N \cdot d_h^2)}_{\texttt{conv}} + 2 \times \underbrace{O(N \cdot d_h^2)}_{\texttt{lin}} \tag{22}$$

compared to the supergraph cost of $O((4E + 2N) \cdot d_h + 2N \cdot d_h^2)$ for a single `conv` over $2N$ nodes. The decomposed version halves the node count per call and eliminates the overhead of constructing the supergraph edge index (node ID offsets, feature concatenation, and additional $2N$ cross-modal identity edges).

D.2 Optimization II: Linear Merging of Terms 1 and 3

For GNN layers whose aggregation operator is linear in the input features, Terms 1 and 3 can be merged into a single `conv` call:

$$\texttt{conv}(\mathbf{h}_t^{(l)},\, \mathbf{A}) + \rho \cdot \texttt{conv}(\mathbf{h}_v^{(l)},\, \mathbf{A}) = \texttt{conv}\big(\mathbf{h}_t^{(l)} + \rho \cdot \mathbf{h}_v^{(l)},\, \mathbf{A}\big) \tag{23}$$

Without merging, computing Terms 1 and 3 for one modality requires two `conv` calls:

$$\underbrace{\texttt{conv}(\mathbf{h}_t^{(l)},\, \mathbf{A})}_{\text{Term 1}} + \rho \cdot \underbrace{\texttt{conv}(\mathbf{h}_v^{(l)},\, \mathbf{A})}_{\text{Term 3}} \quad \Rightarrow \quad 2 \times O(E \cdot d_h) + 2 \times O(N \cdot d_h^2) \tag{24}$$

After merging, the cost reduces to a single element-wise addition followed by one `conv` call:

$$\underbrace{\mathbf{h}_t^{(l)} + \rho \cdot \mathbf{h}_v^{(l)}}_{O(N \cdot d_h)} \longrightarrow \underbrace{\texttt{conv}(\,\cdot\,,\, \mathbf{A})}_{O(E \cdot d_h + N \cdot d_h^2)} \tag{25}$$

The $O(N \cdot d_h)$ addition is negligible compared to the `conv` cost, so merging effectively halves both the aggregation and linear transformation costs per modality. The complete per-layer update becomes:

$$\mathbf{h}_t^{(l+1)} = \text{LN}\Big(\texttt{conv}\big(\mathbf{h}_t^{(l)} + \rho\, \mathbf{h}_v^{(l)},\ \mathbf{A}\big) + \gamma \cdot \texttt{lin}(\mathbf{h}_v^{(l)})\Big) \tag{26}$$

$$\mathbf{h}_v^{(l+1)} = \text{LN}\Big(\texttt{conv}\big(\mathbf{h}_v^{(l)} + \rho\, \mathbf{h}_t^{(l)},\ \mathbf{A}\big) + \gamma \cdot \texttt{lin}(\mathbf{h}_t^{(l)})\Big) \tag{27}$$

requiring two `conv` calls and two `lin` calls per layer, yielding the total per-layer cost of $O(E \cdot d_h + N \cdot d_h^2)$ reported in Table 11.

Applicability. The merging in Eq. 23 requires linearity of the aggregation operator and holds for GCN, GraphSAGE-mean, and other linear-aggregation GNNs. For attention-based architectures such as GAT, where the aggregation weights depend nonlinearly on the input features, the merging does not hold exactly. In this case, we fall back to the decomposed implementation (four `conv` calls), which still avoids constructing the supergraph.

E Top-K Selection (Eqns. 3 and 7).

Table 6 examines the sensitivity of both the Residual and Cosine refiners to the number of candidate neighbors K. We use CLIP as the encoder and GCN as the backbone for all experiments. Across all

Table 6: Ablation study on the number of candidate neighbors K in the graph refiner.

Refiner	K	Node Classification (%)		Link Prediction (MRR %)	
		Ele-Fashion	Goodreads	Amazon-Sports	Amazon-Clothing
sim	1	87.60 $\pm$0.05	89.65 $\pm$0.05	36.72 $\pm$0.11	27.07 $\pm$0.10
	3	88.17 $\pm$0.03	90.43 $\pm$0.09	37.81 $\pm$0.06	27.93 $\pm$0.07
	5	88.56 $\pm$0.05	90.31 $\pm$0.05	38.53 $\pm$0.03	28.30 $\pm$0.03
	8	88.53 $\pm$0.05	90.16 $\pm$0.06	38.32 $\pm$0.09	28.01 $\pm$0.06
	12	87.95 $\pm$0.06	89.43 $\pm$0.10	37.23 $\pm$0.10	26.68 $\pm$0.05
res	1	87.88 $\pm$0.08	89.65 $\pm$0.08	37.36 $\pm$0.09	27.37 $\pm$0.07
	3	88.30 $\pm$0.06	90.68 $\pm$0.06	38.51 $\pm$0.05	28.35 $\pm$0.03
	5	88.60 $\pm$0.08	90.59 $\pm$0.05	38.57 $\pm$0.03	28.25 $\pm$0.02
	8	88.38 $\pm$0.03	90.66 $\pm$0.03	38.51 $\pm$0.08	28.02 $\pm$0.05
	12	87.85 $\pm$0.08	90.07 $\pm$0.05	37.70 $\pm$0.08	26.91 $\pm$0.06

Table 7: Effect of the overselection ratio η in the residual refiner ($k = 5$, CLIP features, GCN backbone). $\eta = 1$ disables the fine selection stage, since the coarse filter already returns exactly k candidates.

η	Ele-fashion (NC, Acc %)	Amazon-Sports (LP, MRR %)
1	87.96 $\pm$0.05	37.98 $\pm$ 0.05
3	88.60 $\pm$ 0.08	38.57 $\pm$ 0.03
5	88.36 $\pm$ 0.09	38.42 $\pm$ 0.01

datasets and both refiners, we observe a consistent pattern: performance improves as K increases from 1 to around 3–5, and then gradually degrades when $K = 12$.

When $K = 1$, the refiner has too few candidates to meaningfully adjust the graph topology, leading to weaker performance. Conversely, an overly large K may introduce semantically distant nodes as candidates, injecting noise into the refined adjacency and diluting the quality of the added edges. Notably, performance remains relatively stable in the range of $K = 3$ to $K = 5$, where the candidate pool is large enough to capture missing semantic neighbors while remaining small enough to avoid spurious connections. Based on these results, we set $K = 3$ or $K = 5$ as the default in all experiments for simplicity.

F Coarse Candidate Ratio

We investigate the effect of the coarse-to-fine candidate ratio η, which controls the number of candidates retained after coarse filtering, i.e., $c = \eta k$. We fix $k = 5$ and vary η, with the results reported in Table 7.

The results exhibit a clear trade-off. When η is too small, the coarse stage retains too few candidates, limiting the opportunity to recover informative missing connections. Conversely, an excessively large η introduces more weakly related candidates into the fine-scoring stage, which can increase noise in topology refinement. Performance is relatively stable around moderate values of η; therefore, we set $\eta = 3$ in our experiments.

G Effect of Multi-Modal Modeling

ReCoG preserves modality-specific information at two key stages: topology refinement and graph propagation. We conduct two complementary controlled studies to examine whether this design is necessary. First, we vary the edge scorer while keeping the remaining architecture fixed, testing whether multimodal pairwise interactions provide stronger refinement signals than single-modality or early-fused similarity functions. Second, we compare single-stream and modality-specific propagation over the same refined topology, isolating whether preserving separate modality streams remains beneficial after graph refinement. Together, these experiments examine the role of modality-specific modeling throughout the ReCoG pipeline.

Table 8: Effect of the edge scorer in the residual refiner (CLIP features for Amazon-Sports, L+ C for Movies, GCN backbone, $k = 5$, $\eta = 3$). NC: accuracy (%); LP: MRR (%).

Edge scorer	Movies (NC)	Amazon-Sports (LP)
Text-only f_θ	56.48	30.15
Vision-only f_θ	56.37	32.20
Projected cosine $\tilde{S}$	56.99	38.25
Full $\varphi(i,j)$ (ours)	57.45	38.57

G.1 Multimodal Information in Graph Refinement

To isolate how multimodal information is used during topology refinement, we compare the proposed residual scorer against single-modality and early-fusion alternatives while keeping all subsequent components fixed.

For the text-only variant, we retain only the textual blocks of the pairwise feature:

$$\varphi_t(i,j) = \left[x_i^t,\, x_j^t,\, x_i^t \odot x_j^t,\, |x_i^t - x_j^t|\right], \qquad r_t(i,j) = \tanh\big(f_\theta^t(\varphi_t(i,j))\big). \tag{28}$$

Similarly, the vision-only variant uses

$$\varphi_v(i,j) = \left[x_i^v,\, x_j^v,\, x_i^v \odot x_j^v,\, |x_i^v - x_j^v|\right], \qquad r_v(i,j) = \tanh(f_\theta^v(\varphi_v(i,j))). \tag{29}$$

These two variants directly test whether graph refinement can rely on a single modality alone. In contrast, the projected-cosine variant uses both modalities but collapses them into a single representation before computing pairwise similarity:

$$h_i = W_c[x_i^t; x_i^v], \qquad \widetilde{S}(i,j) = \cos(h_i, h_j), \qquad \widetilde{S}(i,j) \to A_{\text{refine}}. \tag{30}$$

The proposed residual scorer instead preserves modality-specific pairwise interactions:

$$\varphi(i,j) = \left[x_i^t, x_j^t, x_i^v, x_j^v, x_i^t \odot x_j^t, x_i^v \odot x_j^v, |x_i^t - x_j^t|, |x_i^v - x_j^v|\right], \tag{31}$$

$$r(i,j) = \tanh(f_\theta(\varphi(i,j))), \qquad A_{\text{refine}}(i,j) = \text{Clamp}\left(A_{\text{orig}}(i,j) + \delta r(i,j), 0, 1\right). \tag{32}$$

As shown in Table 8, using both modalities consistently outperforms either text-only or vision-only refinement, demonstrating that textual and visual signals provide complementary evidence for topology learning. The projected-cosine variant substantially narrows this gap by incorporating both modalities, but remains below the full residual scorer on both datasets. This suggests that the benefit of ReCoG does not arise merely from access to multimodal features; preserving modality-specific pairwise interactions in $\varphi(i,j)$ provides a more informative refinement signal than collapsing the two modalities into a single projected similarity.

G.2 Modality-Specific Information in Graph Propagation

We next examine whether preserving modality-specific streams remains beneficial after the refined topology has already been constructed. To isolate the propagation architecture, all variants use the same refined graph A_{refine} produced by the full residual refiner.

The single-stream baseline first concatenates the two modalities into one representation,

$$h_i^{(0)} = \text{Proj}\big([x_i^t; x_i^v]\big), \tag{33}$$

and then performs conventional graph propagation over the same **refined topology** used by ReCoG:

$$h_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} A_{\text{refine}}(i,j) h_j^{(\ell)} W^{(\ell)}\right). \tag{34}$$

Equivalently,

$$[x_i^t; x_i^v] \longrightarrow h_i \longrightarrow \text{GNN}(A_{\text{refine}}) \longrightarrow z_i. \tag{35}$$

Importantly, this is different from the GCN and GraphSAGE baselines reported in the main text, which propagate concatenated multimodal features over the *original observed graph* A_{orig}. By instead applying the single-stream GNN to the same A_{refine} as ReCoG, this control removes topology as a confounding factor and isolates the effect of the propagation architecture.

Table 9 further separates the effects of topology refinement and cross-modal propagation. Applying the refined topology to a conventional single-stream GCN improves accuracy from 79.83% to 85.15%, showing that topology refinement alone provides a substantial benefit. Importantly, under the same

Table 9: Propagation scheme versus topology (CLIP features, GCN backbone) on Ele-fashion.

Propagation	Observed graph	Refined graph
GCN	79.83	85.15
ReCoG-GCN	87.09	88.60

Table 10: Node classification accuracy (%) on Movies across GNN baselines.

Model	L+C	QWen	LLaVL
GCN	53.98	55.19	54.92
SAGE	56.08	58.01	57.39
GAT	55.42	51.92	56.26
GATv2	54.78	50.31	56.08
GCNII	53.56	52.05	54.50
ReCoG-sim (GCN)	56.97	57.72	57.51
ReCoG-sim (SAGE)	56.94	58.92	60.00
ReCoG-res (GCN)	57.45	60.12	58.62
ReCoG-res (SAGE)	57.93	60.84	60.45

refined graph, the coupled ReCoG propagation further improves performance to 88.60%. This controlled comparison shows that ReCoG's improvement cannot be explained by the refined topology alone: preserving modality-specific streams and enabling layer-wise cross-modal information exchange provides an additional benefit beyond graph refinement.

H GNN Baselines

In this section, we extend the comparison on Movies with GAT (Veličković et al., 2017), GATv2 (Brody et al., 2021), and GCNII (Chen et al., 2020a). All models consume the same frozen encoder features, concatenated across the two modalities, which is the standard single-stream alternative to modality-specific streams.

Table 10 shows that ReCoG outperforms all of these backbones under every encoder configuration. This is consistent with our main claim: refining the topology with cross-modal evidence and propagating modality-specific streams provide gains that a stronger single-stream aggregator over the observed graph does not recover.

I Topology–semantics Misalignment

Let the observed multimodal graph be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the observed edge set. Each node $i \in \mathcal{V}$ is associated with multimodal features $\mathbf{x}_i = [\mathbf{x}_i^t \,\|\, \mathbf{x}_i^v]$, where $\mathbf{x}_i^t \in \mathbb{R}^{d_t}$ and $\mathbf{x}_i^v \in \mathbb{R}^{d_v}$ denote textual and visual features, respectively. Denote by $\mathbf{A}_{\text{orig}}$ the original adjacency matrix, $\mathbf{A}_{\text{orig}}(i,j) = 1$ if $(i,j) \in \mathcal{E}$, 0 otherwise.

A fundamental challenge in multimodal graphs is that the observed topology is only partially aligned with multimodal semantics. To examine this issue, we analyze the multimodal semantic similarity distributions on Reddit-M (Fig. 2) and Movies (Yan et al., 2025). For a node pair (i,j), we compute the textual and visual cosine similarities as

$$S_T(i,j) = \cos(\mathbf{x}_i^t, \mathbf{x}_j^t), \qquad S_V(i,j) = \cos(\mathbf{x}_i^v, \mathbf{x}_j^v), \tag{36}$$

For simplicity, we plot the multimodal semantic similarity as the average

$$S_{MM}(i,j) = \frac{1}{2}\big(S_T(i,j) + S_V(i,j)\big), \tag{37}$$

where the encoded features are obtained from CLIP.

As shown on the left side of both Fig. 2 and Fig. 3, we compare the S_{MM} distributions of connected and disconnected node pairs in the original graph. Connected pairs generally exhibit higher multimodal similarity, indicating that the original topology provides useful structural priors. However, the two distributions show considerable overlap, revealing two types of topology–semantics discrepancy: *(i)* some connected pairs have low multimodal similarity, suggesting the presence of noisy edges; *(ii)* some disconnected pairs have high multimodal similarity, indicating missing semantic neighbors.

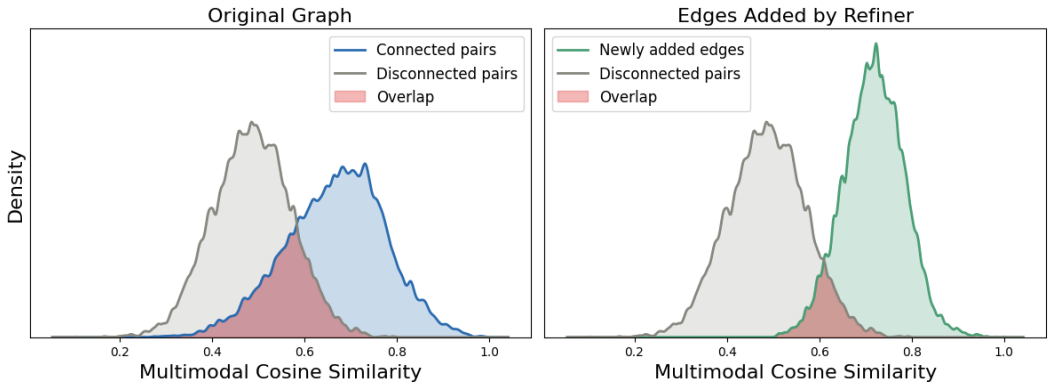


Figure 3: Topology–semantics alignment.

The right side of both Fig. 2 (Cosine Similarity Refiner) and Fig. 3 (Residual Refiner) illustrates the effect of the refiner. The newly added edges concentrate in a higher similarity range compared to the original connected pairs, and their overlap with the disconnected distribution is substantially reduced. This confirms that the refiner selectively recovers semantically meaningful missing edges without introducing topological noise.

These observations motivate a topology refinement mechanism that preserves the observed connectivity as a structural prior while adaptively suppressing noisy edges and recovering missing semantic neighbors based on joint textual and visual evidence.

J Complexity Analysis

We analyze the per-batch computational complexity of ReCoG (Table 11). Let N denote the number of nodes in the sampled subgraph, B the number of seed nodes, E the number of edges, k the number of new edges per seed, η the overselection ratio, $d = d_t + d_v$ the raw feature dimension, d_h the hidden dimension in the projection at coarse filtering, and L the number of GNN layers.

Table 11: Per-batch computational complexity of ReCoG major components.

Component	**Complexity**
Cosine similarity (S_T, S_V)	$O(BNd)$
Original edge scoring (MLP)	$O(E \cdot d \cdot d_h)$
Coarse filtering	$O(BNd_h)$
Fine scoring (MLP)	$O(B\eta k \cdot d \cdot d_h)$
Cross-modal message passing ($\times L$)	$O(L(E \cdot d_h + N \cdot d_h^2))$

K Efficiency Analysis

In addition to the theoretical complexity analysis in Appendix J, we evaluate the empirical computational overhead of ReCoG against SOTA multimodal graph baselines. Because the reported runtimes are obtained under different experimental environments, including different hardware, absolute training times are not directly comparable across papers. We therefore normalize the per-epoch training cost of each method by the MMGCN cost reported in the corresponding study under the same experimental setting, and additionally report the corresponding predictive accuracy to characterize the accuracy–efficiency trade-off.

As shown in Fig. 4, ReCoG incurs computational overhead of the same order as recent multimodal graph methods, while generally achieving stronger predictive performance. On Ele-Fashion, ReCoG-sim has a normalized training cost comparable to NSG-MoE while attaining higher accuracy. On Movies, ReCoG-res achieves higher accuracy than TMTE** while requiring lower normalized per-epoch training cost. Overall, these results show that the additional topology refinement and

cross-modal propagation in ReCoG provide a favorable accuracy–computation trade-off without introducing disproportionate computational overhead.

***Across the two datasets overlapping with TMTE, ReCoG-res also achieves higher reported accuracy under the comparable Qwen-based setting, with **60.84 vs. 60.31** on Movies and **86.56 vs. 84.18** on Grocery.*

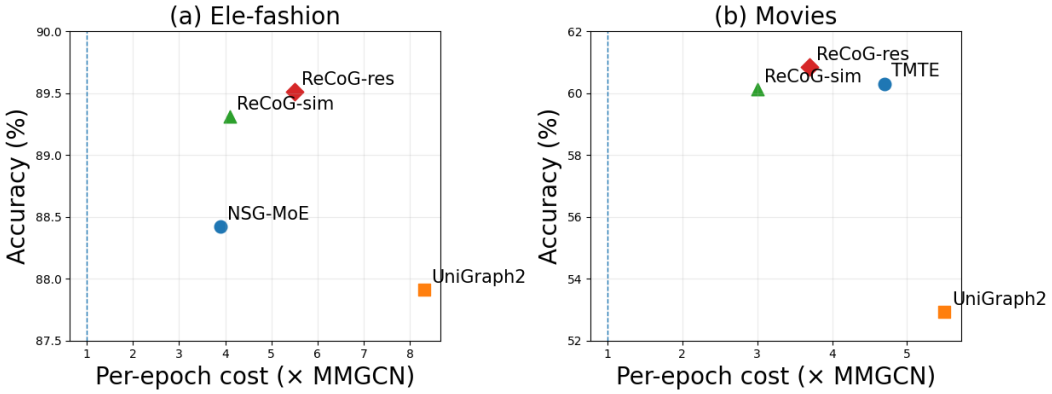


Figure 4: Model Efficiency.

L Proof of Proposition 1 (Strict Generalization of Residual Refinement)

We prove $\mathcal{F}_{\text{sim}} \subsetneq \mathcal{F}_{\text{res}}$ by establishing inclusion and then strictness. Recall that the similarity refiner induces the edge-scoring family

$$g_{\text{sim}}(i,j) = \alpha\, A_{\text{orig}}(i,j) + (1-\alpha)\big[\lambda\, S_V(i,j) + (1-\lambda)\, S_T(i,j)\big], \quad (i,j) \in \mathcal{C}, \tag{38}$$

with global coefficients $\alpha, \lambda \in [0,1]$, while the residual refiner induces

$$g_{\text{res}}(i,j) = A_{\text{orig}}(i,j) + \delta \cdot \tanh\big(f_\theta(\phi(i,j))\big), \quad (i,j) \in \mathcal{C}, \tag{39}$$

where $\phi(i,j)$ is the multimodal pair feature (Eq. 5) and f_θ is an MLP with sufficient width.

Part 1: $\mathcal{F}_{\text{sim}} \subseteq \mathcal{F}_{\text{res}}$. Take any $g_{\text{sim}} \in \mathcal{F}_{\text{sim}}$. For each $(i,j) \in \mathcal{C}$, define the target value

$$u(i,j) = \operatorname{arctanh}\Big(\tfrac{g_{\text{sim}}(i,j) - A_{\text{orig}}(i,j)}{\delta}\Big). \tag{40}$$

By the range assumption and $|g_{\text{sim}}(i,j) - A_{\text{orig}}(i,j)| < \delta$, the argument of $\operatorname{arctanh}$ lies in $(-1,1)$, so $u(i,j)$ is well-defined and finite for every $(i,j) \in \mathcal{C}$.

The similarity-based score depends continuously on $(S_T(i,j), S_V(i,j))$, which are themselves continuous functions of the pair features $\phi(i,j)$ on the fixed candidate set $\mathcal{C}$. Since $\operatorname{arctanh}$ is continuous on $(-1,1)$, the composite map $\phi(i,j) \mapsto u(i,j)$ is continuous on $\mathcal{C}$.

By the universal approximation theorem for MLPs with sufficient width, there exists an MLP f_θ such that

$$\big|f_\theta(\phi(i,j)) - u(i,j)\big| < \epsilon, \quad \forall (i,j) \in \mathcal{C}, \tag{41}$$

for any $\epsilon > 0$. Substituting into the residual scorer:

$$\begin{aligned} g_{\text{res}}(i,j) &= A_{\text{orig}}(i,j) + \delta \cdot \tanh\big(f_\theta(\phi(i,j))\big) && (42)\\ &\approx A_{\text{orig}}(i,j) + \delta \cdot \tanh\big(u(i,j)\big) && (43)\\ &= A_{\text{orig}}(i,j) + \delta \cdot \frac{g_{\text{sim}}(i,j) - A_{\text{orig}}(i,j)}{\delta} && (44)\\ &= g_{\text{sim}}(i,j). && (45) \end{aligned}$$

Since the approximation error can be made arbitrarily small, every function in $\mathcal{F}_{\text{sim}}$ can be realized (or approximated to arbitrary precision) by a function in $\mathcal{F}_{\text{res}}$, establishing $\mathcal{F}_{\text{sim}} \subseteq \mathcal{F}_{\text{res}}$.

Part 2: Strictness ($\mathcal{F}_{\text{sim}} \neq \mathcal{F}_{\text{res}}$). Every function in $\mathcal{F}_{\text{sim}}$ depends on a node pair (i,j) only through the two scalars $(S_T(i,j), S_V(i,j))$, combined via a global linear rule with shared coefficients α and

λ. Therefore, for any two pairs (i, j) and (u, v) satisfying

$$S_T(i,j) = S_T(u,v), \quad S_V(i,j) = S_V(u,v), \quad A_{\text{orig}}(i,j) = A_{\text{orig}}(u,v),$$

every $g_{\text{sim}} \in \mathcal{F}_{\text{sim}}$ must assign them the same score:

$$g_{\text{sim}}(i,j) = g_{\text{sim}}(u,v). \tag{46}$$

Now consider the residual scorer. Its input is the full pair feature $\phi(i, j)$ (Eq. 5), which includes concatenated raw features, element-wise Hadamard products $\mathbf{x}_i^t \odot \mathbf{x}_j^t$, $\mathbf{x}_i^v \odot \mathbf{x}_j^v$, and absolute differences $|\mathbf{x}_i^t - \mathbf{x}_j^t|$, $|\mathbf{x}_i^v - \mathbf{x}_j^v|$.

We construct a concrete counterexample. Let $d_t = 2$ and consider two node pairs:

$$\text{Pair 1:} \quad \mathbf{x}_i^t = (1,0), \quad \mathbf{x}_j^t = (0,1), \tag{47}$$

$$\text{Pair 2:} \quad \mathbf{x}_u^t = \tfrac{1}{\sqrt{2}}(1,1), \quad \mathbf{x}_v^t = \tfrac{1}{\sqrt{2}}(1,-1). \tag{48}$$

Both pairs have unit-norm vectors with $S_T = \mathbf{x}_i^t \cdot \mathbf{x}_j^t = 0$ and $S_T = \mathbf{x}_u^t \cdot \mathbf{x}_v^t = 0$. Assigning identical vision features to both pairs ensures S_V is also equal.

However, the Hadamard products differ:

$$\mathbf{x}_i^t \odot \mathbf{x}_j^t = (0,0), \tag{49}$$

$$\mathbf{x}_u^t \odot \mathbf{x}_v^t = \tfrac{1}{2}(1,-1), \tag{50}$$

so $\phi(i, j) \neq \phi(u, v)$. Since an MLP can realize non-linear, dimension-weighted functions of ϕ, there exists $g_{\text{res}} \in \mathcal{F}_{\text{res}}$ such that

$$g_{\text{res}}(i,j) \neq g_{\text{res}}(u,v), \tag{51}$$

which is impossible for any $g_{\text{sim}} \in \mathcal{F}_{\text{sim}}$ by Eq. 46.

Therefore $\mathcal{F}_{\text{sim}} \subsetneq \mathcal{F}_{\text{res}}$. □

Remark 2. *Proposition 1 characterizes* expressiveness*, not generalization. The inclusion follows from a universal-approximation argument on the residual MLP, and strictness from dimension-wise interactions (e.g., $\mathbf{x}_i^t \odot \mathbf{x}_j^t$) that cosine similarity collapses to a scalar. Whether this additional capacity translates to better task performance depends on the learning context: our ablations (Sec. 5.3) show that the residual refiner's advantage is not consistent without cross-modal propagation, and that the residual refiner gains more from coupling than the similarity refiner on Fashion and Reddit-S (Table 4).*

M Proof of Proposition 2 (Joint Optimization Contains Non-Reciprocal Alternatives)

Proposition 2 (Joint Optimization Contains Non-Reciprocal Alternatives). *Let $\mathcal{J}(\theta_{\text{ref}}, \theta_{\text{gnn}}) = \ell(f_{\theta_{\text{gnn}}}(X, A_{\text{refine}}(\theta_{\text{ref}})), Y)$ denote the end-to-end task objective. Then the joint optimization space contains any fixed or frozen refiner as a special case:*

$$\inf_{\theta_{\text{ref}},\, \theta_{\text{gnn}}} \mathcal{J}(\theta_{\text{ref}}, \theta_{\text{gnn}}) \leq \inf_{\theta_{\text{gnn}}} \mathcal{J}(\theta_{\text{ref}}^{\text{fix}}, \theta_{\text{gnn}}), \tag{52}$$

for any fixed $\theta_{\text{ref}}^{\text{fix}}$, including the identity refiner $A_{\text{refine}} = A_{\text{orig}}$ when it belongs to the refiner parameter space, and any frozen two-stage refiner.

Let Θ_{ref} and Θ_{gnn} denote the parameter spaces of the refiner and the GNN, respectively. Any fixed refiner $\theta_{\text{ref}}^{\text{fix}}$ restricts the optimization to the slice $\{\theta_{\text{ref}}^{\text{fix}}\} \times \Theta_{\text{gnn}}$, which is a subset of the full feasible set $\Theta_{\text{ref}} \times \Theta_{\text{gnn}}$. Therefore, the infimum over the full joint space cannot exceed the infimum over this restricted slice:

$$\inf_{(\theta_{\text{ref}},\, \theta_{\text{gnn}}) \in \Theta_{\text{ref}} \times \Theta_{\text{gnn}}} \mathcal{J}(\theta_{\text{ref}}, \theta_{\text{gnn}}) \leq \inf_{\theta_{\text{gnn}} \in \Theta_{\text{gnn}}} \mathcal{J}(\theta_{\text{ref}}^{\text{fix}}, \theta_{\text{gnn}}). \tag{53}$$

This includes the identity refiner $A_{\text{refine}} = A_{\text{orig}}$ when it belongs to the refiner parameter space, as well as any frozen two-stage refiner. □

N Proof of Theorem 1 (Cross-Modal Information Accessibility)

We prove both parts for the text stream; the vision stream follows by symmetry.

Part (i): Decoupled model — zero cross-modal Jacobian. A decoupled multimodal GNN updates each modality stream independently, with arbitrary per-modality weights:

$$H_T^{(\ell+1)} = \sigma\big(A\, H_T^{(\ell)}\, W_T^{(\ell)}\big), \qquad H_V^{(\ell+1)} = \sigma\big(A\, H_V^{(\ell)}\, W_V^{(\ell)}\big), \tag{54}$$

where $W_T^{(\ell)}$ and $W_V^{(\ell)}$ may differ.

We prove $\frac{\partial H_T^{(\ell)}}{\partial H_V^{(0)}} = 0$ for all $\ell \geq 0$ by induction.

Base case ($\ell = 0$). $H_T^{(0)}$ is the projected text input, which is independent of $H_V^{(0)}$ by construction.

Inductive step. Suppose $\frac{\partial H_T^{(\ell)}}{\partial H_V^{(0)}} = 0$. Since A and $W_T^{(\ell)}$ are independent of $H_V^{(0)}$, and $H_T^{(\ell+1)} = \sigma(A\, H_T^{(\ell)}\, W_T^{(\ell)})$ depends on $H_V^{(0)}$ only through $H_T^{(\ell)}$, by the chain rule:

$$\frac{\partial H_T^{(\ell+1)}}{\partial H_V^{(0)}} = \frac{\partial H_T^{(\ell+1)}}{\partial H_T^{(\ell)}} \cdot \frac{\partial H_T^{(\ell)}}{\partial H_V^{(0)}} = \frac{\partial H_T^{(\ell+1)}}{\partial H_T^{(\ell)}} \cdot 0 = 0. \tag{55}$$

Note that this holds for *any* choice of per-modality weights $W_T^{(\ell)}, W_V^{(\ell)}$, any activation σ, and any graph A. The result is a structural consequence of the decoupled update rule, not a property of specific parameters. □

Part (ii): Coupled model — non-zero cross-modal Jacobian. The coupled text-stream update is:

$$H_T^{(\ell+1)} = \sigma\Big(W^{(\ell)}\big(A\, H_T^{(\ell)} + \gamma\, H_V^{(\ell)} + \rho\, A\, H_V^{(\ell)}\big)\Big). \tag{56}$$

We show that $\frac{\partial H_T^{(\ell)}}{\partial H_V^{(0)}} \neq 0$ for all $\ell \geq 1$ when $\rho > 0$ or $\gamma > 0$.

Layer $\ell = 1$. From Eq. 56 with $\ell = 0$:

$$H_T^{(1)} = \sigma\Big(W^{(0)}\big(A\, H_T^{(0)} + \gamma\, H_V^{(0)} + \rho\, A\, H_V^{(0)}\big)\Big). \tag{57}$$

Let $Z = W^{(0)}\big(A\, H_T^{(0)} + \gamma\, H_V^{(0)} + \rho\, A\, H_V^{(0)}\big)$ denote the pre-activation. By the chain rule:

$$\frac{\partial H_T^{(1)}}{\partial H_V^{(0)}} = \mathrm{diag}\big(\sigma'(Z)\big) \cdot W^{(0)} \cdot (\gamma I + \rho\, A). \tag{58}$$

When $\gamma > 0$ or $\rho > 0$, the factor $(\gamma I + \rho\, A) \neq 0$. The diagonal matrix $\mathrm{diag}(\sigma'(Z))$ is non-zero for generic inputs (i.e., except on a measure-zero set where all pre-activations are at non-differentiable points of σ). Hence $\frac{\partial H_T^{(1)}}{\partial H_V^{(0)}} \neq 0$ generically.

Deeper layers ($\ell \geq 2$). By the chain rule and the multivariate dependency structure (where $H_T^{(\ell)}$ depends on $H_V^{(0)}$ both directly through the coupling terms and indirectly through $H_V^{(k)}$ for $k < \ell$), the Jacobian $\frac{\partial H_T^{(\ell)}}{\partial H_V^{(0)}}$ accumulates contributions from all cross-modal paths through layers $1, \ldots, \ell$. Since the layer-1 contribution 58 is generically non-zero and subsequent layers compose via non-singular weight matrices, the Jacobian remains generically non-zero at all depths. □

O Proof of Corollary 1 (Spectral Characterization)

Since A_{sup} is a 2×2 block matrix with commuting blocks (all blocks are polynomials of A_{refine}), we can diagonalize it via the block eigendecomposition.

Let $\mathbf{v}_i$ be an eigenvector of A_{refine} with eigenvalue μ_i. Consider the two $2N$-dimensional vectors:

$$\mathbf{q}_i^{+} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{v}_i \\ \mathbf{v}_i \end{pmatrix}, \qquad \mathbf{q}_i^{-} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{v}_i \\ -\mathbf{v}_i \end{pmatrix}. \tag{59}$$

Symmetric mode.

$$A_{\text{sup}}\,\mathbf{q}_i^+ = \frac{1}{\sqrt{2}}\begin{pmatrix} A_{\text{refine}}\,\mathbf{v}_i + (\gamma I + \rho\, A_{\text{refine}})\,\mathbf{v}_i \\ (\gamma I + \rho\, A_{\text{refine}})\,\mathbf{v}_i + A_{\text{refine}}\,\mathbf{v}_i \end{pmatrix}$$
$$= \frac{1}{\sqrt{2}}\begin{pmatrix} \mu_i\,\mathbf{v}_i + \gamma\,\mathbf{v}_i + \rho\mu_i\,\mathbf{v}_i \\ \gamma\,\mathbf{v}_i + \rho\mu_i\,\mathbf{v}_i + \mu_i\,\mathbf{v}_i \end{pmatrix} = \big[(1+\rho)\mu_i + \gamma\big]\,\mathbf{q}_i^+. \tag{60}$$

Anti-symmetric mode.

$$A_{\text{sup}}\,\mathbf{q}_i^- = \frac{1}{\sqrt{2}}\begin{pmatrix} A_{\text{refine}}\,\mathbf{v}_i + (\gamma I + \rho\, A_{\text{refine}})\,(-\mathbf{v}_i) \\ (\gamma I + \rho\, A_{\text{refine}})\,\mathbf{v}_i + A_{\text{refine}}\,(-\mathbf{v}_i) \end{pmatrix}$$
$$= \frac{1}{\sqrt{2}}\begin{pmatrix} \mu_i\,\mathbf{v}_i - \gamma\,\mathbf{v}_i - \rho\mu_i\,\mathbf{v}_i \\ \gamma\,\mathbf{v}_i + \rho\mu_i\,\mathbf{v}_i - \mu_i\,\mathbf{v}_i \end{pmatrix}$$
$$= \frac{1}{\sqrt{2}}\begin{pmatrix} [(1-\rho)\mu_i - \gamma]\,\mathbf{v}_i \\ -[(1-\rho)\mu_i - \gamma]\,\mathbf{v}_i \end{pmatrix} = \big[(1-\rho)\mu_i - \gamma\big]\,\mathbf{q}_i^-. \tag{61}$$

Since $\{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ form a complete eigenbasis for A_{refine}, the $2N$ vectors $\{\mathbf{q}_1^+, \ldots, \mathbf{q}_N^+, \mathbf{q}_1^-, \ldots, \mathbf{q}_N^-\}$ form a complete eigenbasis for A_{sup}, with eigenvalues $\{(1+\rho)\mu_i + \gamma\}_{i=1}^N \cup \{(1-\rho)\mu_i - \gamma\}_{i=1}^N$. □

Interpretation. The spectral decomposition reveals how ρ and γ jointly shape the coupled propagation:

- *Symmetric amplification.* Cross-modal coupling amplifies graph spectral components in the symmetric subspace by a factor $(1+\rho)$ relative to the decoupled case ($\rho = 0$). The self-injection γ provides an additional frequency-independent boost to symmetric modes. For **homophilic** graphs, where low-frequency components (large μ_i) carry the dominant class signal, this amplification strengthens the informative spectral components and accelerates convergence to useful representations.
- *Anti-symmetric attenuation.* Anti-symmetric modes are attenuated by factor $(1-\rho)$ and shifted by $-\gamma$. When ρ and γ are large, the model aggressively suppresses cross-modal disagreement, which is beneficial when the two modalities carry consistent semantic signals.
- *Adaptivity to graph heterophily.* For **heterophilic** graphs where high-frequency components (small or negative μ_i) carry discriminative signal, the model can learn small ρ and γ, preserving high-frequency components in both symmetric and anti-symmetric subspaces. Since $\rho = \sigma(\rho_{\text{logit}})$ and $\gamma = \sigma(\gamma_{\text{logit}})$ are learnable, the spectral filter adapts to the homophily/heterophily characteristics of the data without manual tuning.

P Proof of Theorem 2 (Shared-Weight Cross-Modal Consistency)

The coupled update rules are:

$$H_T^{(\ell+1)} = \sigma\Big(W^{(\ell)}\big(A_{\text{refine}}\,H_T^{(\ell)} + \gamma\,H_V^{(\ell)} + \rho\,A_{\text{refine}}\,H_V^{(\ell)}\big)\Big), \tag{62}$$
$$H_V^{(\ell+1)} = \sigma\Big(W^{(\ell)}\big(A_{\text{refine}}\,H_V^{(\ell)} + \gamma\,H_T^{(\ell)} + \rho\,A_{\text{refine}}\,H_T^{(\ell)}\big)\Big). \tag{63}$$

Step 1: Express the pre-activation difference. Define the pre-activation inputs:

$$u_T = A_{\text{refine}}\,H_T^{(\ell)} + \gamma\,H_V^{(\ell)} + \rho\,A_{\text{refine}}\,H_V^{(\ell)}, \tag{64}$$
$$u_V = A_{\text{refine}}\,H_V^{(\ell)} + \gamma\,H_T^{(\ell)} + \rho\,A_{\text{refine}}\,H_T^{(\ell)}. \tag{65}$$

Their difference is:

$$u_T - u_V = A_{\text{refine}}\big(H_T^{(\ell)} - H_V^{(\ell)}\big) + \gamma\big(H_V^{(\ell)} - H_T^{(\ell)}\big) + \rho\,A_{\text{refine}}\big(H_V^{(\ell)} - H_T^{(\ell)}\big)$$
$$= A_{\text{refine}}\,\Delta^{(\ell)} - \gamma\,\Delta^{(\ell)} - \rho\,A_{\text{refine}}\,\Delta^{(\ell)}$$
$$= (1-\rho)\,A_{\text{refine}}\,\Delta^{(\ell)} - \gamma\,\Delta^{(\ell)}. \tag{66}$$

Step 2: Apply the Lipschitz bound. Taking the difference of Eqs. 62–63:

$$\Delta^{(\ell+1)} = H_T^{(\ell+1)} - H_V^{(\ell+1)} = \sigma\big(W^{(\ell)}u_T\big) - \sigma\big(W^{(\ell)}u_V\big). \tag{67}$$

Since $\sigma(\cdot)$ is 1-Lipschitz (e.g., ReLU, sigmoid):

$$\left\|\Delta^{(\ell+1)}\right\| \le \left\|W^{(\ell)}(u_T - u_V)\right\| \le \left\|W^{(\ell)}\right\| \cdot \|u_T - u_V\|. \tag{68}$$

Step 3: Bound the pre-activation difference. From Eq. 66 and the triangle inequality:

$$\begin{aligned}\|u_T - u_V\| &\le \|(1-\rho)\, A_{\text{refine}}\, \Delta^{(\ell)}\| + \|\gamma\, \Delta^{(\ell)}\| \\ &= (1-\rho)\, \|A_{\text{refine}}\|\, \|\Delta^{(\ell)}\| + \gamma\, \|\Delta^{(\ell)}\|,\end{aligned} \tag{69}$$

where $|1-\rho| = 1-\rho$ since $\rho = \sigma(\rho_{\text{logit}}) \in (0,1)$.

Step 4: Combine and telescope. Substituting Eq. 69 into Eq. 68:

$$\left\|\Delta^{(\ell+1)}\right\| \le \left\|W^{(\ell)}\right\| \Big((1-\rho)\, \|A_{\text{refine}}\| + \gamma \Big) \left\|\Delta^{(\ell)}\right\|. \tag{70}$$

Telescoping over $\ell = 0, 1, \ldots, L-1$:

$$\left\|\Delta^{(L)}\right\| \le \prod_{\ell=0}^{L-1} \left\|W^{(\ell)}\right\| \Big((1-\rho)\, \|A_{\text{refine}}\| + \gamma \Big) \cdot \left\|\Delta^{(0)}\right\|. \tag{71}$$

□

Q Dataset Statistics

Table 12: Dataset statistics. Datasets marked with † are from the MAGB benchmark (Yan et al., 2025); those marked with ‡ are from MM-Graph (Zhu et al., 2025). NC = node classification, LP = link prediction.

Dataset	#Nodes	#Edges	#Classes	Task
Ele-fashion‡	97,766	199,602	12	NC
Goodreads-NC‡	685,294	7,235,084	10	NC
Movies†	16,672	218,390	20	NC
Toys†	20,695	126,886	18	NC
Grocery†	17,074	171,340	20	NC
Reddit-S†	15,894	566,160	20	NC
Reddit-M†	99,638	1,167,188	50	NC
Amazon-Sports‡	50,250	356,202	—	LP
Amazon-Cloth‡	125,839	951,271	—	LP

R Hyperparameters

We provide the major hyperparameter search space:

- K: $\{1,\ 3,\ 5,\ 8,\ 12\}$.
- Learning rate : $\{0.005,\ 0.003,\ 0.001\}$.
- Dropout: $\{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.8\}$.
- GNN layers: $\{1,\ 2,\ 3\}$.
- GNN hidden dimension of tokens: $\{256,\ 512,\ 1024\}$.
- Batch Size: $\{256,\ 512,\ 1024\}$.
- Weight Decay: $\{0,\ 1e-05,\ 5e-05,\ 1e-04\}$.

S Limitation

ReCoG depends on the quality of pretrained multimodal features, and noisy or weakly aligned inputs can degrade topology learning, especially for the similarity refiner. In addition, the current design focuses on two modalities and may require extension for more heterogeneous settings.