DYNAMIC MULTIMODAL GRAPH LEARNING WITH CONTENT-CONDITIONED EDGE REWEIGHTING AND HIERARCHICAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal graph learning combines relational structure with textual and visual attributes, but the utility of a neighbor depends on the target content. We introduce DMGraph, which learns content-conditioned weights over induced multimodal subgraphs and aligns text and vision at node, subgraph, and batch-context levels. Its dynamic edge scorer learns rank-controlled bilinear relevance functions, while hardness-aware contrastive learning handles semantically similar negatives. Across WikiWeb2M, FB15K237, Ele-Fashion, and Goodreads-LP, DMGraph consistently improves over graph-multimodal baselines. In expanded neighborhoods, intermediate learned sparsity consistently outperforms dense reweighting and hard Top-K. Under 50% edge corruption, DMGraph limits relative performance degradation to about 15% on both evaluated datasets, demonstrating effective selection of task-relevant graph context.

1 INTRODUCTION

Multimodal graph learning connects relational structure with textual, visual, and other modality-specific attributes. Nodes may represent webpages, products, entities, or documents, while edges encode semantic, spatial, temporal, or behavioral relations (Ektefaie et al., 2023). Applications include scientific knowledge discovery (Kim et al., 2023), recommendation (Xu et al., 2025b), and document understanding (Ning et al., 2025). The value of this context depends on the target instance: a graph neighbor may be structurally close yet contribute little to the target's text or image. Effective fusion therefore requires deciding both which neighbors to emphasize and how to align the information they carry.

These decisions are coupled. Static graph structures provide candidate relations, while content determines their relevance to a particular target (Franceschi et al., 2019; Chen et al., 2020). Reweighting those relations changes the context presented to each modality. Text and vision representations, in turn, have different geometries, and direct fusion can favor one modality when their feature distributions are misaligned (Liang et al., 2022; Wang et al., 2020; Peng et al., 2022). The alignment problem also extends beyond individual text-image pairs. Local neighborhoods, subgraph communities, and global patterns organize graph semantics at different scales (Ying et al., 2021). Matching two node embeddings leaves their surrounding subgraphs unconstrained, even though those subgraphs supply the context for prediction.

We propose **DMGraph** (**D**ynamic **M**ultimodal **Graph** Learning with content-conditioned edge reweighting and hierarchical alignment) to learn neighborhood relevance and cross-modal consistency jointly. For each target, DMGraph scores candidate edges within induced text and visual subgraphs and uses the learned weights in multi-hop diffusion (Gasteiger et al., 2019). Modality-specific projections then map the resulting representations into a shared space, where debiased contrastive learning accounts for potentially false negatives (Chuang et al., 2020). Alignment losses act on paired nodes, subgraph summaries, and batch-global representations. The foundation model thus receives context that has been adapted to node content and aligned at several granularities.

The components address different parts of this computation. Edge weighting controls which messages enter a representation; modality-specific projection controls how text and vision are compared; hierarchical alignment connects local correspondence to broader graph context. Their joint training

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.

lets the task loss and alignment objectives shape the same relational representations. This design makes neighborhood adaptation useful to cross-modal fusion while retaining the input graph as a structural prior.

Our analysis characterizes bilinear edge-function expressiveness on the candidate set and separates the population InfoNCE reference from the implemented hard-negative and batch-context objectives. Experiments on generative and discriminative tasks evaluate the resulting representations under matched parameter-efficient tuning settings. Our contributions are:

- A multimodal graph framework that jointly learns content-dependent neighborhood weights and cross-modal representations before foundation-model prediction.
- A dynamic adjacency mechanism with an exact bilinear realization result and a rank-controlled approximation bound on the candidate set.
- Hierarchical contrastive alignment with modality-specific projections, hardness-aware negative weighting, and batch-context regularization.
- Improvements on four multimodal graph benchmarks, together with pruning-regime and edge-corruption experiments that isolate the value of adaptive structure.

2 Related Work

2.1 Multimodal Graph Learning

Multimodal graph learning has emerged as a paradigm for jointly modeling heterogeneous data sources with relational structure (Ektefaie et al., 2023). Early approaches extend graph neural networks to handle multiple feature types through concatenation (Wei et al., 2019) or cross-modal attention (Yang et al., 2021), but treat modalities symmetrically and therefore obscure their distributional differences. GraphMFT (Li et al., 2023a) separates intra-modal and inter-modal attention but still operates on fixed topologies. COHESION (Xu et al., 2025a) introduces dual-stage fusion for composite graphs, yet its fusion strategy remains uniform across node pairs. Graph4MM (Ning et al., 2025) advances the field by introducing hop-aware visual querying for LLM integration, but keeps the input adjacency as the structural carrier and aligns modalities primarily at the node level. Recent concurrent work has begun to adapt multimodal graph structure more explicitly. DiP builds dynamic information pathways through modality-specific pseudo nodes (Hong et al., 2026), while TMTE co-evolves task-aware topology and modality representations (Zhu et al., 2026). DMGraph develops the LLM-integrated graph-multimodal approach by learning edge relevance within induced subgraphs and coordinating cross-modal alignment at node, subgraph, and batch-global levels.

2.2 Graph Structure Learning and Edge Reweighting

Learning graph structure from data has evolved through several paradigms. Bilevel optimization (Franceschi et al., 2019) and iterative anchor-based refinement (Chen et al., 2020) pioneered the area but scale poorly. NodeFormer (Wu et al., 2022) achieves $O(N)$ all-pair message passing via kernelized Gumbel-Softmax, while DIFFormer (Wu et al., 2023) derives structure learning from energy-constrained diffusion. PR-MPNNs (Qian et al., 2024) use differentiable k-subset sampling for rewiring, and GSR decouples structure estimation from downstream training through self-supervised refinement (Zhao et al., 2023). Recent benchmarks (Zhou et al., 2023) show that graph structure learning does not consistently outperform vanilla GNNs, suggesting that structure adaptation alone is insufficient without task and modality context. The integration of language models with graphs (Chen et al., 2024; He et al., 2024) demonstrates cross-domain structure understanding, and Graph-MLLM further shows that fine-tuned MLLM-as-predictor baselines can be strong even without explicit graph modules (Liu et al., 2025). DMGraph learns adjacency weights locally within induced multimodal subgraphs. This connects graph structure learning to the content and alignment objectives used by the downstream foundation model.

2.3 Cross-Modal Alignment

Bridging the representational gap between modalities is central to multimodal learning. Contrastive learning methods align paired samples across modalities (Liang et al., 2022), but standard methods

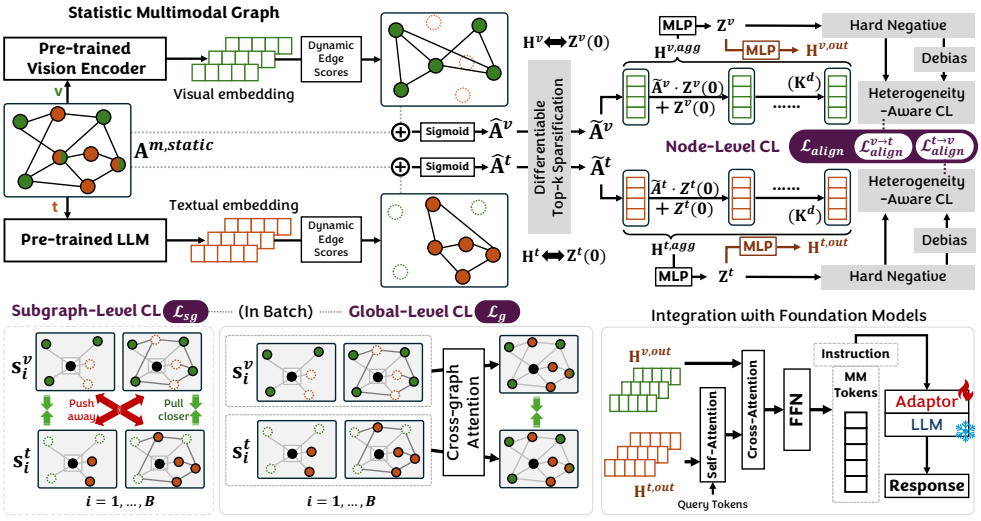

Figure 1: Our DMGraph comprises three modules: Content-Conditioned Edge Reweighting, Heterogeneity-Aware Cross-Modal Alignment, and Hierarchical Multi-Granularity Alignment.

treat all negatives uniformly, ignoring false negatives in graph-structured data where neighboring nodes often share semantics. Debiased contrastive learning (Chuang et al., 2020) adjusts the negative term for false-negative contamination. In DMGraph, visual proximity supplies a heuristic signal for negative weighting, while the debiasing term adjusts the contrastive objective. Multi-scale alignment has been studied in hierarchical graph learning (Ying et al., 2021), yet existing methods usually optimize a single granularity or do not coordinate node, subgraph, and global consistency. DMGraph integrates modality-specific projections (Park et al., 2024) with hierarchical contrastive objectives and reports ablations that isolate these design choices.

3 Method

Figure 1 gives the overall pipeline. DMGraph first induces text and visual subgraphs around a target node, then computes content-conditioned edge scores within each induced subgraph. The resulting representations are aligned across modalities at node, subgraph, and batch-global levels before being passed to the foundation-model interface.

3.1 Problem Setup

We consider a multimodal attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}^t, \mathcal{X}^v)$, where nodes are connected by graph edges and may carry textual and visual attributes. For each target node, we induce modality-specific text and visual subgraphs by selecting neighbors whose corresponding attributes are available. DMGraph is evaluated on both generative tasks, where the model produces a textual response conditioned on the target and its induced subgraphs, and discriminative tasks, where the response is mapped to a class label or link-prediction decision.

The central object learned by DMGraph is a content-conditioned adjacency $\tilde{\mathbf{A}}(\mathbf{H}; \Theta)$ over each induced candidate subgraph. Full notation, subgraph construction, and task objectives are given in Appendix A and Appendix A.1.

3.2 Content-Conditioned Edge Reweighting

The dynamic adjacency assigns content-dependent weights to candidate neighbors. Dot-product scores combine node representations with a static graph prior, and a differentiable gate controls their contribution to diffusion.

3.2.1 ATTENTION-BASED DYNAMIC EDGE SCORING

For each modality $m \in \{t, v\}$, let $\mathbf{H}^m = [\mathbf{h}_1^m, \ldots, \mathbf{h}_{n_m}^m]^\top \in \mathbb{R}^{n_m \times d}$ denote the node embedding matrix, where $n_m = |\mathcal{V}^m|$. We compute dynamic edge scores via scaled dot-product attention:

$$s_{ij}^m = (\mathbf{W}_q^m \mathbf{h}_i^m)^\top (\mathbf{W}_k^m \mathbf{h}_j^m)/\sqrt{d}, \tag{1}$$

where $\mathbf{W}_q^m, \mathbf{W}_k^m \in \mathbb{R}^{d \times d}$ are learnable query and key projections. To incorporate structural priors from the static adjacency $\mathbf{A}^{m,\text{static}}$, we define the combined score:

$$\hat{s}_{ij}^m = s_{ij}^m + \beta^m \cdot \mathbf{A}_{ij}^{m,\text{static}}, \tag{2}$$

where $\beta^m > 0$ is a learnable scalar balancing learned dynamics and static priors. Let $\mathcal{C}^m$ denote the valid candidate pairs in the induced subgraph (existing edges plus self-loops). For $(i, j) \in \mathcal{C}^m$, the dynamic adjacency is $\hat{\mathbf{A}}_{ij}^m = \sigma(\hat{s}_{ij}^m)$, where $\sigma(\cdot)$ is the sigmoid function; entries outside $\mathcal{C}^m$ are fixed to zero and are not passed to the gate.

3.2.2 DIFFERENTIABLE TOP-K SPARSIFICATION

Direct top-K_s selection is non-differentiable, precluding end-to-end gradient-based optimization. We adopt a relaxed formulation inspired by (Xie et al., 2020; Toenshoff et al., 2021). For each node i, we define a soft gate: $g_{ij}^m = \sigma(\frac{\hat{\mathbf{A}}_{ij}^m - \mu_i^m}{\epsilon})$, where μ_i^m is a row-wise threshold defined below and $\epsilon > 0$ is a temperature parameter. The sparsified adjacency is $\tilde{\mathbf{A}}_{ij}^m = \hat{\mathbf{A}}_{ij}^m \cdot g_{ij}^m$.

Row-wise threshold. Let r_i denote the number of valid candidate entries in row i of $\hat{\mathbf{A}}^m$ (existing edges plus self-loop), and let $\hat{a}_{i,(1)}^m \geq \cdots \geq \hat{a}_{i,(r_i)}^m$ be its sorted entries. We define

$$\mu_i^m \;=\; \hat{a}^m_{i,\,(\min(K_s,\,r_i))} \;-\; \eta_{K_s,r_i}, \quad \eta_{K_s,r_i} = \begin{cases} 0, & K_s \leq r_i, \\ \eta_0, & K_s > r_i, \end{cases} \tag{3}$$

where $\eta_0 = 10^{-3}$ is a small fixed margin. This yields two regimes. *Pruning regime* ($K_s \leq r_i$): μ_i^m is the K_s-th largest entry, so g_{ij}^m converges to the standard top-K mask as $\epsilon \to 0$. *Reweighting regime* ($K_s > r_i$): μ_i^m lies η_0 below the smallest valid entry, so $g_{ij}^m \in (\frac{1}{2}, 1)$ for every valid (i, j) and is monotonically increasing in attention rank. All valid entries remain active, with the spread of soft gate values controlled by ϵ.

Annealing. We initialize $\epsilon_0 = 1.0$ and anneal $\epsilon^{(t)} = \max(\epsilon_{\min}, \epsilon_0 \cdot \gamma^t)$ with $\gamma = 0.98$ and $\epsilon_{\min} = 0.01$. We employ the straight-through estimator (STE) (Bengio et al., 2013): forward pass uses the hard mask $\mathbf{1}[\hat{\mathbf{A}}_{ij}^m \geq \mu_i^m]$ when $K_s \leq r_i$ and the all-ones mask when $K_s > r_i$; backward pass routes gradients through g_{ij}^m in both regimes.

3.2.3 SYMMETRIC NORMALIZATION AND DYNAMIC MULTI-HOP DIFFUSION

To ensure proper message passing dynamics, we symmetrize and normalize the sparsified adjacency:

$$\tilde{\mathbf{A}}^{m,\text{norm}} = \tilde{\mathbf{D}}^{-1/2}(\bar{\mathbf{A}}^m + \mathbf{I})\tilde{\mathbf{D}}^{-1/2}, \tag{4}$$

where $\bar{\mathbf{A}}^m = \frac{1}{2}(\tilde{\mathbf{A}}^m + (\tilde{\mathbf{A}}^m)^\top)$ is symmetrized and $\tilde{\mathbf{D}}_{ii} = \sum_j (\bar{\mathbf{A}}_{ij}^m + \delta_{ij})$. We incorporate learned dynamic structure into a multi-hop diffusion process inspired by APPNP (Gasteiger et al., 2019):

$$\mathbf{Z}^{m,(k+1)} = (1 - \alpha)\tilde{\mathbf{A}}^{m,\text{norm}}\mathbf{Z}^{m,(k)} + \alpha\mathbf{Z}^{m,(0)}, \tag{5}$$

where $\mathbf{Z}^{m,(0)} = \mathbf{H}^m$ and $\alpha \in (0, 1)$ is the teleport probability ensuring the initial features are preserved. After K_d diffusion steps, the final modality-specific representations are obtained via a learnable transformation:

$$\mathbf{H}^{m,\text{out}} = \mathbf{Z}^{m,(K_d)}\mathbf{W}_o^m + \mathbf{b}_o^m. \tag{6}$$

Edge scoring and diffusion operate within each induced subgraph. Appendix B.3 gives their dependence on subgraph size, feature dimension, and diffusion depth.

3.3 HETEROGENEITY-AWARE CROSS-MODAL ALIGNMENT

Text and vision modalities exhibit fundamentally different representational properties: text is discrete and symbolic, while vision is continuous and perceptual. We propose a heterogeneity-aware alignment mechanism with modality-specific projections and debiased hard negative mining.

3.3.1 MODALITY-SPECIFIC PROJECTIONS

We extract representations from multiple diffusion levels. Let $\{\mathbf{Z}^{m,(k)}\}_{k=0}^{K_d}$ denote the intermediate representations from Eq. (5). We aggregate them via learnable level-wise attention:

$$\mathbf{h}_i^{m,\text{agg}} = \sum_{k=0}^{K_d} \omega_{i,k}^m \mathbf{z}_i^{m,(k)}, \tag{7}$$

where $\omega_{i,k}^m = \frac{\exp(\mathbf{w}^m \cdot \mathbf{z}_i^{m,(k)})}{\sum_{k'=0}^{K_d} \exp(\mathbf{w}^m \cdot \mathbf{z}_i^{m,(k')})}$ and $\mathbf{w}^m \in \mathbb{R}^d$ is a learnable attention vector. Each modality is then projected into a shared semantic space via modality-specific MLPs:

$$\mathbf{z}_i^t = \mathbf{W}_2^t \cdot \text{GELU}(\mathbf{W}_1^t \mathbf{h}_i^{t,\text{agg}} + \mathbf{b}_1^t) + \mathbf{b}_2^t, \quad \mathbf{z}_j^v = \mathbf{W}_2^v \cdot \text{GELU}(\mathbf{W}_1^v \mathbf{h}_j^{v,\text{agg}} + \mathbf{b}_1^v) + \mathbf{b}_2^v, \tag{8}$$

where $\mathbf{W}_1^m \in \mathbb{R}^{d_h \times d}$, $\mathbf{W}_2^m \in \mathbb{R}^{d \times d_h}$, and $d_h = 4d$.

3.3.2 DEBIASED HARD NEGATIVE CONTRASTIVE LEARNING

We define positive pairs based on cross-modal correspondence within the same node: $\mathcal{S}^+ = \{(i,j) \mid v_i \in \mathcal{V}^t, v_j \in \mathcal{V}^v, i = j, \mathbf{x}_i^t \neq \varnothing, \mathbf{x}_j^v \neq \varnothing\}$. For a text anchor $\mathbf{z}_i^t$, we identify hard negatives as visual samples semantically similar to the anchor's corresponding visual representation:

$$w_{ik} = \begin{cases} \exp\left(-\rho \|\mathbf{z}_i^v - \mathbf{z}_k^v\|_2^2\right) & \text{if } k \neq i, \\ 0 & \text{if } k = i, \end{cases} \tag{9}$$

with normalized weights $\tilde{w}_{ik} = w_{ik} / \sum_{k' \neq i} w_{ik'}$.

False-negative adjustment. Following (Chuang et al., 2020), we introduce a contamination adjustment for semantically matched samples in the negative pool. Let $\tau^+ = \Pr(Y_k = Y_i)$ denote the semantic collision prior: the probability that a sampled negative belongs to the anchor's latent semantic class. For an anchor i, let $N_i = |\mathcal{B}| - 1$ be the number of in-batch negatives. We define the hardness-weighted negative statistic and the paired-positive statistic as

$$\widehat{N}^{\text{hard}}(\mathbf{z}_i^t) = \sum_{k \neq i} \tilde{w}_{ik}\, e^{\text{sim}(\mathbf{z}_i^t, \mathbf{z}_k^v)/\tau}, \quad \widehat{P}^+(\mathbf{z}_i^t) = e^{\text{sim}(\mathbf{z}_i^t, \mathbf{z}_i^v)/\tau}, \tag{10}$$

where the weights in $\widehat{N}^{\text{hard}}$ sum to one over the N_i negatives. The adjusted statistic subtracts the estimated positive-class contribution, rescales by $1 - \tau^+$, and clips at a numerical floor:

$$\tilde{g}_i \;=\; \max\left\{ \frac{\widehat{N}^{\text{hard}}(\mathbf{z}_i^t) - \tau^+ \widehat{P}^+(\mathbf{z}_i^t)}{1 - \tau^+},\; e^{-1/\tau} \right\}. \tag{11}$$

For labeled tasks, we estimate τ^+ from training-label collision frequencies; otherwise, we select it on the validation split. The numerical floor follows Chuang et al. (Chuang et al., 2020), while the hard-negative weighting in Eq. (9) is specific to our graph setting. Weighting, estimation, and clipping produce an anchor-dependent training statistic $\tilde{g}_i$. The symmetric alignment loss is

$$\mathcal{L}_{\text{align}} = \tfrac{1}{2}(\mathcal{L}_{\text{align}}^{t \to v} + \mathcal{L}_{\text{align}}^{v \to t}), \tag{12}$$

where $\mathcal{L}_{\text{align}}^{t \to v} = -\frac{1}{|\mathcal{S}^+|} \sum_{(i,j) \in \mathcal{S}^+} \log \frac{\exp(\text{sim}(\mathbf{z}_i^t, \mathbf{z}_j^v)/\tau)}{\exp(\text{sim}(\mathbf{z}_i^t, \mathbf{z}_j^v)/\tau) + N_i \tilde{g}_i}$, and $\mathcal{L}_{\text{align}}^{v \to t}$ is defined analogously by swapping the modalities. The factor N_i places the normalized statistic on the total-negative scale used in the contrastive denominator.

Hardness Annealing. To prevent early training instability from overly hard negatives, we linearly increase ρ from $\rho_{\min}$ to $\rho_{\max}$: $\rho^{(t)} = \rho_{\min} + (\rho_{\max} - \rho_{\min}) \cdot \min(1, t/T_{\text{wp}})$, where T_{wp} is the warm-up period.

3.4 HIERARCHICAL MULTI-GRANULARITY ALIGNMENT

Paired nodes, local subgraphs, and batch-global summaries provide distinct views of multimodal context. We align each view with a contrastive objective.

Node-level alignment. The node-level term is $\mathcal{L}_{\text{align}}$ in Eq. (12). It aligns paired text and visual node representations after modality-specific projection.

Subgraph-level alignment. To align local graph semantics, we employ a structure-aware readout function that weighs nodes by their learned importance:

$$\mathbf{s}_i^m = \sum_{j \in \mathcal{V}_i^m} \alpha_j^m \, \mathbf{z}_j^m, \quad \alpha_j^m = \frac{\exp(\text{MLP}_{\text{gate}}^m(\mathbf{z}_j^m))}{\sum_{j' \in \mathcal{V}_i^m} \exp(\text{MLP}_{\text{gate}}^m(\mathbf{z}_{j'}^m))}, \tag{13}$$

where $\text{MLP}_{\text{gate}}^m : \mathbb{R}^d \to \mathbb{R}$ is a learnable gating network. Given a batch of B multimodal graphs, the subgraph-level contrastive loss is:

$$\mathcal{L}_{\text{sg}} = -\frac{1}{B} \sum_{b=1}^{B} \log \frac{\exp(\text{sim}(\mathbf{s}_b^t, \mathbf{s}_b^v)/\tau_s)}{\sum_{b'=1}^{B} \exp(\text{sim}(\mathbf{s}_b^t, \mathbf{s}_{b'}^v)/\tau_s)}. \tag{14}$$

Batch-global alignment. We construct a batch-global context by aggregating the subgraph summaries in the current minibatch through cross-graph attention:

$$\mathbf{c}^m = \sum_{b=1}^{B} \gamma_b^m \mathbf{s}_b^m, \quad \gamma_b^m = \frac{\exp(\mathbf{w}_g^\top \mathbf{s}_b^m)}{\sum_{b'=1}^{B} \exp(\mathbf{w}_g^\top \mathbf{s}_{b'}^m)}, \tag{15}$$

where $\mathbf{w}_g \in \mathbb{R}^d$ is a learnable attention vector. For each graph b, we compute its global representation via a gating mechanism: $\mathbf{g}_b^m = \lambda_b^m \odot \mathbf{s}_b^m + (1 - \lambda_b^m) \odot \mathbf{c}^m$, where $\lambda_b^m = \sigma\left(\mathbf{W}_\lambda [\mathbf{s}_b^m; \mathbf{c}^m] + \mathbf{b}_\lambda\right)$. The batch-global contrastive loss is:

$$\mathcal{L}_g = -\frac{1}{B} \sum_{b=1}^{B} \log \frac{\exp(\text{sim}(\mathbf{g}_b^t, \mathbf{g}_b^v)/\tau_g)}{\sum_{b'=1}^{B} \exp(\text{sim}(\mathbf{g}_b^t, \mathbf{g}_{b'}^v)/\tau_g)}. \tag{16}$$

Each $\mathbf{g}_b^m$ depends on every subgraph in the current minibatch through $\mathbf{c}^m$. Accordingly, $\mathcal{L}_g$ serves as a batch-context alignment regularizer whose contribution is measured by ablation.

Auxiliary regularization. To preserve essential connectivity while allowing task-adaptive modifications, we use $\mathcal{L}_{\text{struct}} = \frac{1}{2} \sum_{m \in \{t,v\}} \|\tilde{\mathbf{A}}^m - \mathbf{A}^{m,\text{static}}\|_F^2$. This penalty discourages large departures from the static adjacency. With $\lambda_2 = 0.1$, it anchors the learned topology to the static prior while still permitting edge reweighting driven by $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{align}}$.

3.5 INTEGRATION WITH FOUNDATION MODELS

Following previous work (Ning et al., 2025), we employ a visual querying transformer to extract compact visual tokens. The enhanced node embeddings $\mathbf{H}^{t,\text{out}}, \mathbf{H}^{v,\text{out}}$ from dynamic diffusion serve as inputs. We initialize learnable query tokens $\mathbf{Q} \in \mathbb{R}^{N_q \times d}$, concatenate with textual embeddings, and apply L layers of self-attention and cross-attention:

$$\bar{\mathbf{Q}}^{(\ell)} = \text{SelfAttn}(\mathbf{Q}^{(\ell-1)}), \quad \mathbf{Q}^{(\ell)} = \text{FFN}(\text{CrossAttn}(\bar{\mathbf{Q}}^{(\ell)}, \mathbf{H}^{v,\text{out}}, \mathbf{H}^{v,\text{out}})). \tag{17}$$

The final multimodal tokens $\mathbf{Q}^{(L)}$ are projected to the PLM's embedding space:

$$\mathbf{X}_{\text{input}} = [\text{Embed}(\mathcal{I}); \text{Embed}(\mathcal{X}^t); \text{Proj}(\mathbf{Q}^{(L)})]. \tag{18}$$

3.6 TRAINING OBJECTIVE

We combine task-specific and auxiliary losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_1(\mathcal{L}_{\text{align}} + \mathcal{L}_{\text{sg}} + \mathcal{L}_{\text{g}}) + \lambda_2 \mathcal{L}_{\text{struct}}, \tag{19}$$

where $\mathcal{L}_{\text{task}}$ is the cross-entropy loss for generation or classification. We adopt a two-phase training strategy: (1) *Warm-up Phase* (first 20% of training): train with $\mathcal{L}_{\text{task}}$ only to stabilize PLM fine-tuning; (2) *Joint Phase* (remaining 80%): train with $\mathcal{L}_{\text{total}}$ to enable end-to-end multimodal learning.

Table 1: Representative main results on both generative and discriminative benchmark. WikiWeb2M evaluates long-context multimodal generation, while Goodreads-LP avoids the saturation observed in Ele-Fashion categorical metrics. Best and second-best results are **bolded** and underlined.

	Backbone	Variant	WikiWeb2M			Goodreads-LP			
			BLEU-4	ROUGE-L	CIDEr	R-L	Acc (%)	Rec (%)	Pre (%)
VLMs	BLIP	Node's Text	0.0000	0.0496	0.0060	0.1238	18.74	8.13	14.05
		Subgraph's Text	0.0000	0.0530	0.0063	0.1672	27.45	12.31	27.86
	QwenVL	Node's Text	0.0000	0.1223	0.0069	0.0689	9.42	3.78	9.67
		Subgraph's Text	0.0000	0.1192	0.0084	0.1071	10.81	4.97	12.43
OPT-125M	PLM	Node's Text	0.0078±0.0001	0.1871±0.0024	0.0783±0.0008	0.2089±0.0027	60.32±0.80	35.71±0.48	43.24±0.57
		Subgraph's Text	0.0164±0.0002	0.2198±0.0031	0.1551±0.0024	0.2541±0.0036	59.84±0.91	29.38±0.42	35.71±0.51
	MMGL	Node's Text	0.0642±0.0008	0.3807±0.0055	0.6241±0.0081	0.6314±0.0087	65.42±0.88	60.81±0.82	62.18±0.93
		Subgraph's Text	0.0770±0.0011	0.3992±0.0051	0.7606±0.0120	0.7038±0.0099	92.45±1.42	65.27±0.91	66.04±0.99
		Node's Text & Image	0.0643±0.0009	0.3825±0.0060	0.6371±0.0096	0.6428±0.0084	67.18±0.98	62.41±1.00	76.85±1.18
		Subgraph's Text & Image	0.0778±0.0010	0.4041±0.0069	0.7712±0.0116	0.7042±0.0107	92.61±1.29	75.83±1.09	76.92±1.13
		Subgraph's T & I + GNN	0.0633±0.0010	0.3814±0.0058	0.6326±0.0089	0.5193±0.0072	63.74±0.83	50.27±0.76	56.84±0.81
	Graph4MM	Hop-Aware MM-QFormer	0.0801±0.0012	0.4063±0.0055	0.7736±0.0124	0.7081±0.0096	95.83±1.31	92.74±1.25	92.81±1.34
		Hop-Diffused MM-QFormer	0.0800±0.0010	0.4076±0.0067	0.7831±0.0120	0.7139±0.0109	96.27±1.46	93.42±1.31	93.51±1.40
	TMTE	Task-aware Co-evolution	0.0754±0.0011	0.3969±0.0059	0.7580±0.0111	0.7072±0.0102	92.17±1.24	84.12±1.17	85.29±1.25
	DMGraph		**0.0834**±0.0013	**0.4119**±0.0064	**0.7994**±0.0127	**0.7389**±0.0115	**97.84**±1.51	**95.71**±1.38	**95.83**±1.44
LLaMA-1B	PLM	Node's Text	0.0757±0.0010	0.3691±0.0052	0.7185±0.0101	0.6347±0.0088	87.52±1.19	71.84±1.04	76.93±1.09
		Subgraph's Text	0.0960±0.0015	0.3949±0.0057	0.9179±0.0142	0.6582±0.0094	92.36±1.35	74.93±1.05	75.18±1.12
	MMGL	Node's Text	0.0947±0.0013	0.4375±0.0061	0.9046±0.0136	0.6358±0.0089	87.61±1.23	71.92±1.02	77.04±1.15
		Subgraph's Text	0.1087±0.0016	0.4426±0.0073	1.0426±0.0161	0.6627±0.0098	92.41±1.31	75.06±1.13	75.27±1.08
		Node's Text & Image	0.0981±0.0014	0.4449±0.0067	0.9367±0.0145	0.6018±0.0082	89.74±1.17	77.85±1.16	81.42±1.22
		Subgraph's Text & Image	0.1157±0.0018	0.4685±0.0075	1.1072±0.0176	0.6691±0.0103	90.83±1.25	78.04±1.11	79.13±1.19
		Subgraph's T & I + GNN	0.1003±0.0015	0.4487±0.0065	0.9631±0.0139	0.3438±0.0048	79.25±1.08	57.18±0.81	56.31±0.85
	Graph4MM	Hop-Aware MM-QFormer	0.1186±0.0017	0.4731±0.0072	1.1262±0.0179	0.7729±0.0119	98.42±1.50	91.36±1.27	91.45±1.32
		Hop-Diffused MM-QFormer	0.1177±0.0016	0.4713±0.0068	1.1221±0.0163	0.7702±0.0112	98.27±1.37	91.74±1.34	91.82±1.29
	TMTE	Task-aware Co-evolution	0.1141±0.0018	0.4633±0.0070	1.0825±0.0168	0.7626±0.0108	95.73±1.36	84.90±1.18	86.01±1.24
	DMGraph		**0.1234**±0.0019	**0.4819**±0.0077	**1.1443**±0.0180	**0.7980**±0.0126	**99.18**±0.55	**94.25**±1.35	**94.37**±1.42

4 Theoretical Analysis

On a fixed candidate set, dot-product scoring represents bilinear edge logits with matrix $\mathbf{M} = \mathbf{W}_q^\top \mathbf{W}_k$. A rank-$r$ factorization realizes every bilinear target of rank at most r, while truncated SVD and the $1/4$-Lipschitz sigmoid yield a spectral approximation bound. The candidate mask supplies sparse support; the score function supplies weights within that support. Appendix B gives the formal statement, proof, and the score-margin condition for pruning.

The alignment objectives have different statistical roles. Standard population InfoNCE applies to unweighted sample-wise representations with independent marginal negatives (Poole et al., 2019). DMGraph's node loss instead uses an anchor-dependent hardness tilt and collision-prior adjustment, while each global representation depends on the shared minibatch context. We use these terms as training surrogates and measure their contribution through ablation. Appendix B.4 states the assumptions and bounds the distribution shift induced by the idealized hardness tilt.

5 Experiment

5.1 Experiment Settings

Datasets We use four multimodal graph benchmarks. **Generative**: WikiWeb2M (Burns et al., 2023) for webpage section summary generation and **FB15K237** (Toutanova et al., 2015) for relation-conditioned knowledge-graph generation. **Discriminative**: Ele-Fashion (Zhu et al., 2025) for product classification and **Goodreads-LP** (Zhu et al., 2025) for book recommendation link prediction. Together they cover web-scale and knowledge-base graphs, dense and sparse visual coverage, and both supervision types. Dataset construction and split details are provided in Appendix C.

Baselines Our baselines cover PLMs (OPT-125M (Zhang et al., 2022) and LLaMA-1B (Touvron et al., 2023)), inference-only VLM references (BLIP2-OPT-2.7B (Li et al., 2023b) and Qwen2-VL-7B (Wang et al., 2024)), and multimodal graph methods, including MMGL (Yoon et al., 2023), Graph4MM (Ning et al., 2025), and TMTE (Zhu et al., 2026). Within each backbone, DMGraph and trainable PLM-based baselines use the same PEFT method and budget: 20 prefix tokens for OPT-125M and LoRA rank 64 for LLaMA-1B and Qwen2.5-VL-7B. Inference-only VLM rows are

contextual references rather than optimization-matched competitors. Baseline configurations and evaluation mapping are detailed in Appendix D.

Implementation Details We apply Prefix-Tuning for OPT-125M and LoRA for LLaMA-1B / Qwen2.5-VL-7B. The vision encoder is CLIP. The default benchmark configuration caps subgraphs at 11 nodes and uses learning rate 1×10^{-4}, effective batch size 64, K_s=15, K_d=2, and λ_1=λ_2=0.1. This places the default configuration in the reweighting regime. Section 5.3 expands neighborhoods to 32 and 64 nodes to evaluate actual edge pruning. We use 20% warm-up followed by joint training on NVIDIA H20-NVLink GPUs, and report three-seed means for learned models. Additional preprocessing and training details are reported in Appendix G.

5.2 Overall Performance

Table 1 reports the main comparison on WikiWeb2M and Goodreads-LP, representative generative and discriminative benchmarks. DMGraph obtains the best reported score under both OPT-125M and LLaMA-1B, improving over the closest graph-multimodal competitors Graph4MM and TMTE. The comparison also separates content-conditioned structure learning from generic multimodal prompting and fixed graph integration. Complete results for FB15K237 and Ele-Fashion, backbone compatibility, and held-out-class analyses are provided in Appendix E–I.

The gains are consistent across task types rather than concentrated in one metric. With LLaMA-1B, DMGraph improves ROUGE-L on WikiWeb2M from 0.4731 to 0.4819 over Graph4MM and improves R-L on Goodreads-LP from 0.7729 to 0.7980. Under OPT-125M, the corresponding gains are 0.0056 and 0.0308. The larger margin on Goodreads-LP indicates that adaptive graph context is especially useful when the prediction depends on relational compatibility, while the WikiWeb2M gain shows that the same mechanism also improves long-context generation.

5.3 Structure Effectiveness under Pruning and Noise

Figure 2 tests whether content-conditioned edges remain useful when the candidate neighborhood is large enough to require pruning. We increase the neighborhood cap to $N_{\text{cap}} \in \{32, 64\}$ and vary $K_s \in \{5, 10, 15, \text{all}\}$ with LLaMA-1B. The comparison includes DMGraph, a hard Top-K gate using the same edge budget, Dynamic Dense with all candidate edges, and Graph4MM. For the corruption test, models trained on clean graphs are evaluated after random non-edges equal to 10%, 30%, or 50% of the original candidate set are added at test time.

An intermediate budget gives the strongest result in every expanded-neighborhood setting. At $K_s = 15$, DMGraph reaches 0.4812/0.4795 ROUGE-L on WikiWeb2M and 0.7955/0.7935 R-L on Goodreads-LP for caps 32/64. The corresponding gains over hard Top-K are 0.0141–0.0150 on WikiWeb2M and 0.0213–0.0217 on Goodreads-LP. DMGraph also exceeds Graph4MM and Dynamic Dense at both caps. Retaining all candidates reduces performance close to the dense baseline, linking the gain to learned neighborhood selection rather than edge scoring alone.

The same structure mechanism limits sensitivity to corrupted neighborhoods. With 50% injected edges, DMGraph drops by 15.4% on WikiWeb2M and 15.2% on Goodreads-LP, compared with 21.9%/22.1% for hard Top-K, 24.5%/24.7% for Graph4MM, and 30.7%/28.7% for Dynamic Dense. DMGraph retains 30.8% and 32.2% of the injected edges at this noise level, versus 43.5% and 45.2% for hard Top-K. Full secondary metrics, efficiency measurements, and edge-retention diagnostics are reported in Appendix F.

Mechanism and efficiency. At $K_s = 15$, the retained-edge ratio decreases from 78.2% with a 32-node cap to 47.6% with a 64-node cap, yet the same budget remains optimal on both datasets. At 50% noise, DMGraph retains 30.8%/32.2% of injected edges on WikiWeb2M/Goodreads-LP, compared with 43.5%/45.2% for hard Top-K. These results connect the gain to target-specific selection under a useful sparse support. With OPT-125M, DMGraph uses 0.2319/0.0699 seconds per training/generation batch, below Graph4MM at 0.2419/0.0745 and above MMGL at 0.2105/0.0595. Appendix H gives the full analysis.

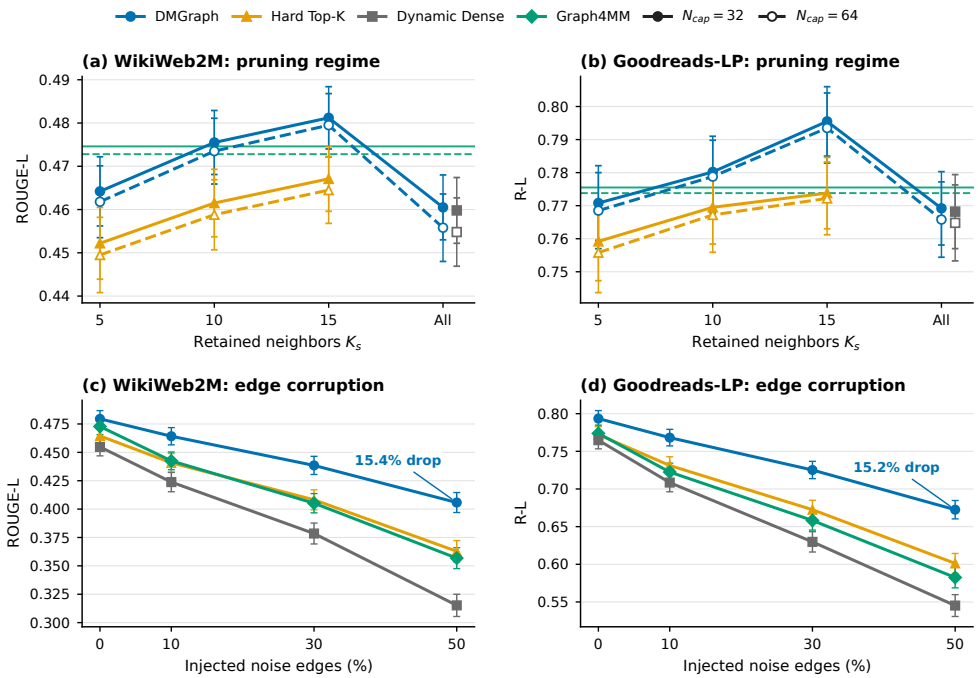

Figure 2: **Structure effectiveness in pruning and noisy-graph regimes.** (a,b) Task performance as the retained-neighbor budget varies for 32- and 64-node candidate neighborhoods. Filled solid and open dashed markers denote $N_{\text{cap}} = 32$ and 64; horizontal green lines show Graph4MM at the corresponding cap. (c,d) Performance after injecting random non-edges at test time with $N_{\text{cap}} = 64$ and $K_s = 15$. Points and error bars report mean $\pm$ standard deviation over three seeds.

5.4 Ablation Study

Figure 3 shows lower performance after removing each module across the evaluated backbones and task types. Content-conditioned edge reweighting contributes most to generation, while hierarchical alignment is more influential for discrimination. Parameter-matched replacements recover at most 0.0065 ROUGE-L, versus $\geq$ 0.0084 for the full module, separating the computational effect from parameter count. Appendix G.3 tests modality-specific projections, negative weighting, the edge gate, and the two coarser alignment losses; Appendix G.4 visualizes learned edge weights.

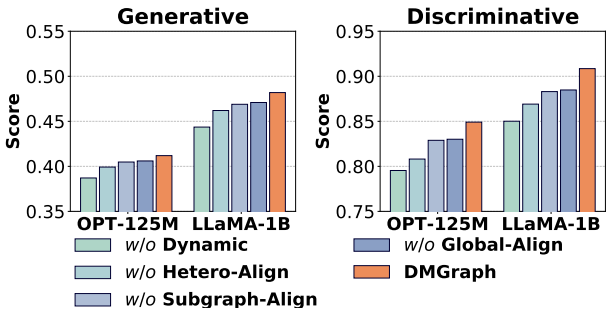

Figure 3: Ablation study in terms of ROUGE-L and R-L.

6 Conclusion

DMGraph learns content-conditioned neighborhoods jointly with node-, subgraph-, and batch-context alignment for multimodal graph prediction. This couples target-specific message passing with cross-modal consistency. Across four benchmarks and multiple backbones, DMGraph improves generative and discriminative performance over strong graph-multimodal baselines.

Expanded-neighborhood experiments locate the gain in selective structure: intermediate learned sparsity outperforms dense reweighting and hard Top-K, retains fewer injected edges, and preserves more performance under noise. Ablations and runtime measurements further show that edge selection and hierarchical alignment add complementary value without dense graph computation.

AI USE STATEMENT

The authors used generative AI tools for limited language editing, checklist-style self-review, and LaTeX migration assistance. The research ideas, method design, theorem statements, proofs, experimental implementation, data processing, numerical results, figures, tables, and citations were created by the authors. All AI-assisted edits were manually reviewed, and the authors take responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The method definition, losses, hyperparameters, dataset splits, baseline settings, and implementation details are reported in Sections 3–5 and Appendix G. Within each backbone, trainable PLM-based comparisons use the same parameter-efficient tuning method and budget. We report three-seed averages where available and identify evaluation regimes where saturated discriminative metrics limit interpretability. Code, preprocessing scripts, sampled WikiWeb2M page identifiers, and cached subgraph splits will be released as anonymized supplementary material.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Andrea Burns, Krishna Srinivasan, Joshua Ainslie, Geoff Brown, Bryan A Plummer, Kate Saenko, Jianmo Ni, and Mandy Guo. Wikiweb2m: A page-level multimodal wikipedia dataset. *arXiv preprint arXiv:2305.05432*, 2023.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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*, volume 33, pp. 19314–19326, 2020.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Yasha Ektefaie, George Dasoulas, Ayush Noori, Manan Bhargava, and Marinka Zitnik. Multimodal learning with graphs. *Nature Machine Intelligence*, 5(4):340–350, 2023.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1972–1982, 2019.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *International Conference on Learning Representations*, 2024.

Xiaobin Hong, Mingkai Lin, Xiaoli Wang, Chaoqun Wang, and Wenzhong Li. Multimodal graph representation learning with dynamic information pathways. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14829–14837, 2026.

Sein Kim, Namkyeong Lee, Junseok Lee, Dongmin Hyun, and Chanyoung Park. Heterogeneous graph learning for multi-modal medical data analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 5141–5150, 2023.

Jiang Li, Xiaoping Wang, Guoqing Lv, and Zhigang Zeng. Graphmft: A graph network based multimodal fusion technique for emotion recognition in conversation. *Neurocomputing*, 550: 126427, 2023a.

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, 2023b.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17612–17625, 2022.

Jiajin Liu, Dongzhe Fan, Jiacheng Shen, Chuanhao Ji, Daochen Zha, and Qiaoyu Tan. Graph-mllm: harnessing multimodal large language models for multimodal graph learning. *arXiv preprint arXiv:2506.10282*, 2025.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 188–197, 2019.

Xuying Ning, Dongqi Fu, Tianxin Wei, Wujiang Xu, and Jingrui He. Graph4mm: Weaving multimodal learning with structural information. In *Forty-second International Conference on Machine Learning*, 2025.

Jinyoung Park, Minseong Bae, Dohwan Ko, and Hyunwoo J Kim. Llamo: Large language model-based molecular graph assistant. *Advances in Neural Information Processing Systems*, 37: 131972–132000, 2024.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8238–8247, 2022.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pp. 5171–5180. PMLR, 2019.

Chendi Qian, Andrei Manolache, Kareem Ahmed, Zhe Zeng, Guy Van den Broeck, Mathias Niepert, and Christopher Morris. Probabilistically rewired message-passing neural networks. In *International Conference on Learning Representations*, 2024.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 2443–2449, 2021.

Jan Toenshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. Graph neural networks for maximum constraint satisfaction. *Frontiers in artificial intelligence*, 3:580607, 2021.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, 2015.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12695–12705, 2020.

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

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*, volume 35, pp. 27387–27401, 2022.

Qitian Wu, Chenxiao Chen, Zijie Yang, Yixuan He, David Wipf, and Junchi Yan. DIFFormer: Scalable (graph) transformers induced by energy constrained diffusion. In *International Conference on Learning Representations*, 2023.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. *Advances in neural information processing systems*, 33:20520–20531, 2020.

Jinfeng Xu, Zheyu Chen, Wei Wang, Xiping Hu, Sang-Wook Kim, and Edith CH Ngai. Cohesion: Composite graph convolutional network with dual-stage fusion for multimodal recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1830–1839, 2025a.

Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Wei Wang, Xiping Hu, Steven Hoi, and Edith Ngai. A survey on multimodal recommender systems: Recent advances and future directions. *arXiv preprint arXiv:2502.15711*, 2025b.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Aber, and Xing Xie. Graphformers: Gnn-nested transformers for representation learning on textual graph. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28798–28810, 2021.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888, 2021.

Minji Yoon, Jing Yu Koh, Bryan Hooi, and Ruslan Salakhutdinov. Multimodal graph learning for generative tasks. *arXiv preprint arXiv:2310.07478*, 2023.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Jianan Zhao, Qianlong Wen, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. Self-supervised graph structure refinement for graph neural networks. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pp. 159–167, 2023.

Zhiyao Zhou, Sheng Zhou, Bochao Mao, Xuanyi Zhou, Jiawei Chen, Qiaoyu Tan, Daochen Zha, Yan Feng, Chun Chen, and Can Wang. OpenGSL: A comprehensive benchmark for graph structure learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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.

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.

APPENDIX

A NOTATION

Table 2: Summary of recurring notation.

Symbol	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}^t, \mathcal{X}^v)$	Multimodal attributed graph
$\mathbf{A}$	Static adjacency matrix
$\hat{\mathbf{A}}^m$	Content-conditioned dense edge-score matrix for modality m
$\tilde{\mathbf{A}}^m$	Gated dynamic adjacency matrix for modality m
$\mathbf{H}^m$	Input node embeddings for modality m
$\mathbf{Z}^{m,(k)}$	Diffused node embeddings after k steps
$\mathbf{s}^m$ / $\mathbf{g}^m$	Subgraph-level / batch-global representation
K_s / K_d	Top-K gate budget / diffusion depth
λ_1 / λ_2	Alignment-loss / structure-regularization weight
τ, τ_s, τ_g	Node, subgraph, and batch-global contrastive temperatures
τ^+	Semantic collision prior for false-negative correction

A.1 PROBLEM SETUP AND TASK OBJECTIVES

Definition A.1 (Multimodal attributed graph)**.** A multimodal attributed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}^t, \mathcal{X}^v)$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set with adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$. Textual attributes are denoted by $\mathcal{X}^t = \{\mathbf{x}_i^t\}_{i=1}^n$, where $\mathbf{x}_i^t \in \mathbb{R}^{d_t}$ or $\mathbf{x}_i^t = \varnothing$. Visual attributes are denoted by $\mathcal{X}^v = \{\mathbf{x}_i^v\}_{i=1}^n$, where $\mathbf{x}_i^v \in \mathbb{R}^{d_v}$ or $\mathbf{x}_i^v = \varnothing$. We follow Graph4MM (Ning et al., 2025) for the static multimodal attributed graph setting.

Modality-specific subgraph induction. Given a target node v_i and hop distance τ, we induce modality-specific subgraphs that retain neighbors with available attributes in corresponding modality:

$$\begin{aligned} \mathcal{G}_i^t &= (\mathcal{V}_i^t, \mathcal{E}_i^t), \quad \mathcal{V}_i^t = \{v \in \mathcal{V} \mid \text{dist}(v, v_i) \le \tau, \mathbf{x}_v^t \neq \varnothing\}, \\ \mathcal{G}_i^v &= (\mathcal{V}_i^v, \mathcal{E}_i^v), \quad \mathcal{V}_i^v = \{v \in \mathcal{V} \mid \text{dist}(v, v_i) \le \tau, \mathbf{x}_v^v \neq \varnothing\}, \end{aligned} \tag{20}$$

where $\mathcal{E}_i^m = \{(u,v) \in \mathcal{E} \mid u, v \in \mathcal{V}_i^m\}$ for $m \in \{t, v\}$, and $\text{dist}(\cdot,\cdot)$ denotes shortest-path distance in $\mathcal{G}$.

Task objectives. For generative tasks, given target node v_i, instruction $\mathcal{I}$, and induced subgraphs $\mathcal{G}_i^t, \mathcal{G}_i^v$, the model generates response $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_L)$ by maximizing

$$\hat{y} = \arg\max_{y} \prod_{\ell=1}^{L} P_\theta(y_\ell \mid y_{<\ell}, \mathcal{G}_i^t, \mathcal{G}_i^v, \mathcal{I}). \tag{21}$$

For discriminative tasks, the model predicts class $c^* \in \{c_1, \ldots, c_{\mathcal{C}}\}$ through

$$c^* = \arg\max_{c_j} \phi\big(\text{PLM}_\theta(\mathcal{G}_i^t, \mathcal{G}_i^v, \mathcal{I}), \mathbf{d}_{c_j}\big), \tag{22}$$

where $\phi(\cdot,\cdot)$ is a similarity function and $\mathbf{d}_{c_j}$ is the embedding of the class description.

B ADDITIONAL THEORETICAL DETAILS

B.1 SCOPE OF THE DYNAMIC ADJACENCY RESULT

Theorem B.2 is stated on the valid candidate set $\mathcal{C}$ of an induced subgraph:

$$\mathbf{A}_{ij}^*(\mathbf{H}) = \mathbf{1}[(i,j) \in \mathcal{C}]\, \sigma\big(\mathbf{h}_i^\top \mathbf{M} \mathbf{h}_j / \sqrt{d} + \beta^* \mathbf{A}_{ij}^{\text{static}}\big).$$

The candidate mask supplies exact zeros outside $\mathcal{C}$; the sigmoid assigns a weight in $(0, 1)$ only to valid candidate pairs. This distinction is necessary because an unmasked sigmoid matrix is dense and cannot itself define a sparse support. The bilinear score and scalar structural prior match the parameterization in Section 3; arbitrary nonlinear pairwise functions lie outside this class.

For a rank-r factorization $\mathbf{M}_r = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top$, choose

$$\mathbf{W}_q = \mathbf{U}_r^\top, \qquad \mathbf{W}_k = \mathbf{\Sigma}_r \mathbf{V}_r^\top.$$

Then

$$(\mathbf{W}_q \mathbf{h}_i)^\top (\mathbf{W}_k \mathbf{h}_j)/\sqrt{d} = \mathbf{h}_i^\top \mathbf{M}_r \mathbf{h}_j/\sqrt{d}.$$

When $r = d$, this realizes any bilinear target exactly. More generally, exact realization only requires $r \geq \mathrm{rank}(\mathbf{M})$. A lower-rank approximation is controlled by Eq. (24). The scalar static prior is recovered by setting $\beta = \beta^*$.

Assumption B.1 (Candidate-domain conditions)**.** (i) The node embedding space $\mathcal{H} \subset \mathbb{R}^{n \times d}$ is compact with $\sup_{\mathbf{H} \in \mathcal{H}} \|\mathbf{H}\|_F \leq R$ for $R > 0$. (ii) The candidate set $\mathcal{C} \subseteq \{1, \ldots, n\}^2$ is fixed for an induced subgraph, contains r_i valid entries in row i, and masks entries outside $\mathcal{C}$ to zero.

Theorem B.2 (Bilinear realization on the candidate set)**.** *Let $\mathcal{F}_{\mathrm{bilinear}}(\mathcal{C})$ denote masked edge functions*

$$\mathbf{A}^*_{ij}(\mathbf{H}) = \mathbf{1}[(i,j) \in \mathcal{C}]\, \sigma\big(\mathbf{h}_i^\top \mathbf{M} \mathbf{h}_j/\sqrt{d} + \beta^* \mathbf{A}^{\textit{static}}_{ij}\big). \tag{23}$$

In the reweighting regime $K_s \geq \max_i r_i$, query and key maps in $\mathbb{R}^{r \times d}$ realize every target with $\mathrm{rank}(\mathbf{M}) \leq r$. For the rank-$r$ truncated SVD $\mathbf{M}_r$, the resulting adjacency satisfies

$$\sup_{\mathbf{H} \in \mathcal{H}} \|\tilde{\mathbf{A}}_r(\mathbf{H}) - \mathbf{A}^*(\mathbf{H})\|_F \leq \frac{R^2 \sqrt{|\mathcal{C}|}}{4\sqrt{d}} \|\mathbf{M} - \mathbf{M}_r\|_2. \tag{24}$$

B.2 Reweighting and Pruning Regimes

Let r_i denote the number of valid candidate entries in row i of the induced subgraph. The threshold in Eq. (3) gives two regimes.

- If $K_s \leq r_i$, the gate converges to a hard top-K_s mask as $\epsilon \to 0$ and prunes candidate edges.
- If $K_s > r_i$, the threshold lies below the smallest valid entry, so no valid edge is pruned. The gate remains monotone in the learned score and acts as differentiable edge reweighting.

The default benchmark configuration caps subgraphs at 11 nodes and uses $K_s = 15$, placing it in the reweighting regime. The structural evaluation in Section 5.3 raises the cap to 32 and 64 and varies $K_s \in \{5, 10, 15, \mathrm{all}\}$, directly testing pruning as well as dense reweighting.

Support stability in the pruning regime requires a score margin. Let $u^*_{i,(K_s)}$ and $u^*_{i,(K_s+1)}$ denote the K_s-th and $(K_s{+}1)$-th largest target logits among the candidates in row i. If $u^*_{i,(K_s)} - u^*_{i,(K_s+1)} > 2\delta$ and the learned logits approximate all candidate logits within δ, then their top-K_s sets coincide. This statement concerns ranking within $\mathcal{C}$; it does not treat positive sigmoid outputs as sparse support.

B.3 Complexity and the Role of K_s

Computing attention scores in Eq. (1) requires $O(n_m^2 d)$ multiply-adds per induced subgraph, where n_m is the number of modality-specific nodes and d is the embedding dimension. Each diffusion step costs $O(n_m K_s' d)$ with $K_s' = \min(K_s, r_i)$, giving total per-subgraph cost

$$O(n_m^2 d + K_d n_m K_s' d).$$

For the default setting ($n_m \leq 11$, $K_s = 15$, $K_d = 2$, $d = 768$), the attention term contributes roughly 9×10^4 multiply-adds per subgraph and diffusion contributes a comparable amount. Sorting and gate computation are negligible at this subgraph size. The attention cost includes feature dimension d as well as the $n_m^2 = 121$ node-pair interactions.

The deployment-relevant scaling is mainly in the number and size of induced subgraphs. Table 12 reports the default-cap cost, while Appendix F reports inference time and memory for 32- and 64-node neighborhoods. When $n_m > K_s$, the same parameterization becomes a top-K_s pruning operator; the measured results show that this regime improves both task performance and inference cost relative to dense reweighting.

B.4 Scope of the Contrastive Interpretation

The standard population InfoNCE result assumes that each representation is a function of its own sample and that negatives are independently drawn from the corresponding marginal. Under these conditions,

$$I(\mathbf{R}^t; \mathbf{R}^v) \geq \log N - \mathcal{L}_{\mathrm{NCE}}^{\mathbf{R},p}(N).$$

This gives a reference interpretation for the unweighted node objective and for subgraph summaries under independent graph sampling. It does not apply directly to Eq. (16): every $\mathbf{g}_b^m$ contains the batch context $\mathbf{c}^m$, so the contrasted representations are coupled through the same minibatch. We use $\mathcal{L}_g$ as a batch-context consistency objective and evaluate its value through ablation.

The implemented node loss uses an anchor-dependent hardness tilt, an estimated collision prior, and clipping. Proposition B.4 characterizes the distribution shift caused by the idealized tilt. The implemented objective is evaluated as a training surrogate through the component ablations.

B.5 Proof of Theorem B.2

Proof. **Step 1: exact bilinear realization.** The dot-product attention score

$$s_{ij}(\mathbf{H}; \Theta) = (\mathbf{W}_q \mathbf{h}_i)^\top (\mathbf{W}_k \mathbf{h}_j)/\sqrt{d}$$

realizes a bilinear target $f(\mathbf{h}_i, \mathbf{h}_j) = \mathbf{h}_i^\top \mathbf{M} \mathbf{h}_j/\sqrt{d}$. Let $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be its thin singular value decomposition. Setting $\mathbf{W}_q = \mathbf{\Sigma}^{1/2}\mathbf{U}^\top$ and $\mathbf{W}_k = \mathbf{\Sigma}^{1/2}\mathbf{V}^\top$ gives

$$(\mathbf{W}_q \mathbf{h}_i)^\top (\mathbf{W}_k \mathbf{h}_j)/\sqrt{d} = \mathbf{h}_i^\top \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \mathbf{h}_j/\sqrt{d} = \mathbf{h}_i^\top \mathbf{M} \mathbf{h}_j/\sqrt{d}.$$

Thus any target with $\mathrm{rank}(\mathbf{M}) \leq r$ is realized exactly by maps in $\mathbb{R}^{r \times d}$. For a rank-r truncation $\mathbf{M}_r$, the logit residual satisfies

$$\sup_{\mathbf{H} \in \mathcal{H}} \left| \mathbf{h}_i^\top (\mathbf{M} - \mathbf{M}_r) \mathbf{h}_j \right| / \sqrt{d} \leq \frac{R^2}{\sqrt{d}} \|\mathbf{M} - \mathbf{M}_r\|_2.$$

The sigmoid is $1/4$-Lipschitz. Summing the squared entry-wise errors over $\mathcal{C}$ gives

$$\sup_{\mathbf{H} \in \mathcal{H}} \|\tilde{\mathbf{A}}_r(\mathbf{H}) - \mathbf{A}^*(\mathbf{H})\|_F \leq \frac{R^2 \sqrt{|\mathcal{C}|}}{4\sqrt{d}} \|\mathbf{M} - \mathbf{M}_r\|_2.$$

Step 2: static prior and candidate mask. The static term $\beta^* \mathbf{A}_{ij}^{\mathrm{static}}$ is recovered exactly by setting $\beta = \beta^*$ in Eq. (2). In the reweighting regime, the forward gate is one on every valid candidate, so the candidate mask sets entries outside $\mathcal{C}$ to zero and leaves the realized sigmoid weights unchanged inside $\mathcal{C}$. This proves exact realization and the rank-r bound without assigning zero support to sigmoid outputs. □

Theorem B.2 characterizes the function class represented by dot-product attention on a fixed candidate set, including finite-rank bilinear similarities. Its rank-controlled approximation term is

$$\epsilon_{\mathrm{approx}}(r) = \frac{R^2 \sqrt{|\mathcal{C}|}}{4\sqrt{d}} \sigma_{r+1}(\mathbf{M}), \tag{25}$$

where $\sigma_{r+1}(\mathbf{M}) = \|\mathbf{M} - \mathbf{M}_r\|_2$. On WikiWeb2M, 50 singular values capture 97.1% of the empirical attention-matrix energy. Replacing bilinear scoring with a 2-layer MLP ($3\times$ parameters) gives near-identical performance (0.4825 vs. 0.4819 ROUGE-L), suggesting that the bilinear parameterization is adequate for the evaluated setting.

B.6 Lipschitz Continuity of Dynamic Adjacency

Proposition B.3 (Lipschitz continuity)**.** *Under bounded weights $\|\mathbf{W}_q\|_2, \|\mathbf{W}_k\|_2 \leq B_W$ and Assumption B.1(i), the dynamic adjacency satisfies*

$$\|\tilde{\mathbf{A}}(\mathbf{H}; \Theta) - \tilde{\mathbf{A}}(\mathbf{H}'; \Theta)\|_F \leq L_{\mathbf{A}} \|\mathbf{H} - \mathbf{H}'\|_F,$$

where $L_{\mathbf{A}} = O(B_W^2 R/\sqrt{d})$ for fixed induced-subgraph size.

Proof. The unnormalized score $s_{ij}(\mathbf{H}) = (\mathbf{W}_q\mathbf{h}_i)^\top(\mathbf{W}_k\mathbf{h}_j)/\sqrt{d}$ is bilinear in $(\mathbf{h}_i, \mathbf{h}_j)$. Differentiating and using sub-multiplicativity of operator norms gives

$$\left\|\frac{\partial s_{ij}}{\partial \mathbf{h}_i}\right\|_2 \le \|\mathbf{W}_q\|_2\|\mathbf{W}_k\|_2\|\mathbf{h}_j\|_2/\sqrt{d} \le B_W^2 R/\sqrt{d},$$

and the same bound holds for $\partial s_{ij}/\partial \mathbf{h}_j$. The sigmoid in $\hat{\mathbf{A}}_{ij}^m = \sigma(\hat{s}_{ij}^m)$ is $1/4$-Lipschitz, and the soft gate $g_{ij}^m = \sigma(\frac{\hat{\mathbf{A}}_{ij}^m - \mu_i^m}{\epsilon})$ is Lipschitz up to constants governed by the annealed temperature. Summing the resulting entry-wise bounds over the bounded induced subgraph gives the stated Frobenius bound. □

B.7 Hard-Negative Distribution Shift

Proposition B.4 (Bounding hard-negative distribution shift)**.** *Suppose the squared distance $c(v) = \|\mathbf{z}_i^v - \mathbf{z}^v\|^2$ is bounded by M on the support of p. Under hard-negative reweighting $q(v) = e^{-\rho c(v)}p(v)/\mathbb{E}_p[e^{-\rho c}]$ with $\rho \ge 0$,*

$$\Delta_{\mathrm{hard}} = D_{\mathrm{KL}}(q\|p) \le e^{\rho^2 M^2/2} - 1 = O(\rho^2 M^2), \tag{26}$$

and for small ρ,

$$\Delta_{\mathrm{hard}} = \frac{\rho^2}{2}\mathrm{Var}_p[c] + O(\rho^3). \tag{27}$$

Proof. By the Donsker–Varadhan representation and the standard inequality $D_{\mathrm{KL}} \le \chi^2$,

$$D_{\mathrm{KL}}(q\|p) \le \chi^2(q,p) = \frac{\mathbb{E}_p[e^{-2\rho c}]}{\big(\mathbb{E}_p[e^{-\rho c}]\big)^2} - 1.$$

Let $\xi(\theta) = \mathbb{E}_p[e^{-\theta c}]$. Jensen's inequality gives $\xi(\rho)^2 \ge e^{-2\rho\mathbb{E}_p[c]}$. Hoeffding's lemma gives $\xi(2\rho) \le e^{-2\rho\mathbb{E}_p[c] + \rho^2 M^2/2}$ because $c - \mathbb{E}_p[c]$ lies in an interval of length at most M. Hence $\xi(2\rho)/\xi(\rho)^2 \le e^{\rho^2 M^2/2}$, which proves Eq. (26). The expansion in Eq. (27) follows from the second-order Taylor expansion $\log \xi(\theta) = -\theta\mathbb{E}_p[c] + \theta^2\mathrm{Var}_p[c]/2 + O(\theta^3)$. □

Role of the collision-prior adjustment. Equation (11) reduces the contribution of samples estimated to share the anchor's semantic class, while N_i places the normalized statistic on the total-negative scale. Hardness weighting and clipping make this a biased, anchor-dependent optimization statistic. Removing the adjustment reduces WikiWeb2M ROUGE-L from 0.4819 to 0.4653 in Table 11.

C Dataset Details

WikiWeb2M. WikiWeb2M (Burns et al., 2023) is a multimodal webpage dataset extending WIT (Srinivasan et al., 2021) with page-level structure, section text, images, captions, and hierarchical relationships. We follow MMGL (Yoon et al., 2023) and use section summary generation, where the model generates a missing first sentence from section text, page context, images, and graph-neighbor context. Due to storage constraints, we sample 10K Wikipedia pages, yielding 13,539 training samples and 1,768 test samples. The sampled page identifiers and preprocessing cache will be released with the anonymized supplementary material.

FB15K237. FB15K237 (Toutanova et al., 2015) is used as a knowledge-graph generation benchmark with text and visual entity descriptions. It provides a denser connectivity pattern and shorter textual context than WikiWeb2M, which helps test whether dynamic edge weighting remains useful outside long webpage inputs.

Ele-Fashion. Ele-Fashion (Zhu et al., 2025) is a multimodal product classification dataset derived from Amazon-Fashion (Ni et al., 2019). Nodes represent products, edges encode co-purchase relationships, and node attributes include product text and images. The graph has 11 categories, 97,766

nodes, and 199,602 edges. For OPT-125M, we train on 6 classes and evaluate zero-shot on 5 held-out classes. For LLaMA-1B, we train on 2 classes and evaluate on 9 held-out classes to increase difficulty.

Goodreads-LP. Goodreads-LP (Zhu et al., 2025) is a book recommendation link-prediction benchmark with sparser visual attributes than Ele-Fashion. It provides a less saturated discriminative setting and is therefore more informative than Ele-Fashion accuracy for comparing graph-multimodal methods.

D Baseline and Evaluation Details

We compare PLMs, VLMs, MMGL variants (Yoon et al., 2023), Graph4MM (Ning et al., 2025), and TMTE (Zhu et al., 2026).

PLMs. We evaluate OPT-125M (Zhang et al., 2022) and LLaMA-1B (Touvron et al., 2023). The input settings include Node's Text and Subgraph's Text. PLM-based methods use matched parameter-efficient tuning settings.

VLMs. We evaluate BLIP2-OPT-2.7B (Li et al., 2023b) and Qwen2-VL-7B-Instruct (Wang et al., 2024) in inference-only settings. These baselines are not fine-tuned, so they are useful as generic multimodal references but are not capacity- and optimization-matched competitors.

MMGL variants. We evaluate Node's Text, Subgraph's Text, Node's Text & Image, Subgraph's Text & Image, and Subgraph's T & I + GNN. The GNN variant maps text and visual features into graph embeddings and appends them to the PLM input.

Graph4MM variants. We evaluate Hop-aware MM-QFormer and Hop-diffused MM-QFormer. These are the closest baselines because they also integrate graph structure with a multimodal querying interface.

TMTE. We include TMTE as a close baseline for task-aware modality and topology co-evolution. Comparable TMTE runs are reported for the representative WikiWeb2M and Goodreads-LP settings used in the main comparison.

Discriminative evaluation. The model generates a textual class response, which is mapped to a class label by normalized string matching. We report ROUGE-L (R-L) between generated and target class strings, as well as categorical accuracy, recall, and precision. Because Ele-Fashion categorical metrics saturate for several methods, we interpret them cautiously and rely on R-L and Goodreads-LP for finer comparisons.

E Full Main Results

The main text reports one representative dataset from each task family to keep the comparison readable within the page budget. Tables 3 and 4 provide the complete benchmark coverage. WikiWeb2M and Goodreads-LP are kept in the main table because they are the most diagnostic settings for generation and discrimination. FB15K237 complements WikiWeb2M by testing relation-conditioned generation on a denser knowledge graph. Ele-Fashion complements Goodreads-LP by testing zero-shot product classification, but its categorical metrics saturate for strong methods.

The full tables preserve the same trend as the representative results. DMGraph consistently improves over the closest Graph4MM variants under matched backbone settings, while generic VLMs remain weaker than tuned PLM-based graph methods. The available TMTE results provide an additional task-aware co-evolution comparison on WikiWeb2M and Goodreads-LP, where DMGraph remains strongest across the reported metrics. For discrimination, Goodreads-LP gives the cleaner ranking, while Ele-Fashion is best treated as supporting evidence through R-L and held-out-class robustness.

E.1 Backbone Compatibility

Table 5 evaluates DMGraph with OPT-125M, LLaMA-1B, and Qwen2.5-VL-7B (Bai et al., 2025). Performance increases across these configurations on all datasets, showing that content-conditioned edge reweighting and hierarchical alignment remain compatible with stronger text- and vision-

Table 3: Full generative results on WikiWeb2M and FB15K237. BLEU-4, ROUGE-L, and CIDEr evaluate language modeling accuracy (higher is better). Gray small values denote standard deviations. Best and second-best results are **bolded** and underlined.

	Backbone	Variant	WikiWeb2M			FB15K237		
			BLEU-4	ROUGE-L	CIDEr	BLEU-4	ROUGE-L	CIDEr
VLMs	BLIP	Node's Text	0.0000	0.0496	0.0060	0.0021	0.0834	0.0142
		Subgraph's Text	0.0000	0.0530	0.0063	0.0035	0.0901	0.0170
	QwenVL	Node's Text	0.0000	0.1223	0.0069	0.0102	0.1657	0.0294
		Subgraph's Text	0.0000	0.1192	0.0084	0.0118	0.1738	0.0331
OPT-125M	PLM	Node's Text	0.0078±0.0001	0.1871±0.0024	0.0783±0.0008	0.0356±0.0005	0.3274±0.0043	0.2185±0.0026
		Subgraph's Text	0.0164±0.0002	0.2198±0.0031	0.1551±0.0024	0.0512±0.0008	0.3681±0.0052	0.3047±0.0044
	MMGL	Node's Text	0.0642±0.0008	0.3807±0.0055	0.6241±0.0081	0.1342±0.0017	0.5126±0.0073	0.7458±0.0101
		Subgraph's Text	0.0770±0.0011	0.3992±0.0051	0.7606±0.0120	0.1517±0.0022	0.5394±0.0087	0.8521±0.0132
		Node's Text & Image	0.0643±0.0009	0.3825±0.0060	0.6371±0.0096	0.1361±0.0015	0.5158±0.0079	0.7563±0.0114
		Subgraph's Text & Image	0.0778±0.0010	0.4041±0.0069	0.7712±0.0116	0.1542±0.0025	0.5462±0.0075	0.8634±0.0140
		Subgraph's T & I + GNN	0.0633±0.0010	0.3814±0.0058	0.6326±0.0089	0.1289±0.0019	0.5071±0.0081	0.7392±0.0107
	Graph4MM	Hop-Aware MM-QFormer	0.0801±0.0012	0.4063±0.0055	0.7736±0.0124	0.1583±0.0023	0.5567±0.0090	0.8794±0.0141
		Hop-Diffused MM-QFormer	0.0800±0.0010	0.4076±0.0067	0.7831±0.0120	0.1604±0.0026	0.5641±0.0085	0.8917±0.0138
	TMTE	Task-aware Co-evolution	0.0754±0.0011	0.3969±0.0059	0.7580±0.0111	0.1484±0.019	0.5122±0.0068	0.8208±0.0117
	DMGraph		**0.0834**±0.0013	**0.4119**±0.0064	**0.7994**±0.0127	**0.1672**±0.0024	**0.5863**±0.0096	**0.9213**±0.0148
LLaMA-1B	PLM	Node's Text	0.0757±0.0010	0.3691±0.0052	0.7185±0.0101	0.1621±0.0023	0.4982±0.0071	0.8736±0.0125
		Subgraph's Text	0.0960±0.0015	0.3949±0.0057	0.9179±0.0142	0.1843±0.0029	0.5347±0.0083	1.0125±0.0160
	MMGL	Node's Text	0.0947±0.0013	0.4375±0.0061	0.9046±0.0136	0.1816±0.0025	0.5689±0.0087	1.0287±0.0149
		Subgraph's Text	0.1087±0.0016	0.4426±0.0073	1.0426±0.0161	0.1972±0.0031	0.5841±0.0094	1.1196±0.0174
		Node's Text & Image	0.0981±0.0014	0.4449±0.0067	0.9367±0.0145	0.1845±0.0027	0.5728±0.0080	1.0431±0.0158
		Subgraph's Text & Image	0.1157±0.0018	0.4685±0.0075	1.1072±0.0176	0.2086±0.0035	0.6128±0.0099	1.1842±0.0181
		Subgraph's T & I + GNN	0.1003±0.0015	0.4487±0.0065	0.9631±0.0139	0.1769±0.0024	0.5536±0.0079	1.0049±0.0151
	Graph4MM	Hop-Aware MM-QFormer	0.1186±0.0017	0.4731±0.0072	1.1262±0.0179	0.2138±0.0032	0.6304±0.0101	1.2197±0.0189
		Hop-Diffused MM-QFormer	0.1177±0.0016	0.4713±0.0068	1.1221±0.0163	0.2127±0.0030	0.6286±0.0093	1.2173±0.0176
	TMTE	Task-aware Co-evolution	0.1141±0.0018	0.4633±0.0070	1.0825±0.0168	0.2001±0.0028	0.5919±0.0090	1.1219±0.0165
	DMGraph		**0.1234**±0.0019	**0.4819**±0.0077	**1.1443**±0.0180	**0.2218**±0.0036	**0.6619**±0.0109	**1.2741**±0.0204

Table 4: Full discriminative results on Ele-Fashion and Goodreads-LP. ROUGE-L (R-L) measures response accuracy; Accuracy (Acc), Recall (Rec), and Precision (Pre) assess classification and link prediction. Gray small values denote standard deviations. Best and second-best results are **bolded** and underlined.

	Backbone	Variant	Ele-Fashion				Goodreads-LP			
			R-L	Acc (%)	Rec (%)	Pre (%)	R-L	Acc (%)	Rec (%)	Pre (%)
VLMs	BLIP	Node's Text	0.1450	21.89	9.62	16.27	0.1238	18.74	8.13	14.05
		Subgraph's Text	0.1907	31.37	14.49	31.73	0.1672	27.45	12.31	27.86
	QwenVL	Node's Text	0.0793	11.00	4.43	10.90	0.0689	9.42	3.78	9.67
		Subgraph's Text	0.1233	12.33	5.78	14.26	0.1071	10.81	4.97	12.43
OPT-125M	PLM	Node's Text	0.2270±0.0032	68.57±0.95	40.44±0.55	48.76±0.72	0.2089±0.0027	60.32±0.80	35.71±0.48	43.24±0.57
		Subgraph's Text	0.2762±0.0041	67.68±1.02	33.14±0.52	40.26±0.61	0.2541±0.0036	59.84±0.91	29.38±0.42	35.71±0.51
	MMGL	Node's Text	0.7047±0.0102	70.33±1.12	66.67±0.94	66.67±1.01	0.6314±0.0087	65.42±0.88	60.81±0.82	62.18±0.93
		Subgraph's Text	0.8149±0.0131	99.74±0.19	71.03±1.06	71.43±1.15	0.7038±0.0099	92.45±1.42	65.27±0.91	66.04±0.99
		Node's Text & Image	0.7180±0.0098	73.90±1.05	68.67±1.18	83.33±1.34	0.6428±0.0084	67.18±0.98	62.41±1.00	76.85±1.18
		Subgraph's Text & Image	0.8144±0.0126	99.85±0.13	83.25±1.42	83.33±1.21	0.7042±0.0107	92.61±1.29	75.83±1.09	76.92±1.13
		Subgraph's T & I + GNN	0.5771±0.0081	70.89±0.99	56.08±0.84	62.50±0.91	0.5193±0.0072	63.74±0.83	50.27±0.76	56.84±0.81
	Graph4MM	Hop-Aware MM-QFormer	0.8097±0.0118	100.00±0.00	100.00±0.00	100.00±0.00	0.7081±0.0096	95.83±1.31	92.74±1.25	92.81±1.34
		Hop-Diffused MM-QFormer	0.8282±0.0124	100.00±0.00	100.00±0.00	100.00±0.00	0.7139±0.0109	96.27±1.46	93.42±1.31	93.51±1.40
	TMTE	Task-aware Co-evolution	–	–	–	–	0.7072±0.0102	92.17±1.24	84.12±1.17	85.29±1.25
	DMGraph		**0.8491**±0.0143	**100.00**±0.00	**100.00**±0.00	**100.00**±0.00	**0.7389**±0.0115	**97.84**±1.51	**95.71**±1.38	**95.83**±1.44
LLaMA-1B	PLM	Node's Text	0.7081±0.0099	94.76±1.33	78.07±1.08	83.11±1.18	0.6347±0.0088	87.52±1.19	71.84±1.04	76.93±1.09
		Subgraph's Text	0.7327±0.0106	99.79±0.12	81.42±1.27	81.27±1.14	0.6582±0.0094	92.36±1.35	74.93±1.05	75.18±1.12
	MMGL	Node's Text	0.7083±0.0101	94.80±1.29	78.12±1.10	83.19±1.22	0.6358±0.0089	87.61±1.23	71.92±1.02	77.04±1.15
		Subgraph's Text	0.7375±0.0113	99.84±0.13	81.51±1.20	81.58±1.25	0.6627±0.0098	92.41±1.31	75.06±1.13	75.27±1.08
		Node's Text & Image	0.6696±0.0091	97.26±1.40	84.33±1.31	88.38±1.44	0.6018±0.0082	89.74±1.17	77.85±1.16	81.42±1.22
		Subgraph's Text & Image	0.7446±0.0118	98.07±1.45	84.55±1.23	85.50±1.30	0.6691±0.0103	90.83±1.25	78.04±1.11	79.13±1.19
		Subgraph's T & I + GNN	0.3823±0.0055	86.73±1.13	62.80±0.89	61.44±0.92	0.3438±0.0048	79.25±1.08	57.18±0.81	56.31±0.85
	Graph4MM	Hop-Aware MM-QFormer	0.8891±0.0141	99.87±0.09	89.86±1.34	89.88±1.29	0.7729±0.0119	98.42±1.50	91.36±1.27	91.45±1.32
		Hop-Diffused MM-QFormer	0.8822±0.0135	100.00±0.00	100.00±0.00	100.00±0.00	0.7702±0.0112	98.27±1.37	91.74±1.34	91.82±1.29
	TMTE	Task-aware Co-evolution	–	–	–	–	0.7626±0.0108	95.73±1.36	84.90±1.18	86.01±1.24
	DMGraph		**0.9085**±0.0150	**100.00**±0.00	**100.00**±0.00	**100.00**±0.00	**0.7980**±0.0126	**99.18**±0.55	**94.25**±1.35	**94.37**±1.42

language representations. Because models differ in architecture, pretraining, and PEFT method, this comparison measures backbone compatibility rather than an isolated parameter-scaling effect.

F Expanded-Neighborhood and Noise Results

Tables 6 and 7 report the complete pruning-regime results behind Figure 2, including secondary metrics, retained-edge ratios, inference time, and peak memory. Increasing the candidate cap from

Table 5: Backbone scalability (generative ROUGE-L; discriminative R-L). Gray values denote standard deviations.

Backbone	Wiki.	FB15K	Ele-F.	Good.
OPT-125M	0.4119±0.0064	0.5863±0.0096	0.8491±0.0143	0.7389±0.0115
LLaMA-1B	0.4819±0.0077	0.6619±0.0109	0.9085±0.0150	0.7980±0.0126
Qwen2.5-VL-7B	**0.5101**±0.0068	**0.6869**±0.0081	**0.9218**±0.0144	**0.8123**±0.0130

Table 6: Expanded-neighborhood pruning results on WikiWeb2M with LLaMA-1B. Retained edges are measured relative to the candidate set.

N_{cap}	Method	K_s	ROUGE-L ↑	CIDEr ↑	Retained (%) ↓	s/batch ↓	Memory (GB) ↓
32	Graph4MM	NA	0.4746±0.0069	1.1282±0.0168	NA	0.0882±0.0005	19.8±0.4
	Dynamic Dense	all	0.4598±0.0076	1.0872±0.0175	100	0.0915±0.0005	20.3±0.4
	Hard Top-K	5	0.4522±0.0083	1.0585±0.0186	26.1	0.0838±0.0004	19.4±0.3
	Hard Top-K	10	0.4615±0.0078	1.0838±0.0177	52.1	0.0845±0.0004	19.5±0.3
	Hard Top-K	15	0.4671±0.0075	1.1042±0.0172	78.2	0.0852±0.0005	19.6±0.4
	DMGraph	5	0.4642±0.0080	1.0918±0.0179	26.1	0.0848±0.0004	19.9±0.4
	DMGraph	10	0.4755±0.0074	1.1172±0.0170	52.1	0.0855±0.0005	20.1±0.4
	DMGraph	15	0.4812±0.0072	1.1338±0.0168	78.2	0.0862±0.0005	20.2±0.4
	DMGraph	all	0.4605±0.0075	1.0895±0.0174	100	0.0915±0.0005	20.3±0.4
64	Graph4MM	NA	0.4728±0.0072	1.1218±0.0172	NA	0.1085±0.0006	24.2±0.5
	Dynamic Dense	all	0.4548±0.0079	1.0762±0.0181	100	0.1142±0.0006	25.0±0.5
	Hard Top-K	5	0.4495±0.0087	1.0512±0.0192	15.9	0.0998±0.0005	23.5±0.4
	Hard Top-K	10	0.4588±0.0081	1.0778±0.0180	31.7	0.1012±0.0005	23.7±0.5
	Hard Top-K	15	0.4645±0.0077	1.0968±0.0174	47.6	0.1028±0.0006	23.9±0.5
	DMGraph	5	0.4618±0.0083	1.0845±0.0183	15.9	0.1015±0.0005	24.1±0.5
	DMGraph	10	0.4735±0.0076	1.1118±0.0172	31.7	0.1032±0.0006	24.4±0.5
	DMGraph	15	0.4795±0.0073	1.1298±0.0169	47.6	0.1048±0.0006	24.6±0.5
	DMGraph	all	0.4558±0.0078	1.0788±0.0179	100	0.1142±0.0006	25.0±0.5

32 to 64 makes a fixed K_s substantially sparser: $K_s = 15$ retains 78.2% of candidate edges at cap 32 and 47.6% at cap 64. The performance maximum remains at $K_s = 15$ on both datasets, while inference is faster than the all-edge setting.

Table 7: Expanded-neighborhood pruning results on Goodreads-LP with LLaMA-1B. Retained edges are measured relative to the candidate set.

N_{cap}	Method	K_s	R-L ↑	AUC ↑	Retained (%) ↓	s/batch ↓	Memory (GB) ↓
32	Graph4MM	NA	0.7755±0.0108	0.9782±0.0031	NA	0.0865±0.0005	19.5±0.4
	Dynamic Dense	all	0.7682±0.0112	0.9745±0.0034	100	0.0898±0.0005	20.0±0.4
	Hard Top-K	5	0.7592±0.0119	0.9695±0.0038	26.1	0.0822±0.0004	19.1±0.3
	Hard Top-K	10	0.7695±0.0111	0.9752±0.0033	52.1	0.0828±0.0004	19.2±0.4
	Hard Top-K	15	0.7738±0.0108	0.9772±0.0032	78.2	0.0835±0.0005	19.3±0.4
	DMGraph	5	0.7708±0.0113	0.9758±0.0034	26.1	0.0832±0.0004	19.6±0.4
	DMGraph	10	0.7802±0.0108	0.9812±0.0030	52.1	0.0838±0.0005	19.8±0.4
	DMGraph	15	0.7955±0.0105	0.9842±0.0028	78.2	0.0845±0.0005	19.9±0.4
	DMGraph	all	0.7692±0.0111	0.9750±0.0033	100	0.0898±0.0005	20.0±0.4
64	Graph4MM	NA	0.7738±0.0110	0.9775±0.0032	NA	0.1062±0.0006	23.8±0.5
	Dynamic Dense	all	0.7648±0.0115	0.9722±0.0035	100	0.1118±0.0006	24.5±0.5
	Hard Top-K	5	0.7558±0.0121	0.9678±0.0039	15.9	0.0978±0.0005	23.0±0.4
	Hard Top-K	10	0.7672±0.0113	0.9742±0.0034	31.7	0.0992±0.0005	23.2±0.5
	Hard Top-K	15	0.7722±0.0110	0.9765±0.0033	47.6	0.1008±0.0006	23.4±0.5
	DMGraph	5	0.7685±0.0115	0.9748±0.0034	15.9	0.0995±0.0005	23.6±0.5
	DMGraph	10	0.7788±0.0110	0.9802±0.0031	31.7	0.1012±0.0006	23.9±0.5
	DMGraph	15	0.7935±0.0106	0.9838±0.0029	47.6	0.1028±0.0006	24.1±0.5
	DMGraph	all	0.7658±0.0114	0.9728±0.0035	100	0.1118±0.0006	24.5±0.5

Tables 8 and 9 give the complete corruption results. Figure 4 isolates the mechanism by plotting the fraction of injected edges that survive the gate. DMGraph consistently retains fewer corrupted edges than hard Top-K; its learned scores therefore improve selection within the same cardinality budget. Dynamic Dense retains every injected edge by construction.

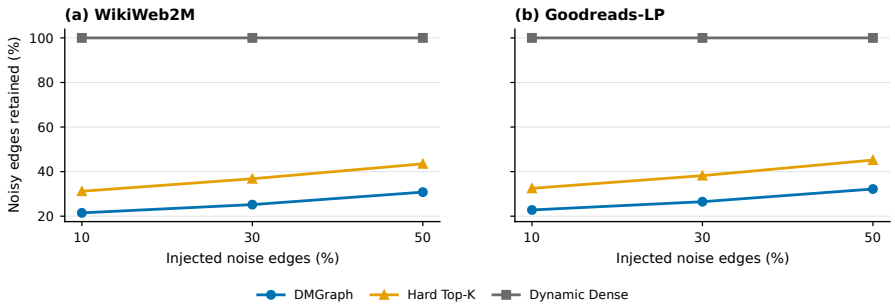

Figure 4: **Retention of injected noise edges.** The fraction of added random non-edges retained at $K_s = 15$ for 64-node candidate neighborhoods. DMGraph uses the same edge budget as hard Top-K but removes a larger share of corrupted candidates on both datasets.

Table 8: Test-time edge-corruption results on WikiWeb2M with $N_{\text{cap}} = 64$ and $K_s = 15$ for sparse methods.

Noise (%)	Method	K_s	ROUGE-L ↑	Drop (%) ↓	Noisy retained (%) ↓	Noisy/Clean weight ↓
0	Graph4MM	NA	0.4728±0.0072	0	NA	NA
	Dynamic Dense	all	0.4548±0.0079	0	NA	NA
	Hard Top-K	15	0.4645±0.0077	0	NA	NA
	DMGraph	15	0.4795±0.0073	0	NA	NA
10	Graph4MM	NA	0.4425±0.0078	6.4	NA	NA
	Dynamic Dense	all	0.4238±0.0085	6.8	100	0.882
	Hard Top-K	15	0.4408±0.0082	5.1	31.2	0.425
	DMGraph	15	0.4642±0.0076	3.2	21.5	0.348
30	Graph4MM	NA	0.4052±0.0085	14.3	NA	NA
	Dynamic Dense	all	0.3785±0.0092	16.8	100	0.845
	Hard Top-K	15	0.4082±0.0088	12.1	36.8	0.458
	DMGraph	15	0.4385±0.0080	8.6	25.2	0.372
50	Graph4MM	NA	0.3568±0.0092	24.5	NA	NA
	Dynamic Dense	all	0.3152±0.0098	30.7	100	0.802
	Hard Top-K	15	0.3628±0.0095	21.9	43.5	0.492
	DMGraph	15	0.4058±0.0088	15.4	30.8	0.405

G Implementation Details

G.1 Training and Preprocessing Details

Subgraph extraction. In WikiWeb2M, where edges and images are sparse, we follow MMGL and select 10 textual neighbors and 4 image neighbors in addition to the target node. In Ele-Fashion, we keep all first-hop neighbors and randomly sample two second-hop neighbors, capping text and visual subgraphs at 11 nodes.

Expanded-neighborhood evaluation. For WikiWeb2M and Goodreads-LP, we additionally cap the candidate neighborhood at 32 or 64 nodes and evaluate $K_s \in \{5, 10, 15, \text{all}\}$ with LLaMA-1B. DMGraph and hard Top-K use identical candidate sets and cardinality budgets. Dynamic Dense scores every candidate without pruning, while Graph4MM follows its native graph propagation. For robustness, we reuse models trained on clean graphs at $N_{\text{cap}} = 64$ and add random non-edges at test time. The number of injected edges equals 10%, 30%, or 50% of the original candidate-edge count. Each method receives the same corrupted graph for a given seed.

Input length. For single-node input, we use 512 tokens for generative tasks and 128 tokens for discriminative tasks. For subgraph input, we use 1024 tokens for generation and 512 tokens for discrimination.

Parameter-efficient tuning. We use 20 prefix tokens for OPT-125M and LoRA rank 64 for LLaMA-1B / Qwen2.5-VL-7B. Within each backbone, DMGraph and all trainable PLM-based baselines use the same PEFT method and budget. The vision encoder is CLIP. Experiments were run on NVIDIA H20-NVLink GPUs.

Table 9: Test-time edge-corruption results on Goodreads-LP with $N_{\text{cap}} = 64$ and $K_s = 15$ for sparse methods.

Noise (%)	Method	K_s	R-L ↑	Drop (%) ↓	Noisy retained (%) ↓	Noisy/Clean weight ↓
0	Graph4MM	NA	0.7738±0.0110	0	NA	NA
	Dynamic Dense	all	0.7648±0.0115	0	NA	NA
	Hard Top-K	15	0.7722±0.0110	0	NA	NA
	DMGraph	15	0.7935±0.0106	0	NA	NA
10	Graph4MM	NA	0.7225±0.0118	6.6	NA	NA
	Dynamic Dense	all	0.7085±0.0122	7.4	100	0.875
	Hard Top-K	15	0.7312±0.0116	5.3	32.5	0.438
	DMGraph	15	0.7682±0.0110	3.2	22.8	0.355
30	Graph4MM	NA	0.6582±0.0128	14.9	NA	NA
	Dynamic Dense	all	0.6298±0.0135	17.6	100	0.838
	Hard Top-K	15	0.6725±0.0125	12.9	38.2	0.468
	DMGraph	15	0.7252±0.0115	8.6	26.5	0.382
50	Graph4MM	NA	0.5825±0.0138	24.7	NA	NA
	Dynamic Dense	all	0.5452±0.0145	28.7	100	0.795
	Hard Top-K	15	0.6012±0.0132	22.1	45.2	0.502
	DMGraph	15	0.6725±0.0122	15.2	32.2	0.415

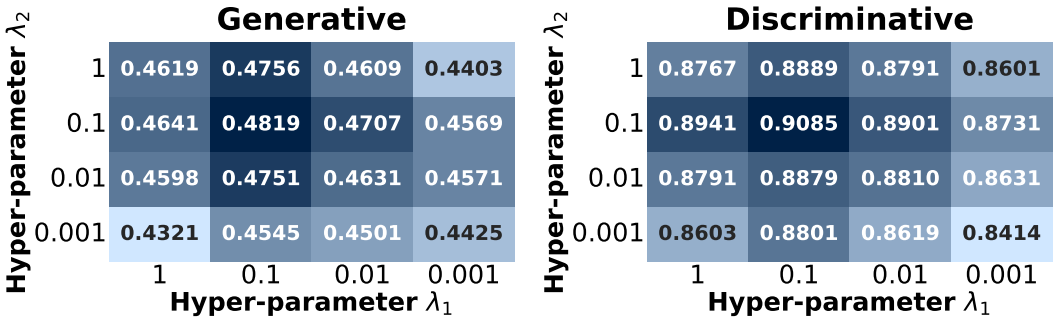

Figure 5: λ_1, λ_2 sensitivity (ROUGE-L / R-L, LLaMA-1B). WikiWeb2M and Ele-Fashion are used as representative generative and discriminative tasks.

Training and seeds. We use learning rate 1×10^{-4}, effective batch size 64, $K_s = 15$, $K_d = 2$, $\lambda_1 = \lambda_2 = 0.1$, $\tau_* = 0.2$, $\epsilon_0 = 1.0$, $\epsilon_{\min} = 0.01$, and $\gamma = 0.98$. For labeled tasks, the semantic collision prior is estimated on the training split as $\widehat{\tau}^+ = \sum_c n_c(n_c-1)/(n(n-1))$; for tasks without class labels, τ^+ is selected on validation data. Learned-model results are point means over three seeds. We will release per-seed logs, sampled IDs, and cached subgraph splits.

G.2 Hyper-parameter Sensitivity

Figures 5 and 6 report sensitivity to the main loss and structure hyperparameters on representative generative and discriminative tasks. The shared configuration uses $\lambda_1 = \lambda_2 = 0.1$, $K_s = 15$, and $K_d = 2$.

The two loss weights have different effects. A very small λ_1 weakens cross-modal alignment, while a large value over-constrains task representations. The effect of λ_2 is smoother because the structural regularizer anchors the dynamic adjacency. For the structure parameters, $K_d = 2$ balances neighborhood propagation and over-smoothing. Under the default 11-node cap, $K_s = 15$ retains the candidate set; Section 5.3 separately evaluates K_s as an active pruning budget in 32- and 64-node neighborhoods.

G.3 Additional Ablation Details

Table 11 complements the main ablation in Figure 3 by isolating smaller design choices on WikiWeb2M with LLaMA-1B. The results are organized around three questions: whether Hetero-Align needs both modality-specific projections and debiased negatives, whether the dynamic edge module benefits from differentiable reweighting, and whether alignment is useful beyond the node level.

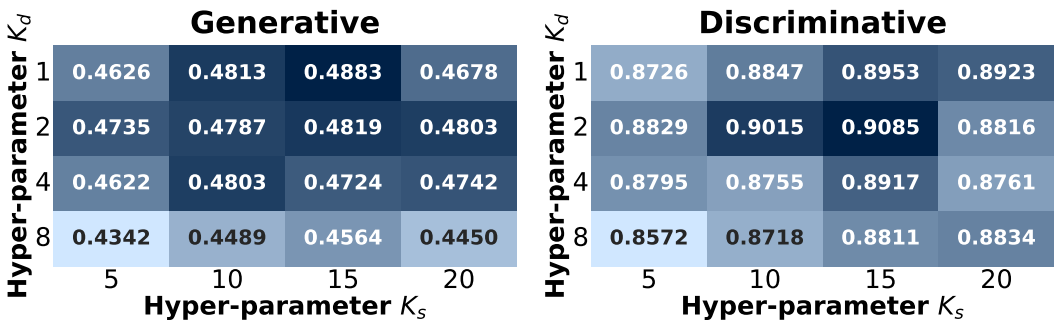

Figure 6: K_s, K_d sensitivity (ROUGE-L / R-L, LLaMA-1B). WikiWeb2M and Ele-Fashion are used as representative generative and discriminative tasks.

Table 10: Hyperparameter settings for generative and discriminative tasks.

Hyperparameter	Generative	Discriminative
Learning Rate	1×10^{-4}	1×10^{-4}
Max Input Length	1024	512
Max Output Length	128	32
Text Neighbor Sampling	11	11
Image Neighbor Sampling	5	11
Batch Size	4	4
Gradient Accumulation Steps	16	16
Visual Tokens per Image	4	4
LoRA Rank	64	64
Prefix Tuning Virtual Tokens	20	20
Decay Rate γ	0.98	0.98
Differentiable Top-K Sparsification K_s	15	15
Number of QFormer Block	1	1
Number of Diffusion Hop K_d	2	2
All temperature parameter τ_*	0.2	0.2
Weight parameter λ_1	0.1	0.1
Weight parameter λ_2	0.1	0.1
Training Epochs (OPT-125M)	50	5
Training Epochs (LLaMA-1B)	3	3

Hetero-Align. Removing modality-specific projections weakens performance because text and vision embeddings enter DMGraph with different geometry and noise profiles. Removing debiased negatives causes a larger degradation, indicating that graph-neighbor contrastive learning is exposed to structural false negatives. The combined removal is not simply additive, which suggests that the two mechanisms interact, but both are necessary for stable cross-modal alignment.

Differentiable edge gate. Dense attention keeps all candidate neighbors and can pass noisy or weakly related context into the backbone. Hard thresholding is worse because it removes low-scoring edges without a useful training signal for the thresholded scores. Gumbel-Softmax top-K is competitive, but the straight-through differentiable gate used by DMGraph gives the strongest result in this comparison. These comparisons use the reweighting configuration specified in Section 5.

Hierarchy of alignment losses. Node-level alignment alone is insufficient because it aligns local text-image pairs without explicitly constraining subgraph semantics or batch-global context. Removing either subgraph-level or batch-global alignment reduces performance, and using only the node-level term gives the weakest variant among the hierarchy settings. Each coarser loss contributes alongside node-level alignment in these comparisons.

G.4 Edge Reweighting Interpretability

We analyze how learned edge mass is distributed across target-centered candidate neighborhoods, relating the reweighted graph to the content of its text and visual neighbors.

Figure 7 combines two case studies with an aggregate analysis. Node labels are short anonymized descriptors derived from dataset records, and edge widths denote learned dynamic weights. In the WikiWeb2M example, the target theatre page assigns stronger weights to content-matched text and visual neighbors, including jazz, Broadway, and theatre-related poster or image nodes. Weakly

Table 11: Fine-grained ablations on WikiWeb2M with LLaMA-1B (R-L).

Hetero-Align		Sparsification		Alignment Level	
Full	**0.4819**	Diff. Top-K (ours)	**0.4819**	Full hierarchy	**0.4819**
w/o Hetero-Align	0.4591	Gumbel-Softmax Top-K	0.4791	w/o Subgraph-Align	0.4713
w/o Mod.-Spec. Proj.	0.4701	Dense attention	0.4572	w/o Global-Align	0.4735
w/o Debiased Neg.	0.4653	Hard thresholding	0.4487	Node-level only	0.4603

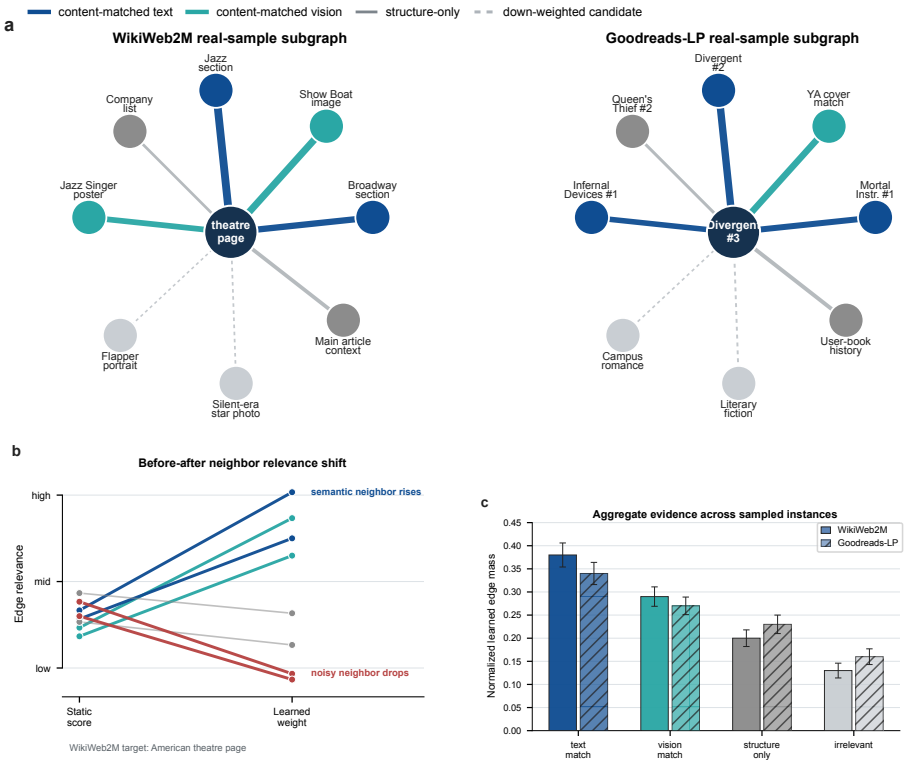

Figure 7: **Interpretability of content-conditioned edge reweighting.** (a) Representative target-centered subgraphs from WikiWeb2M and Goodreads-LP. Edge width denotes the learned dynamic weight. Blue and cyan edges indicate content-matched text and vision neighbors, while gray dashed edges indicate down-weighted static candidates. (b) Before-after rank shifts for the WikiWeb2M case. Content-matched neighbors move upward after dynamic reweighting, whereas noisy candidates move downward. (c) Normalized learned edge mass across sampled target instances. Bars show means with standard deviations over target nodes.

specific structural or visual candidates receive smaller weights. The Goodreads-LP example shows the same behavior for a book-series target: series-related and young-adult fiction neighbors dominate the learned neighborhood, while less aligned candidates are down-weighted.

The rank-shift panel gives a finer view of the WikiWeb2M case. Content-matched text and vision neighbors move from moderate static scores to high learned weights, whereas noisy candidates move downward. The learned weights thus reshape the effective neighborhood according to target-specific multimodal content.

The aggregate panel summarizes learned weights across sampled targets from WikiWeb2M and Goodreads-LP: content-matched text neighbors receive the largest normalized edge mass, followed by content-matched visual neighbors. Structure-only neighbors remain moderate, and irrelevant candidates receive the smallest mass. This ordering supports the interpretation that DMGraph concentrates message passing on content-relevant neighbors under the current bounded-subgraph setting.

Table 13: Performance in discriminative task with OPT-125M backbone across different numbers of unseen classes.

# Unseen	R-L	Acc%	Rec%	Pre%
1	0.9013	100.00	100.00	100.00
3	0.8873	100.00	100.00	100.00
5	0.8741	100.00	100.00	100.00
7	0.8502	100.00	100.00	100.00
9	0.8336	100.00	97.38	97.38

H Efficiency and Robustness

Table 12 reports per-batch training and generation time on OPT-125M. The comparison isolates the cost of content-conditioned edge scoring and hierarchical alignment under the same parameter-efficient tuning setup used by the PLM-based baselines. DMGraph is slightly slower than MMGL and faster than Graph4MM in this setting.

Table 12: Per-batch time on OPT-125M.

Models	**Training** (s/batch)	**Generation** (s/batch)
MMGL	0.2105±0.1215	0.0595±0.0004
Graph4MM	0.2419±0.1394	0.0745±0.0004
DMGraph	0.2319±0.1244	0.0699±0.0004

The reason is architectural. DMGraph does not run dense message passing over the full graph during each update. Instead, dynamic edge scoring, diffusion, and alignment are applied inside capped induced subgraphs, with $n_m \leq 11$ in the reported experiments. The extra parameters are concentrated in lightweight projection, scoring, and alignment layers rather than in a separate large encoder. As a result, the measured overhead reflects bounded local computation rather than graph-size-dependent full-batch propagation.

The Ele-Fashion subset experiment provides a practical scaling check. Per-batch time grows approximately linearly from 10K to 97K nodes and stays close to the trend observed for Graph4MM. These measurements characterize dataset scaling with the induced-subgraph cap held fixed. Increasing that cap changes the local attention and diffusion costs given in Appendix B.3.

I Different Numbers of Unseen Classes

Table 13 varies the number of held-out classes in the Ele-Fashion discriminative task with OPT-125M. This experiment checks whether the zero-shot split difficulty changes the conclusion drawn from the saturated categorical metrics in the main results.

The results show a consistent pattern: categorical accuracy, recall, and precision remain high across several split settings, while R-L is more sensitive to changes in the held-out class set. This confirms that Ele-Fashion is useful for testing robustness to unseen product categories, but its categorical metrics alone are not fine-grained enough to separate strong methods once the label space becomes easy to match. For this reason, the main paper treats Ele-Fashion as supporting evidence and places more weight on ROUGE-L, Goodreads-LP, and the generative benchmarks when comparing graph-multimodal reasoning quality.