NO-FREE-GRAPH: LEARNING WHEN MULTIMODAL DATA SHOULD BE GRAPHIFIED

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal graph learning has recently emerged as an effective paradigm for incorporating inter-entity relationships into multimodal representations. Existing studies have made substantial progress on *how to construct and optimize graphs*, but rarely consider a more fundamental question: *whether additional relational structures should be introduced for a given dataset and task*. Through empirical studies across diverse datasets, tasks, and graph constructors, we reveal that graphification is not consistently beneficial: introducing relational structures can provide substantial improvements in some cases, while offering limited or even negative gains. This observation motivates a new perspective that graph construction should be treated as a selective decision based on its expected utility rather than a default preprocessing step. To address this issue, we propose MAG-SCOUT, a pre-construction graph assessment framework that estimates whether introducing graph structures is beneficial before generating the complete topology. MAG-SCOUT collects limited relational evidence, analyzes its potential task-specific contribution, and estimates the expected utility of graphification together with construction cost to make a build-or-skip decision. Extensive experiments across six multimodal datasets, three downstream tasks, and diverse graph constructors demonstrate that MAG-SCOUT effectively identifies when graph structures should be introduced, avoiding **23.6%** of graph construction work while retaining **96.7%** of held-out positive-gain mass under the pre-registered floor.

1 INTRODUCTION

Multimodal data provide rich entity-level information from multiple modalities, such as textual descriptions, visual contents, and other semantic attributes. Recently, multimodal graph learning has emerged as an effective paradigm to further incorporate relationships among different entities, enabling relational reasoning beyond individual multimodal representations in applications such as recommendation, retrieval, and multimodal understanding (Yan et al., 2025). By constructing connections based on semantic similarity, interactions, or domain knowledge, multimodal graphs allow models to exploit complementary information from both entity attributes and external relations. However, unlike multimodal features that are usually directly available, these inter-entity relations often need to be additionally constructed before graph-based learning. Existing studies have therefore focused on designing better graph construction strategies and graph structure learning methods to discover useful relationships and optimize graph topology for downstream tasks (Franceschi et al., 2019; Chen et al., 2020; Jin et al., 2020).

However, an earlier question remains largely unexplored: *should additional relational structures be introduced for the current multimodal data at all?* Existing multimodal graph learning pipelines implicitly assume that enriching entity representations with graph structures will provide additional benefits once a graph can be constructed. This assumption, however, does not always hold. Although relational information can complement individual multimodal features, constructed graphs may also introduce irrelevant connections, propagate noisy signals, or provide limited additional information when existing multimodal representations already capture sufficient task-related semantics. To investigate this issue, we conduct an empirical study across diverse datasets, downstream tasks, and graph construction strategies. We find that graphification exhibits highly different outcomes under different settings: some cases obtain substantial improvements over raw multimodal predictors, whereas others achieve marginal gains or even performance degradation. These observations

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.

reveal that graph construction is not an inherently beneficial operation, but a task- and construction-dependent decision whose utility should be assessed before graphification.

This problem is not directly addressed by existing graph learning methods. Graph construction and graph structure learning approaches mainly focus on discovering meaningful relations, refining graph topology, or jointly optimizing graph structures with downstream objectives (Franceschi et al., 2019; Chen et al., 2020; Jin et al., 2020). Meanwhile, graph learning and model optimization methods improve prediction performance after graph structures have already been obtained (Park et al., 2022; 2023). Although these methods substantially advance how graphs are constructed and exploited, they assume that graphification has already been selected as an appropriate representation choice. Therefore, they cannot answer the earlier question of whether constructing additional relational structures is worthwhile for a specific dataset, task, and construction strategy. A straightforward solution is to construct the graph first and evaluate its downstream performance afterwards, but this process is inefficient because the graph construction and subsequent graph learning costs have already been incurred.

Motivated by this observation, we propose MAG-SCOUT, a pre-construction framework that estimates whether introducing graph structures is beneficial before generating the complete topology. MAG-SCOUT first performs **Candidate Relation Sampling** to collect limited relational evidence without constructing the full graph. It then conducts **Graph Utility Analysis** to assess whether the sampled relations can provide complementary information beyond existing multimodal representations. Finally, **Cost-aware Utility Prediction** estimates the expected benefit of graph construction and balances the potential gain with construction cost to make the final build-or-skip decision. In this way, MAG-SCOUT complements existing graph construction methods by determining when relational structures should be introduced rather than how they should be constructed.

Our Contributions. ***(1) New Perspective.*** We identify an overlooked decision problem in multimodal graph learning: whether graph construction is beneficial before the graph is built, and show that graphification does not always provide additional benefit. ***(2) New Framework.*** We propose MAG-SCOUT, which evaluates graph construction from limited relational evidence and predicts whether a graph is expected to improve downstream performance before full graph construction. ***(3) New Evaluation.*** We establish a decision-oriented evaluation protocol across six multimodal datasets, three downstream tasks, and diverse graph constructors, demonstrating that MAG-SCOUT avoids unnecessary graph construction work by **23.6%** while retaining **96.7%** of the held-out positive-gain mass and making reliable build-or-skip decisions.

2 PROBLEM FORMULATION

We formulate graphification as a pre-construction decision problem. Given raw multimodal data, a downstream task, and a candidate graph constructor, we aim to determine whether introducing relational structures can provide additional utility beyond raw multimodal representations. Let

$$\mathcal{D} = (\mathcal{V}, \{\mathcal{X}^{(m)}\}_{m\in\mathcal{M}}), \quad \mathcal{G}_{\mathcal{C}} = \mathcal{C}(\mathcal{D}) = (\mathcal{V}, \mathcal{E}_{\mathcal{C}}, \mathbf{A}_{\mathcal{C}}), \tag{1}$$

where $\mathcal{V}$ is the entity set, $\mathcal{X}^{(m)}$ denotes modality m, and $\mathcal{E}_{\mathcal{C}}$ denotes constructed relations. Different constructors may induce different graphs from the same data; therefore, $\mathcal{C}$ is treated as part of the graphification condition. For a downstream task $\mathcal{T}$, the utilities of raw multimodal learning and graph-based learning are defined as:

$$Y_{\text{raw}} = \mathcal{Q}(F_{\text{raw}}(\mathcal{D}), \mathcal{T}), \quad Y_{\text{graph}} = \mathcal{Q}(F_{\text{graph}}(\mathcal{G}_{\mathcal{C}}, \mathcal{D}), \mathcal{T}). \tag{2}$$

The graphification value is measured by:

$$V^* = Y_{\text{graph}} - Y_{\text{raw}}, \quad U^* = V^* - \lambda_c C(\mathcal{D}, \mathcal{C}), \tag{3}$$

where $C(\mathcal{D}, \mathcal{C})$ denotes the graphification cost. However, U^* can only be obtained after graph construction and downstream learning. Before construction, only bounded relational evidence $\mathcal{S}_B$ is available. Therefore, MAG-SCOUT aims to learn:

$$f_\theta(\mathcal{D}, \mathcal{T}, \mathcal{C}, \mathcal{S}_B) \to \widehat{U}, \tag{4}$$

which estimates graphification utility before materializing the graph. Unlike graph structure learning methods that optimize an existing topology, MAG-SCOUT addresses the earlier question of whether graphification should be invoked.

3 EMPIRICAL STUDY

Before presenting MAG-SCOUT, we revisit the assumption behind existing multimodal graph learning pipelines: whether constructing a graph is consistently beneficial compared with learning directly from raw multimodal observations. We measure the relative benefit using $\Delta = Y_{\text{graph}} - Y_{\text{raw}}$, referred to as *GraphGain*, where positive and negative values indicate whether graphification improves or degrades performance. Table 1 reports the comparison across diverse graphification settings. The results reveal that graphification is not universally beneficial: the average GraphGain varies from -2.91 to $+3.72$ across different cases, showing that graph structures can provide substantial improvements in some scenarios while introducing limited or even negative effects in others. Moreover, graphification utility varies significantly with different construction strategies, indicating that its effectiveness depends on the interaction among multimodal data, downstream tasks, and graph constructors rather than the dataset alone or the final graph-model performance. Therefore, neither always constructing graphs nor always relying on raw multimodal representations is optimal. This **No-Free-Graph** phenomenon motivates MAG-SCOUT, which aims to estimate whether graphification is worthwhile before paying the cost of graph construction and downstream graph learning. Detailed experimental settings and additional empirical analyses are provided in Appendix C.

Table 1: GraphGain relative to RawOnly across six multimodal benchmarks, in percentage points ($\Delta \equiv 0$). Cells report five-seed means with standard deviations. All nine constructors are included. The Average row is their unweighted mean on each dataset.

Constructor	**Node classification (Acc)**		**Link prediction (AUC)**		**Cross-modal retrieval (R@5)**	
	Toys	**Grocery**	**Bili Music**	**DY**	**QB**	**Bili Cartoon**
kNN-GCN	+0.58$_{\pm 0.31}$	+2.25$_{\pm 0.30}$	-4.17$_{\pm 1.34}$	-3.77$_{\pm 0.61}$	-3.65$_{\pm 2.03}$	-0.08$_{\pm 2.54}$
MM-kNN-GCN	+0.92$_{\pm 0.36}$	+1.93$_{\pm 0.08}$	-3.89$_{\pm 1.27}$	-2.39$_{\pm 0.49}$	+0.33$_{\pm 0.82}$	-0.34$_{\pm 1.34}$
AutoG	+4.26$_{\pm 0.19}$	+2.92$_{\pm 0.16}$	-0.87$_{\pm 0.49}$	-3.80$_{\pm 0.62}$	-2.41$_{\pm 1.14}$	+1.56$_{\pm 0.96}$
AutoGraph	+7.27$_{\pm 0.60}$	+6.79$_{\pm 0.60}$	+2.02$_{\pm 0.49}$	-2.13$_{\pm 1.41}$	-3.26$_{\pm 0.70}$	-0.34$_{\pm 1.37}$
auGraph	+5.11$_{\pm 0.29}$	+4.97$_{\pm 0.27}$	+1.23$_{\pm 0.92}$	-2.81$_{\pm 1.08}$	-4.31$_{\pm 1.83}$	-0.30$_{\pm 1.26}$
LLaTA	+3.14$_{\pm 0.33}$	+3.68$_{\pm 0.26}$	-2.16$_{\pm 0.84}$	-2.64$_{\pm 0.81}$	-2.15$_{\pm 1.28}$	-0.17$_{\pm 1.42}$
GAugLLM	+1.87$_{\pm 0.28}$	+2.45$_{\pm 0.19}$	-3.45$_{\pm 1.12}$	-3.51$_{\pm 0.58}$	-3.72$_{\pm 1.54}$	-0.41$_{\pm 1.25}$
LAGA	+2.63$_{\pm 0.35}$	+3.12$_{\pm 0.22}$	+0.73$_{\pm 0.68}$	-2.88$_{\pm 0.76}$	-2.83$_{\pm 0.96}$	+0.58$_{\pm 1.08}$
GraphMaster	+4.78$_{\pm 0.42}$	+5.36$_{\pm 0.37}$	+1.58$_{\pm 0.53}$	-2.27$_{\pm 1.05}$	-0.56$_{\pm 1.14}$	+1.04$_{\pm 0.91}$
Average	**+3.40**	**+3.72**	**-1.00**	**-2.91**	**-2.51**	**+0.17**

Positive cells are gains over RawOnly and negative cells are losses. The Average row equals the unweighted mean of the nine constructors above.

4 RELATED WORK

Graph Construction and Structure Learning. Graph structure learning aims to discover or refine relational structures when explicit topology is unavailable. Early methods such as LDS (Franceschi et al., 2019), IDGL (Chen et al., 2020), and Pro-GNN (Jin et al., 2020) jointly optimize graph structures and downstream objectives, while later approaches such as SUBLIME (Liu et al., 2022) and NodeFormer (Wu et al., 2022) improve scalable and self-supervised graph induction. Recent multimodal graph methods, including multimodal graph benchmarks (Yan et al., 2025), further incorporate multimodal information into graph representations. These methods mainly focus on improving *how to construct graphs* once graph-based learning is considered beneficial, but do not examine whether graphification itself is worthwhile.

Graph Utility Assessment and Selection. Another line of work studies the usefulness and optimization of available graphs. Graph property analyses investigate factors such as homophily and neighborhood consistency (Zhu et al., 2020; Pei et al., 2020), while recent studies explore when graph structures benefit prediction, including *When Do GNNs Help* (Luan et al., 2023) and NetInfoF (Lee et al., 2024). Automated graph learning methods further address graph model selection

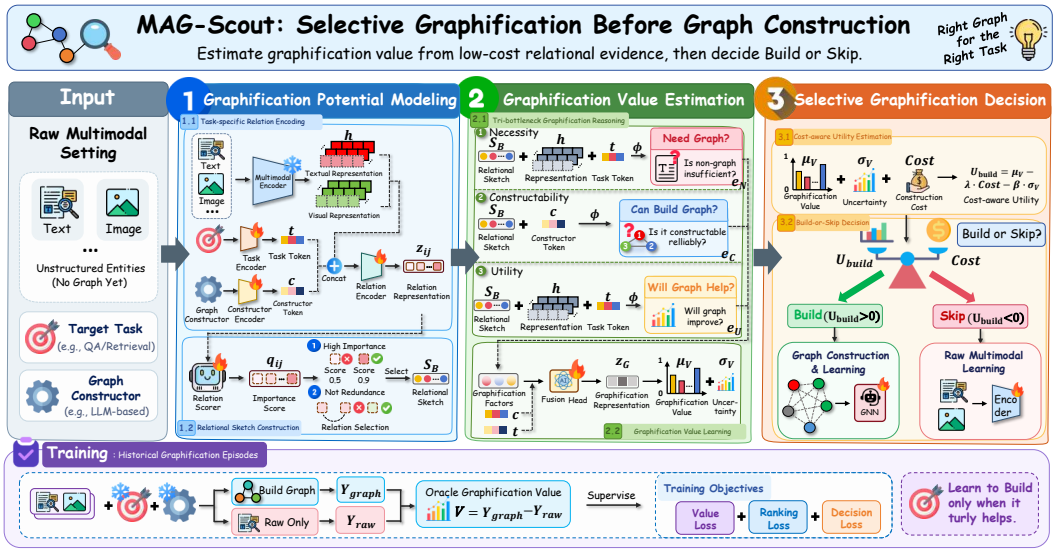


Figure 1: Overview of MAG-SCOUT: (1) Relational encoding on $\mathcal{P}$ yields compact sketch $\mathcal{S}_B$; (2) Tri-bottleneck reasoning $(\mathbf{e}_N, \mathbf{e}_C, \mathbf{e}_U)$ estimates value μ_V and scale σ_V; (3) Cost-aware gating uses calibrated scale $\tilde{\sigma}_V$ to render Build/Skip decisions.

through architecture search and performance prediction, including GraphNAS (Gao et al., 2020), Auto-GNN (Zhou et al., 2022), AutoGL (Zhang et al., 2021), MetaGL (Park et al., 2022), and GLEMOS (Park et al., 2023). However, these approaches assume that graph structures have already been constructed and focus on improving graph-based learning rather than determining whether graph construction should be performed.

5 METHODOLOGY

5.1 OVERVIEW

We propose MAG-SCOUT, a pre-construction framework that estimates the value of multimodal graphification before explicitly constructing the graph. Given multimodal data $\mathcal{D}$, downstream task $\mathcal{T}$, and graph constructor $\mathcal{C}$, MAG-SCOUT learns whether introducing relational structures can provide sufficient benefit over raw multimodal representations under the current graphification condition. It first extracts low-cost relational evidence, then estimates graphification value by analyzing whether relations are necessary, constructable, and useful, and finally makes a cost-aware build-or-skip decision. It consists of three modules: **Graphification Potential Modeling**, **Graphification Value Estimation**, and **Selective Graphification Decision**. Figure 1 illustrates the overall architecture.

5.2 GRAPHIFICATION POTENTIAL MODELING

Motivation. Graphification depends on relational patterns that are unavailable before construction. We extract a compact sketch from a bounded candidate pool to assess these patterns without materializing the complete graph.

Task-conditioned Relational Representation. Given multimodal data $\mathcal{D}$, we obtain entity representations:

$$\mathbf{H} = E_{\text{mm}}(\mathcal{D}) = \{\mathbf{h}_i\}_{i=1}^{N}, \tag{5}$$

where $E_{\text{mm}}(\cdot)$ denotes a multimodal encoder. The task and graph constructor are encoded as $\mathbf{t} = E_{\text{task}}(\mathcal{T})$ and $\mathbf{c} = E_{\text{con}}(\mathcal{C})$. Instead of constructing all possible relations, we first obtain a bounded candidate relation pool $\mathcal{P}$ through inexpensive relation discovery. For each candidate relation $(i, j) \in \mathcal{P}$, we learn a task-conditioned relational representation:

$$\mathbf{z}_{ij} = \phi_\theta(\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_i \odot \mathbf{h}_j, |\mathbf{h}_i - \mathbf{h}_j|, \mathbf{t}, \mathbf{c}), \tag{6}$$

where $\mathbf{z}_{ij}$ describes the potential contribution of this relation to graphification assessment rather than predicting a final graph edge.

Relational Sketch Construction. Since graphification value depends on the collective properties of relations, we select a compact relational sketch by combining relevance with a positive-semidefinite diversity kernel. Let

$$\begin{aligned} q_e &= \mathrm{softplus}(\mathbf{u}^\top \mathbf{z}_e), \qquad \mathbf{v}_e = q_e \frac{\mathbf{z}_e}{\max(\|\mathbf{z}_e\|_2, \epsilon_z)}, \\ f(\mathcal{S}) &= \log\det\left(\mathbf{I}_{d_z} + \sum_{e\in\mathcal{S}} \mathbf{v}_e \mathbf{v}_e^\top\right), \qquad \mathcal{S}^\star \in \arg\max_{\mathcal{S}\subseteq\mathcal{P},\,|\mathcal{S}|\le B} f(\mathcal{S}). \end{aligned} \tag{7}$$

Here $e = (i, j)$ indexes candidate relations, $\mathbf{u}$ is learned, and $\epsilon_z > 0$ guards zero embeddings. Greedy selection returns $\mathcal{S}_B$ with $f(\mathcal{S}_B) \ge (1 - 1/e) f(\mathcal{S}^\star)$ for fixed features and a cardinality budget; Appendix B.2 gives the exact marginal and proof.

5.3 Graphification Value Estimation

Motivation. A sketch reveals potential relations, not their downstream benefit. We organize value estimation around three diagnostic questions: whether raw features lack task-relevant information, whether reliable relations can be formed, and whether propagation sharpens the task-relevant signal.

Tri-Bottleneck Graphification Reasoning. Given the sketch $\mathcal{S}_B$, we extract three latent states:

$$[\mathbf{e}_N, \mathbf{e}_C, \mathbf{e}_U] = \Phi(\mathcal{S}_B, \mathbf{H}, \mathbf{t}, \mathbf{c}), \tag{8}$$

where $\mathbf{e}_N$, $\mathbf{e}_C$, and $\mathbf{e}_U$ characterize whether a graph is needed, can be constructed, and is expected to help, respectively. Specifically, $\mathbf{e}_N$ captures the insufficiency of raw multimodal representations for the target task, $\mathbf{e}_C$ reflects the reliability and coverage of candidate relations, and $\mathbf{e}_U$ describes the potential benefit of relational reasoning. The three bottlenecks are further integrated under the current task and construction condition:

$$\mathbf{z}_G = f_\psi([\alpha_N \mathbf{e}_N \| \alpha_C \mathbf{e}_C \| \alpha_U \mathbf{e}_U \| \mathbf{t} \| \mathbf{c}]), \quad \alpha_k = \sigma(\mathbf{w}_k^\top [\mathbf{e}_k \| \mathbf{t} \| \mathbf{c}]). \tag{9}$$

Based on the resulting graphification representation, we estimate the expected benefit and its uncertainty:

$$(\mu_V, s_V) = F_\omega(\mathbf{z}_G), \quad \sigma_V = \mathrm{softplus}(s_V) + \epsilon_\sigma, \quad \tilde{\sigma}_V = a_{\mathrm{cal}} \sigma_V. \tag{10}$$

Here F_ω predicts the mean and raw scale; σ_V and $\tilde{\sigma}_V$ have GraphGain units. The positive calibration factor a_{cal} is fitted on held-out calibration sources (Appendix B.4).

Value Learning Objective. During training, historical graphification episodes provide oracle values:

$$V^* = Y_{\mathrm{graph}} - Y_{\mathrm{raw}}, \tag{11}$$

where Y_{graph} and Y_{raw} denote downstream performance with and without graph construction. We optimize the value estimator with:

$$\mathcal{L} = \mathcal{L}_{\mathrm{gain}} + \lambda_{\mathrm{rank}} \mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{unc}} \mathcal{L}_{\mathrm{unc}} + \lambda_{\mathrm{dec}} \mathcal{L}_{\mathrm{dec}}, \tag{12}$$

where the ranking term preserves relative graphification utility, the uncertainty term trains the predicted spread, and the decision term aligns value estimation with build-or-skip decisions. The uncertainty likelihood and pairwise ranking term are defined explicitly in Appendix B.4.

5.4 SELECTIVE GRAPHIFICATION DECISION

Motivation. Graphification value alone does not account for execution cost or prediction uncertainty. The decision combines both with the estimated gain.

Cost-aware Utility Optimization. Given the estimated value distribution, we formulate the utility of graph construction as:

$$U_{\text{build}} = \mu_V - \lambda_c \widehat{C}_{\text{pre}}(\mathcal{D}, \mathcal{C}) - \lambda_u \tilde{\sigma}_V, \tag{13}$$

Here $\widehat{C}_{\text{pre}}$ is predicted from pre-construction features; measured costs supervise it offline and score decisions afterward. λ_c converts cost into gain units; $\lambda_u \geq 0$ weights calibrated uncertainty.

Build-or-Skip Decision. The final action is obtained by comparing graph construction with retaining the raw multimodal representation:

$$a^* = \mathbb{I}[U_{\text{build}} \geq \tau_{\text{thr}}], \tag{14}$$

The threshold τ_{thr} is selected on inner policy-validation sources after calibration is frozen. Action $a^* = 1$ builds the graph; $a^* = 0$ retains the raw representation. Equation (14) therefore requires neither the target graph nor its realized gain at inference.

6 EXPERIMENTS

We study four aspects of selective graphification. Q1 measures constructor-level gains and tests whether they can be predicted before graph construction. Q2 examines the effect of removing predictor components. Q3 evaluates Build/Skip policies at a 95% retained-positive-gain target. Q4 tests sensitivity to hyperparameter changes and corrupted relational evidence.

6.1 EXPERIMENTAL SETUP

The constructor-level comparison uses six OpenMAG benchmarks (Wan et al., 2026): *Toys* and *Grocery* for node classification, *Bili Music* and *DY* for link prediction, and *QB* and *Bili Cartoon* for cross-modal retrieval. These are the six benchmarks; node classification, link prediction, and retrieval are the three task families. We compare RawOnly with nine graph constructors: **kNN-GCN**, **MM-kNN-GCN**, **AutoG**, **AutoGraph**, **auGraph**, **LLaTA**, **GAugLLM**, **LAGA**, and **GraphMaster**. The predictor comparison includes **MAG-SCOUT**, **NetInfoF** (Lee et al., 2024), **WDGH** (Luan et al., 2023), **GLEMOS-S2** (Park et al., 2023), and **MetaGL** (Park et al., 2022). Q1a uses the six benchmarks; Q1b, Q3, and Q4 use the 13-source registry; Q2 uses a separate, rebuilt 11-source bank. We report Accuracy, AUC, R@5@100, signed GraphGain, rank correlations, gain MAE, retained positive-gain mass, and saved graph work. Appendix C gives the data splits, seed protocols, and aggregation rules.

6.2 IS GRAPH CONSTRUCTION ALWAYS BENEFICIAL? (Q1)

Q1a measures each constructor's GraphGain relative to RawOnly on the six benchmarks. Q1b tests source-held-out prediction on the 13-source registry. Table 1 gives the benchmark-level gains; Figure 2 and Table 2 summarize predictor quality.

The nine-constructor mean ranges from $\mathbf{-2.91}$ points on DY to $\mathbf{+3.72}$ on Grocery. AutoGraph gains $\mathbf{+7.27}$ points on Toys; auGraph loses $\mathbf{4.31}$ on QB. On QB, eight constructors have negative mean gain, and all nine do on DY. Q1b asks whether information available before full graph construction predicts such differences. On the 13 parent sources, NetInfoF and WDGH reach link-prediction rank correlations of $\rho = \mathbf{0.43}$ and $\rho = \mathbf{0.41}$. MAG-SCOUT reaches $\rho = \mathbf{0.28}$ for cross-modal retrieval and a node-classification gain MAE of $\mathbf{1.63}$ points. All predictors receive raw multimodal features, task and constructor identifiers, and bounded relational evidence. Appendices C.1, C.3, and C.4 describe the data and models.

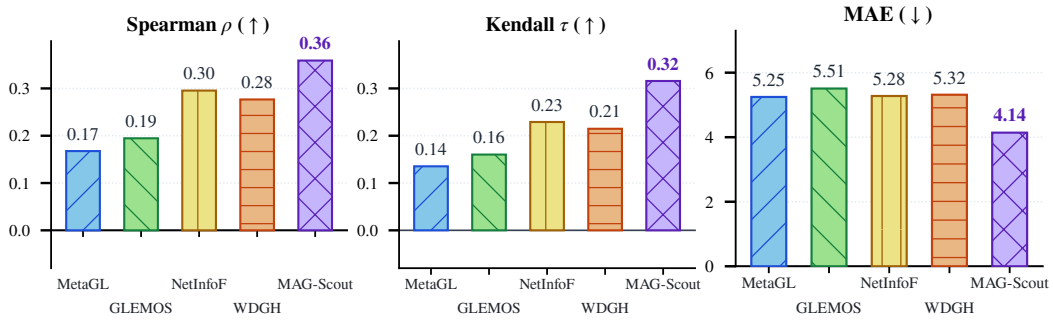


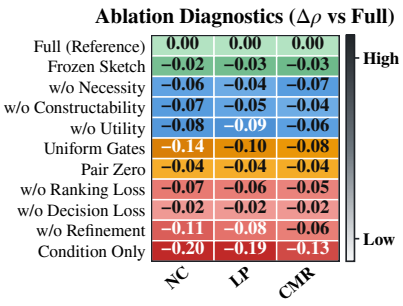
Figure 2: Macro prediction quality across five methods and three task families on the 13 parent sources. Panels report Spearman correlation, Kendall correlation, and gain MAE.

Table 2: Pre-construction prediction quality across the 13 parent sources. Cells report task-family means with seed-level standard deviations; bold and underline indicate the best and second-best values.

Predictor	Spearman rank correlation ($\rho \uparrow$)			Kendall rank correlation ($\tau \uparrow$)			Gain error (MAE $\downarrow$)		
	NC	LP	CMR	NC	LP	CMR	NC	LP	CMR
MAG-Scout	**0.417**$_{\pm 0.05}$	0.383$_{\pm 0.06}$	**0.276**$_{\pm 0.06}$	**0.362**$_{\pm 0.03}$	**0.331**$_{\pm 0.03}$	**0.253**$_{\pm 0.08}$	**1.63**$_{\pm 0.17}$	**6.83**$_{\pm 0.45}$	**3.97**$_{\pm 0.29}$
NetInfoF	0.358$_{\pm 0.11}$	**0.429**$_{\pm 0.08}$	0.099$_{\pm 0.13}$	0.318$_{\pm 0.07}$	0.295$_{\pm 0.06}$	0.074$_{\pm 0.06}$	2.14$_{\pm 0.24}$	8.48$_{\pm 1.53}$	5.22$_{\pm 0.53}$
WDGH	0.336$_{\pm 0.12}$	0.414$_{\pm 0.07}$	0.079$_{\pm 0.12}$	0.294$_{\pm 0.11}$	0.282$_{\pm 0.06}$	0.068$_{\pm 0.08}$	2.39$_{\pm 0.22}$	8.74$_{\pm 1.36}$	4.82$_{\pm 0.79}$
GLEMOS-S2	0.182$_{\pm 0.08}$	0.342$_{\pm 0.11}$	0.059$_{\pm 0.08}$	0.155$_{\pm 0.08}$	0.279$_{\pm 0.08}$	0.046$_{\pm 0.06}$	2.89$_{\pm 0.28}$	7.94$_{\pm 0.93}$	5.70$_{\pm 0.72}$
MetaGL	0.167$_{\pm 0.09}$	0.298$_{\pm 0.10}$	0.037$_{\pm 0.12}$	0.132$_{\pm 0.11}$	0.246$_{\pm 0.10}$	0.027$_{\pm 0.09}$	3.45$_{\pm 0.59}$	7.54$_{\pm 1.18}$	4.76$_{\pm 0.67}$

6.3 Which Components Enable Reliable Assessment? (Q2)

Q2 tests component contributions on a rebuilt 11-source bank, separate from the Q1 and Q3 registry. All variants use the same outer folds, three model seeds, and optimization budget. For node classification, Spearman ρ is $\mathbf{0.42}$ for the full model, $\mathbf{0.28}$ with uniform gates, and $\mathbf{0.31}$ without refinement; the frozen-sketch control reaches $\mathbf{0.40}$. For link prediction, the corresponding values are $\mathbf{0.38}$ for the full model, $\mathbf{0.29}$ without the utility branch, and $\mathbf{0.35}$ for the frozen sketch. Removing the utility branch and replacing learned gates account for the largest ranking drops. Across the ten variants, cross-modal retrieval correlations are at least $\mathbf{0.19}$. Table 3 gives the absolute results; Appendix D.1 gives the differences from the full model.



The Full row serves as the reference. Subsequent rows isolate the three diagnostic signals ($\mathbf{e}_N, \mathbf{e}_C, \mathbf{e}_U$), followed by fusion, interaction, objective, refinement, and frozen-sketch variants.

Figure 3: Matched ablation results relative to Full MAG-Scout on the rebuilt source bank. Colors encode the direction and magnitude of Spearman differences.

Table 3: Matched component-ablation results on the separate rebuilt source bank across node classification, link prediction, and cross-modal retrieval. Cells report means with seed-level standard deviations; bold marks the best value in each column.

Ablation Variant	Spearman ρ (↑)			Kendall τ (↑)			Gain error (MAE ↓)		
	NC	LP	CMR	NC	LP	CMR	NC	LP	CMR
Full MAG-Scout (Reference)	**0.418**$_{\pm 0.058}$	**0.379**$_{\pm 0.047}$	**0.276**$_{\pm 0.071}$	**0.361**$_{\pm 0.034}$	**0.324**$_{\pm 0.041}$	**0.238**$_{\pm 0.052}$	**1.716**$_{\pm 0.112}$	**6.748**$_{\pm 0.418}$	**3.964**$_{\pm 0.346}$
Frozen Heuristic Sketch	0.396$_{\pm 0.041}$	0.351$_{\pm 0.053}$	0.241$_{\pm 0.036}$	0.342$_{\pm 0.039}$	0.297$_{\pm 0.046}$	0.209$_{\pm 0.033}$	1.893$_{\pm 0.168}$	7.164$_{\pm 0.514}$	4.218$_{\pm 0.431}$
w/o Necessity ($\mathbf{e}_N$)	0.361$_{\pm 0.064}$	0.341$_{\pm 0.049}$	0.208$_{\pm 0.043}$	0.312$_{\pm 0.058}$	0.286$_{\pm 0.051}$	0.174$_{\pm 0.064}$	2.163$_{\pm 0.246}$	7.514$_{\pm 0.581}$	4.581$_{\pm 0.374}$
w/o Constructability ($\mathbf{e}_C$)	0.347$_{\pm 0.057}$	0.328$_{\pm 0.038}$	0.239$_{\pm 0.082}$	0.301$_{\pm 0.043}$	0.274$_{\pm 0.067}$	0.206$_{\pm 0.072}$	2.087$_{\pm 0.138}$	7.628$_{\pm 0.894}$	4.443$_{\pm 0.476}$
w/o Utility ($\mathbf{e}_U$)	0.339$_{\pm 0.046}$	0.286$_{\pm 0.054}$	0.214$_{\pm 0.051}$	0.287$_{\pm 0.031}$	0.241$_{\pm 0.063}$	0.183$_{\pm 0.038}$	2.458$_{\pm 0.184}$	8.247$_{\pm 0.512}$	4.916$_{\pm 0.467}$
Uniform Bottleneck Gates	0.281$_{\pm 0.052}$	0.274$_{\pm 0.044}$	0.191$_{\pm 0.037}$	0.246$_{\pm 0.054}$	0.229$_{\pm 0.069}$	0.167$_{\pm 0.047}$	2.341$_{\pm 0.171}$	7.906$_{\pm 0.687}$	4.729$_{\pm 0.385}$
w/o Pairwise Interaction	0.381$_{\pm 0.079}$	0.342$_{\pm 0.036}$	0.237$_{\pm 0.062}$	0.328$_{\pm 0.028}$	0.291$_{\pm 0.042}$	0.204$_{\pm 0.045}$	1.924$_{\pm 0.119}$	7.237$_{\pm 0.642}$	4.306$_{\pm 0.428}$
w/o Ranking Loss ($\mathcal{L}_{\text{rank}}$)	0.352$_{\pm 0.083}$	0.318$_{\pm 0.048}$	0.224$_{\pm 0.074}$	0.298$_{\pm 0.057}$	0.271$_{\pm 0.049}$	0.193$_{\pm 0.053}$	1.847$_{\pm 0.127}$	7.086$_{\pm 0.763}$	4.132$_{\pm 0.407}$
w/o Decision Loss ($\mathcal{L}_{\text{dec}}$)	0.401$_{\pm 0.056}$	0.362$_{\pm 0.034}$	0.258$_{\pm 0.067}$	0.347$_{\pm 0.066}$	0.309$_{\pm 0.037}$	0.223$_{\pm 0.048}$	2.274$_{\pm 0.263}$	7.853$_{\pm 0.719}$	4.637$_{\pm 0.412}$
w/o One-time Refinement	0.312$_{\pm 0.061}$	0.301$_{\pm 0.042}$	0.217$_{\pm 0.058}$	0.268$_{\pm 0.051}$	0.256$_{\pm 0.044}$	0.184$_{\pm 0.062}$	2.118$_{\pm 0.154}$	7.392$_{\pm 0.527}$	4.417$_{\pm 0.483}$

6.4 Can MAG-Scout Make Cost-effective Build/Skip Decisions? (Q3)

Q3 selects one threshold per predictor on inner policy-validation sources. The threshold maximizes task-macro saved graph work subject to a task-macro retained positive-gain mass of at least 95%. Outer-test sources are scored once at that frozen threshold (Appendix C.2). Retention averages source-level positive-gain ratios within each task, then averages the three task families. Saved graph work is computed from measured graphification costs and is the fraction of cost avoided relative to Always-Build; decisions themselves use the pre-construction cost estimate.

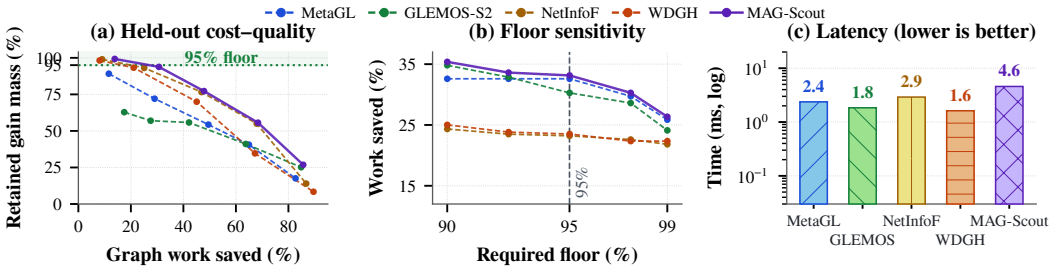


Figure 4: Build/Skip comparison across five predictors: threshold trade-offs, retained-gain floors, and warm decision latency.

Figure 4 shows the held-out policy results; Appendix D.2 describes the operating-point metrics and Appendix D.3 reports the floor sweep. At the validation-selected 95% target, MAG-Scout retains $\mathbf{96.7}\%$ of held-out positive-gain mass, saves $\mathbf{23.6}\%$ of graph work, and builds on $\mathbf{76.7}\%$ of cases. NetInfoF is the only other learned predictor to exceed 95% held-out retention. The paired source-bootstrap point estimate of their saved-work difference is $\mathbf{0.5}$ points (95% CI $[-6.7,\ 7.8]$), so the observed difference is uncertain; this paired estimate need not equal the subtraction of the rounded headline values. Their wrong-skip rates are 13.3% for MAG-Scout and 9.3% for NetInfoF. WDGH's wrong-skip rate is 10.1%, with 90.5% retained mass. MetaGL and GLEMOS-S2 save more work, but retain only 80.2% and 60.5%; their wrong-skip rates are 27.7% and 25.5%. The additional savings therefore come with lower retained mass and more positive-gain cases skipped on the held-out sources.

6.5 Is the Policy Robust to Hyperparameter and Evidence Perturbations? (Q4)

Q4 uses two protocols. The sensitivity grid fixes the decision threshold at zero and varies one hyperparameter at a time. In the corruption test, we freeze the predictor and threshold selected on clean validation data, perturb the relational evidence, and measure the resulting policy changes.

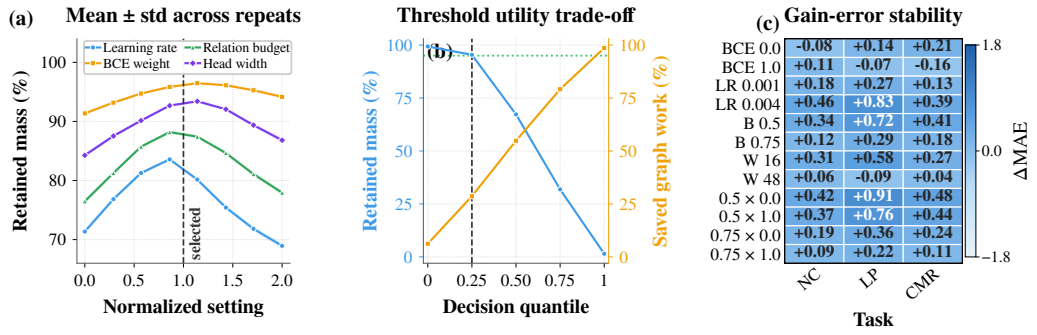


Figure 5: One-factor retained mass at a zero threshold, the threshold trade-off, and gain error under input perturbations.

Table 4: Zero-threshold sensitivity (left) and frozen-policy evidence corruption (right) on 13 parent sources. Each factor has its own reference. Best (Spread) is the grid maximum and its difference from that reference.

Dimension	Pts	$t = 0$ Ref.	$t = 0$ Best (Spread)	Perturbation	ΔRet.	ΔBuild	Flips
Learning Rate	3	81.9	83.6 (1.7)	Rewiring	−0.4	−7.1	2/13
BCE Loss Weight	3	96.2	96.5 (0.3)	Relation Collapse	−0.8	−7.1	2/13
Relation Budget	3	87.8	88.2 (0.4)	Single-Modal Dominant	−1.6	−14.3	3/13
Utility Head Width	3	93.0	93.4 (0.4)	Relation-Bank Partition	−0.1	−0.2	0/13

Figure 5 and Table 4 show the two sets of results. The zero-threshold reference retention values for learning rate, BCE weight, relation budget, and utility-head width are 81.9%, 96.2%, 87.8%, and 93.0%, respectively; the largest one-factor increase is 1.7 points. Under evidence corruption, retained mass decreases by at most 1.6 points, while the build rate decreases by as much as 14.3 points. At most $3/13$ parent-source decisions change; relation-bank partition changes none.

7 Conclusion

In this work, we revisit a fundamental yet underexplored question in multimodal graph learning: whether a graph should be constructed at all. Our empirical study reveals a **No-Free-Graph** phenomenon, showing that graphification can bring substantial gains in some cases while providing limited or even negative benefits in others. Motivated by this observation, we propose MAG-Scout, a pre-construction framework that estimates graphification value before the complete topology is materialized. MAG-Scout first extracts a compact relational sketch, then evaluates graphification from the perspectives of necessity, constructability, and utility, and finally combines the predicted value with construction cost and uncertainty to make a selective Build-or-Skip decision. Extensive experiments across diverse multimodal datasets, tasks, and graph constructors demonstrate that MAG-Scout can reliably identify when graphification is beneficial while avoiding unnecessary graph construction. We hope this study encourages multimodal graph learning to move beyond the assumption of always constructing graphs toward more selective and value-aware graphification.

Reproducibility and AI Use Statements

An anonymized implementation, configurations, and reproducibility checks are available at `https://anonymous.4open.science/r/MAG-Scout-Review-2DC3/`. The main paper details the diagnostic protocols, the MAG-Scout sketch, tri-bottleneck parameterization, joint optimization, and selective decision policy; the appendix adds theoretical and robustness analyses. The code omits raw data and pretrained checkpoints but documents data format, dependencies, and reproduction instructions.

Generative AI tools assisted with language refinement, literature discovery, and research ideation. The authors reviewed and edited the assisted material and take responsibility for the manuscript.

REFERENCES

Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. doi: 10.1016/S0022-0000(03)00025-4.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pp. 380–388, 2002. doi: 10.1145/509907.509965.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024a.

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*, 2020.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (LLMs) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61, 2024b. doi: 10.1145/3655103.3655110.

Zhikai Chen, Han Xie, Jian Zhang, Xiang Song, Jiliang Tang, Huzefa Rangwala, and George Karypis. AutoG: Towards automatic graph construction from tabular data. *arXiv preprint arXiv:2501.15282*, 2025.

Tamara Cucumides and Floris Geerts. From features to structure: Task-aware graph construction for relational and tabular learning with GNNs. *arXiv preprint arXiv:2506.02243*, 2025.

Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. GraphMaster: Automated graph synthesis via LLM agents in data-limited environments. In *Advances in Neural Information Processing Systems*, volume 38, pp. 114736–114783, 2025. doi: 10.52202/085713-3462.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. GAugLLM: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 747–758, 2024. doi: 10.1145/3637528.3672035.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*, 2019.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.

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

Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. Graph neural architecture search. In *International Joint Conference on Artificial Intelligence*, pp. 1403–1409, 2020.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, pp. 2151–2159, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

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.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8622–8642, 2024. doi: 10.1109/TKDE.2024.3469578.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 66–74, 2020. doi: 10.1145/3394486.3403049.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1–2):81–93, 1938. doi: 10.2307/2332226.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012. doi: 10.1561/2200000044.

Meng-Chieh Lee, Haiyang Yu, Jian Zhang, Vassilis N. Ioannidis, Xiang Song, Soji Adeshina, Da Zheng, and Christos Faloutsos. NetInfoF framework: Measuring and exploiting network usable information. In *International Conference on Learning Representations*, 2024.

Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference*, pp. 1392–1403, 2022. doi: 10.1145/3485447.3512186.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28748–28760, 2023. doi: 10.52202/075280-1251.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Empirical Methods in Natural Language Processing*, pp. 188–197, 2019.

Namyong Park, Ryan A. Rossi, Nesreen K. Ahmed, and Christos Faloutsos. MetaGL: Evaluation-free selection of graph learning models via meta-learning. *arXiv preprint arXiv:2206.09280*, 2022.

Namyong Park, Ryan A. Rossi, Xing Wang, Antoine Simoulin, Nesreen K. Ahmed, and Christos Faloutsos. GLEMOS: Benchmark for instantaneous graph learning model selection. In *Advances in Neural Information Processing Systems 36*, pp. 69887–69899, 2023.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh V. Chawla, and Chao Huang. A survey of large language models for graphs. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6616–6626, 2024. doi: 10.1145/3637528.3671460.

Rong Shan, Jianghao Lin, Chenxu Zhu, Bo Chen, Menghui Zhu, Kangning Zhang, Jieming Zhu, Ruiming Tang, Yong Yu, and Weinan Zhang. An automatic graph construction framework based on large language models for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4806–4817, 2025. doi: 10.1145/3711896.3737192.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 (3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

Chenxi Wan, Xunkai Li, Yilong Zuo, Haokun Deng, Sihan Li, Bowen Fan, Hongchao Qin, Ronghua Li, and Guoren Wang. OpenMAG: A comprehensive benchmark for multimodal-attributed graph. *arXiv preprint arXiv:2602.05576*, 2026.

Shiyu Wang, Gang Zhou, Mingjing Lan, Jicang Lu, Zhufeng Li, and Yi Xia. LAGA: A graph adapter for long-tail text classification via semantic space refinement. *Journal of King Saud University Computer and Information Sciences*, 2026. doi: 10.1007/s44443-026-00656-z.

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 35*, pp. 27387–27401, 2022.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

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*, pp. 5842–5853, 2025. doi: 10.1145/3711896.3737404.

Jiaqi Zhang, Yu Cheng, Yongxin Ni, Yunzhu Pan, Zheng Yuan, Junchen Fu, Youhua Li, Jie Wang, and Fajie Yuan. Ninerec: A benchmark dataset suite for evaluating transferable recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5256–5267, 2025a. doi: 10.1109/TPAMI.2024.3373868.

Zhihan Zhang, Xunkai Li, Lei Zhu, Guang Zeng, Bowen Fan, Yanzhe Wen, Hongchao Qin, Rong-Hua Li, and Guoren Wang. Unlocking graph structure learning with tree-guided large language models. *arXiv preprint arXiv:2503.21223*, 2025b.

Ziwei Zhang, Yijian Qin, Zeyang Zhang, Chaoyu Guan, Jie Cai, Heng Chang, Jiyan Jiang, Haoyang Li, Zixin Sun, Beini Xie, Yang Yao, Yipeng Zhang, Xin Wang, and Wenwu Zhu. AutoGL: A library for automated graph learning. *arXiv preprint arXiv:2104.04987*, 2021.

Tong Zhao, Yozen Liu, Leonardo Neves, Oliver J. Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11015–11023, 2021. doi: 10.1609/aaai.v35i12.17315.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. doi: 10.1016/j.aiopen.2021.01.001.

Kaixiong Zhou, Xiao Huang, Qingquan Song, Rui Chen, and Xia Hu. Auto-gnn: Neural architecture search of graph neural networks. *Frontiers in Big Data*, 5, 2022. doi: 10.3389/fdata.2022.1029307.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, 2020.

Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Yuanqi Du, Jieyu Zhang, Qiang Liu, Carl Yang, and Shu Wu. A survey on graph structure learning: Progress and opportunities. *arXiv preprint arXiv:2103.03036*, 2021.

A Reproducibility, Software Environment, and Hardware Infrastructure

A.1 Computing Infrastructure

The pilot evaluations ran on Linux computing servers with the following configuration. **Processors.** Dual-socket Intel(R) Xeon(R) Gold 6240 CPUs @ 2.60 GHz (18 physical cores per socket, 36 physical cores total, supporting 72 execution threads). **System Memory.** 256 GB DDR4-2933 ECC Registered RAM (with over 210 GB dynamically available during peak distributed evaluations). **Accelerators.** 4× NVIDIA A100 Tensor Core GPUs with 40 GB HBM2 memory each (interconnected via PCIe Gen4 with peer-to-peer access). **Software Stack.** Ubuntu 22.04 LTS (kernel 5.15.0-107-generic), Python 3.10.12, PyTorch 2.4.0 with CUDA 12.1 runtime and cuDNN 8.9.2, PyTorch Geometric (Fey & Lenssen, 2019) (`torch_geometric`) 2.5.3, NetworkX 3.2.1, Scikit-learn 1.4.2, SciPy 1.13.0, and Matplotlib 3.8.4.

A.2 Seed Protocol and Scientific Determinism

Q1a, Q3, and Q4 use five matched random seeds across datasets and constructors:

$$\mathcal{S}_{\text{seed}} = \{344701,\ 675949,\ 86547,\ 931443,\ 798884\}. \tag{15}$$

For each episode, all PyTorch, NumPy, Python standard library, and CUDA random number generator states are explicitly initialized with the corresponding seed. All bootstrap estimations (e.g., 10,000 paired bootstrap draws for saved work and retained gain mass confidence intervals) are conducted under fixed random generator states (seed `20260916`). The current Q1b pilot package contains five runs per source with source-matched seed sets; ten sources use the set above and three historical sources use a source-specific five-seed set. Q2 uses three model seeds $\{17, 23, 42\}$ on its separate 11-source bank. These seed scopes are reported with their respective tables.

A.3 Code and Asset Availability

The accompanying paper archive contains the LaTeX sources, figure-generation scripts, and tabulated inputs for the reported plots.

B Extended Methodology and Mathematical Details

Appendix B gives implementation details for the method in Section 5, including regularized log-determinant selection, the three diagnostic heads, the four training terms, scale calibration, and threshold selection. Fitting, calibration, policy validation, and outer testing use separate source groups; Appendix B.4 specifies their data flow and the training objective.

B.1 Notation

B.2 Candidate Relation Probing and Sketch Selection Algorithm

Section 5 defines the compact sketch $\mathcal{S}_B$ selected from candidate pool $\mathcal{P}$. Algorithm 1 summarizes candidate discovery and sketch extraction.

Low-dimensional candidate discovery. The first stage performs approximate neighbour search in a randomly projected space. Writing $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{d' \times d}$ for the projection matrix with $d' \ll d$, the projected representation is

$$\tilde{\mathbf{h}}_i = \frac{\mathbf{W}_{\text{proj}}\mathbf{h}_i}{\max(\|\mathbf{W}_{\text{proj}}\mathbf{h}_i\|_2, \epsilon_z)} \in \mathbb{R}^{d'}, \tag{16}$$

and cosine distance is evaluated on $\tilde{\mathbf{h}}_i$. Random projection reduces the search dimension (Achlioptas, 2003), and angular search admits locality-sensitive hashing (Charikar, 2002). Approximation can change neighbour order; the selection guarantee below is relative to the resulting candidate pool, not to all $N(N-1)/2$ pairs.

Pairwise relational representation. For each candidate pair $(i, j) \in \mathcal{P}$ we form a feature by concatenating the two endpoint embeddings, their Hadamard product, their absolute difference, and

Table 5: Notation used throughout the methodology. Dimensions are for a single source; the encoder width is d and the reduced projection width is $d' \ll d$.

Symbol	Meaning
$\mathcal{D} = (\mathcal{V}, \{\mathcal{X}^{(m)}\}_{m\in\mathcal{M}})$	multimodal dataset: entity set $\mathcal{V}$ (cardinality $N = \lvert\mathcal{V}\rvert$) with modality tensors $\mathcal{X}^{(m)}$
$\mathcal{M}$	set of observed modalities (text, visual, and auxiliary attributes)
$\mathcal{T}, \mathcal{C}$	downstream task and candidate graph constructor
$\mathcal{C}(\mathcal{D}) = (\mathcal{V}, \mathcal{E}_\mathcal{C}, \mathbf{A}_\mathcal{C})$	graph induced by constructor $\mathcal{C}$
$\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^N \in \mathbb{R}^{N\times d}$	frozen multimodal entity representations (E_{mm})
$\mathbf{h}_i^{(m)}$	modality-m representation of entity i
$\mathbf{t} = E_{\text{task}}(\mathcal{T})$, $\mathbf{c} = E_{\text{con}}(\mathcal{C})$	task and constructor conditioning tokens in $\mathbb{R}^d$
$\mathcal{P}$, K	bounded candidate relation pool ($\lvert\mathcal{P}\rvert \le K$) and its budget
$\mathbf{z}_{ij} \in \mathbb{R}^{d_z}$	task-conditioned representation of candidate relation (i, j)
$q_{ij} = \text{softplus}(\mathbf{u}^\top \mathbf{z}_{ij})$	positive relevance quality of relation (i, j)
$\mathbf{v}_e, \mathbf{L}, \mathbf{M}_\mathcal{S}$	quality-weighted relation vector, Gram kernel, and $\mathbf{I} + \sum_{e\in\mathcal{S}} \mathbf{v}_e\mathbf{v}_e^\top$
$\mathcal{S}_B \subseteq \mathcal{P}$, B	greedily selected relation indices and budget; embeddings are attached to these indices
$\epsilon_z, \epsilon_\sigma, \epsilon_a$	positive guards for embedding norms, predicted scales, and scale calibration
$\mathbf{e}_N, \mathbf{e}_C, \mathbf{e}_U \in \mathbb{R}^{d_e}$	necessity, constructability, and utility latent states; d_e is their width
α_k	learned gate weight for branch $k \in \{N, C, U\}$
$\mathbf{z}_G$	fused graphification representation
μ_V, s_V, σ_V	predicted mean, unconstrained scale, and positive Gaussian working-model scale
$a_{\text{cal}}, \tilde{\sigma}_V$	positive calibration multiplier and calibrated scale $a_{\text{cal}}\sigma_V$
$V^* = Y_{\text{graph}} - Y_{\text{raw}}$	realized (oracle) graphification value in percentage points
$\widehat{C}_{\text{pre}}$, $C(\mathcal{D}, \mathcal{C})$	pre-construction cost estimate and measured post-hoc graphification cost
U_{build}	cost- and uncertainty-adjusted utility of building
$\lambda_{\text{rank}}, \lambda_{\text{unc}}, \lambda_{\text{dec}}, \lambda_{\text{c}}, \lambda_u$	loss and utility balancing weights
π, τ_{thr}	Build/Skip policy and its decision threshold
$\tau_{\text{rank}}, T_{\text{dec}}$	positive ranking and decision temperatures in gain units

the two conditioning tokens,

$$\mathbf{z}_{ij} = \phi_\theta\big([\,\mathbf{h}_i \,\|\, \mathbf{h}_j \,\|\, \mathbf{h}_i \odot \mathbf{h}_j \,\|\, |\mathbf{h}_i - \mathbf{h}_j| \,\|\, \mathbf{t} \,\|\, \mathbf{c}\,]\big), \tag{17}$$

where ϕ_θ is a shared MLP with parameters θ, $\|$ denotes concatenation, and $\odot$ the element-wise product. The input dimension is $4d + 2d = 6d$ and the output dimension is d_z. The Hadamard and difference terms expose multiplicative and contrastive interactions between the endpoints, while appending $\mathbf{t}$ and $\mathbf{c}$ conditions the representation on the current task and constructor so that the same pair may be scored differently under different graphification conditions.

Regularized volume selection. For $e = (i, j) \in \mathcal{P}$, let $q_e = \text{softplus}(\mathbf{u}^\top \mathbf{z}_e)$ and $\mathbf{v}_e = q_e\mathbf{z}_e / \max(\|\mathbf{z}_e\|_2, \epsilon_z)$, where $\mathbf{u} \in \mathbb{R}^{d_z}$ is learned and $\epsilon_z > 0$. Stacking the vectors as rows of $\mathbf{V}$ gives a PSD quality–diversity kernel $\mathbf{L} = \mathbf{V}\mathbf{V}^\top$. The determinant identity yields

$$\begin{aligned} \mathbf{M}_\mathcal{S} &= \mathbf{I}_{d_z} + \sum_{e\in\mathcal{S}} \mathbf{v}_e\mathbf{v}_e^\top, \\ f(\mathcal{S}) &= \log\det \mathbf{M}_\mathcal{S} = \log\det(\mathbf{I}_{|\mathcal{S}|} + \mathbf{L}_\mathcal{S}). \end{aligned} \tag{18}$$

This regularized volume criterion is inspired by determinant-based diversity models (Kulesza & Taskar, 2012), but is distinct from unregularized DPP MAP selection of $\log\det \mathbf{L}_\mathcal{S}$. The identity term defines $f(\varnothing) = 0$ and makes every $\mathbf{M}_\mathcal{S}$ positive definite, including for duplicate or zero relation vectors. Equation (7) uses the same objective; $\mathcal{S}^\star$ denotes its optimum and $\mathcal{S}_B$ the greedy approximation returned by Algorithm 1.

Exact incremental rule. The matrix determinant lemma gives

$$\Delta(e \mid \mathcal{S}) = f(\mathcal{S} \cup \{e\}) - f(\mathcal{S}) = \log\big(1 + \mathbf{v}_e^\top \mathbf{M}_\mathcal{S}^{-1}\mathbf{v}_e\big)\,. \tag{19}$$

For the empty set this equals $\log(1 + \|\mathbf{v}_e\|_2^2)$. Candidates aligned with already selected directions receive smaller incremental value. With $\mathbf{P} = \mathbf{M}_{\mathcal{S}}^{-1}$, adding e permits the rank-one update

$$\mathbf{P} \leftarrow \mathbf{P} - \frac{\mathbf{P}\mathbf{v}_e\mathbf{v}_e^\top\mathbf{P}}{1 + \mathbf{v}_e^\top\mathbf{P}\mathbf{v}_e}. \tag{20}$$

Cholesky solves may be used instead of explicitly forming the inverse. For $m = |\mathcal{P}|$, direct marginal scanning costs $O(Bmd_z^2)$ and stores $O(md_z + d_z^2)$ numbers; no $m \times m$ matrix is required.

Proposition (budgeted sketch approximation). For fixed finite $\{\mathbf{v}_e : e \in \mathcal{P}\}$ and integer $1 \le B \le m$, the objective in Equation (18) is normalized, monotone, and submodular. Greedy maximization of Equation (19) for B steps satisfies

$$f(\mathcal{S}_B) \ge \left[1 - \left(1 - \frac{1}{B}\right)^B\right] f(\mathcal{S}^\star) \ge (1 - 1/e) f(\mathcal{S}^\star). \tag{21}$$

Proof. Normalization follows from $\det \mathbf{I} = 1$ and monotonicity from the nonnegative quadratic form in Equation (19). If $\mathcal{S} \subseteq \mathcal{T}$, then $\mathbf{M}_{\mathcal{T}} \succeq \mathbf{M}_{\mathcal{S}} \succ 0$, hence $\mathbf{M}_{\mathcal{T}}^{-1} \preceq \mathbf{M}_{\mathcal{S}}^{-1}$. The marginal therefore decreases as the selected set grows, proving submodularity. At step b, submodularity and monotonicity imply that $f(\mathcal{S}^\star) - f(\mathcal{S}_b)$ is at most the sum of the marginals of at most B elements of $\mathcal{S}^\star \setminus \mathcal{S}_b$. Greedy captures at least $1/B$ of this gap. Iterating the gap recurrence from $f(\varnothing) = 0$ proves the bound. □ The bound concerns sketch volume on the fixed candidate pool, not downstream GraphGain; changing embeddings during selection or adding other constraints requires a separate analysis.

Connection to pairwise redundancy. For $\mathbf{X} = \sum_{e \in \mathcal{S}} \mathbf{v}_e\mathbf{v}_e^\top$, expansion in its nonnegative eigenvalues gives

$$f(\mathcal{S}) = \sum_e \|\mathbf{v}_e\|_2^2 - \frac{1}{2}\sum_{e,e'}(\mathbf{v}_e^\top\mathbf{v}_{e'})^2 + R, \qquad 0 \le R \le \frac{1}{3}\operatorname{Tr}(\mathbf{X}^3), \tag{22}$$

where both sums are over $\mathcal{S}$. The remainder bound follows from $\log(1 + x) = x - x^2/2 + \int_0^x t^2/(1 + t)\, dt$ for $x \ge 0$. Thus the criterion contains a quality term and a squared-overlap penalty as its second-order approximation, with higher-order redundancy handled by the determinant. The second-order expression approximates the log-determinant objective of Equation (7).

B.3 Tri-Bottleneck Latent State Formulations

The tri-bottleneck block maps the sketch $\mathcal{S}_B$, entity representations $\mathbf{H}$, and conditioning tokens $\mathbf{t}, \mathbf{c}$ to three diagnostic states:

$$[\mathbf{e}_N, \mathbf{e}_C, \mathbf{e}_U] = \Phi(\mathcal{S}_B, \mathbf{H}, \mathbf{t}, \mathbf{c}), \tag{23}$$

Here, Φ is a shared trunk with three branch-specific heads, and each $\mathbf{e}_k \in \mathbb{R}^{d_e}$. The following subsections define the input signals and interpretation of each state.

1. Relational Necessity ($\mathbf{e}_N$). The necessity branch estimates whether the downstream task requires information beyond individual multimodal representations. To quantify the insufficiency and ambiguity of raw features, we extract two empirical signals. **Prediction Ambiguity ($\mathbf{x}_{\text{amb}} \in \mathbb{R}^3$).** A linear probe is fit on outer-training sources and kept fixed when applied to each held-out source. Its predictions on raw multimodal features $\mathbf{H}$ yield an entropy distribution across entities:

$$\mathbf{x}_{\text{amb}} = \left[\frac{1}{N}\sum_{i=1}^N \mathcal{H}(\hat{\mathbf{p}}_i),\ \operatorname{Var}_i(\mathcal{H}(\hat{\mathbf{p}}_i)),\ \frac{1}{N}\sum_{i=1}^N \operatorname{Margin}_i(\hat{\mathbf{p}}_i)\right], \tag{24}$$

where $\hat{\mathbf{p}}_i$ is the probe's predictive distribution for entity i, $\mathcal{H}(\hat{\mathbf{p}}_i) = -\sum_k \hat{p}_{ik}\log\hat{p}_{ik}$ is the Shannon entropy (Shannon, 1948) (in nats), Var_i is the population variance of the per-entity entropies, and $\operatorname{Margin}_i = \hat{p}_{i,(1)} - \hat{p}_{i,(2)}$ is the probability margin between the top two predicted classes with $\hat{p}_{i,(1)} \ge \hat{p}_{i,(2)}$. These statistics provide diagnostic measures of ambiguity and confidence.

Cross-Modal Divergence ($\mathbf{x}_{\text{div}} \in \mathbb{R}$). For multimodal entities with modalities $m_1, m_2 \in \mathcal{M}$, we compute the mean cosine divergence between modality-specific representations:

$$\mathbf{x}_{\text{div}} = \frac{1}{N}\sum_{i=1}^N \left(1 - \frac{\mathbf{h}_i^{(m_1)\top}\mathbf{h}_i^{(m_2)}}{\|\mathbf{h}_i^{(m_1)}\|_2\|\mathbf{h}_i^{(m_2)}\|_2}\right), \tag{25}$$

where the term inside the parentheses lies in $[0, 2]$ and is zero when the two modality views of an entity agree. Large divergence indicates that individual modalities carry complementary (potentially contradictory) evidence, which relational structure may help reconcile. The necessity state is then parameterized as

$$\mathbf{e}_N = \mathrm{MLP}_N([\,\mathbf{x}_{\mathrm{amb}} \,\|\, \mathbf{x}_{\mathrm{div}} \,\|\, \mathbf{t}\,])\,, \tag{26}$$

where MLP_N maps the $3 + 1 + d$ dimensional input to d_e.

2. Relational Constructability ($\mathbf{e}_C$). The constructability branch estimates whether the candidate constructor can build a structurally sound and reliable graph from available entities. We quantify structural viability via two signals. **Candidate Stability under Perturbation ($\mathbf{x}_{\mathrm{stab}} \in \mathbb{R}$).** We inject Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.05^2\mathbf{I})$ into $\mathbf{H}$, obtaining $\mathbf{H}_{\boldsymbol{\epsilon}} = \mathbf{H} + \boldsymbol{\epsilon}$, and compute the Jaccard similarity between the top-k nearest neighbour candidate sets $\mathcal{N}_k(i)$ of the clean features and $\mathcal{N}_k^{(\boldsymbol{\epsilon})}(i)$ of the perturbed features:

$$\mathbf{x}_{\mathrm{stab}} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\mathcal{N}_k(i) \cap \mathcal{N}_k^{(\boldsymbol{\epsilon})}(i)|}{|\mathcal{N}_k(i) \cup \mathcal{N}_k^{(\boldsymbol{\epsilon})}(i)|}, \tag{27}$$

where $\mathcal{N}_k(i)$ denotes the k nearest neighbours of i under the constructor's similarity. A value near 1 means the candidate neighbourhood is stable under small feature noise, so the constructor is likely to produce a reproducible topology; a low value flags an unstable, noise-driven construction.

Graph Coverage and Giant Component Ratio ($\mathbf{x}_{\mathrm{cov}} \in \mathbb{R}^2$). Over the candidate sketch $\mathcal{S}_B$, we measure the fraction of non-isolated entities and the size ratio of the largest connected component:

$$\mathbf{x}_{\mathrm{cov}} = \left[\frac{|\{i \in \mathcal{V} \mid \deg_{\mathcal{S}_B}(i) > 0\}|}{N},\ \frac{|\mathcal{V}_{\mathrm{giant}}(\mathcal{S}_B)|}{N}\right], \tag{28}$$

where $\deg_{\mathcal{S}_B}(i)$ is the degree of entity i in the sketch graph and $\mathcal{V}_{\mathrm{giant}}(\mathcal{S}_B)$ is its largest connected component. Both quantities are computed on $\mathcal{S}_B$. The constructability state is parameterized as

$$\mathbf{e}_C = \mathrm{MLP}_C([\,\mathbf{x}_{\mathrm{stab}} \,\|\, \mathbf{x}_{\mathrm{cov}} \,\|\, \mathbf{c}\,])\,. \tag{29}$$

3. Relational Utility ($\mathbf{e}_U$). The utility branch estimates whether graph propagation is likely to improve predictive signal for the current task and constructor. We capture propagation gain via non-parametric feature and pseudo-label smoothing. **Propagation Signal Gain ($\mathbf{x}_{\mathrm{prop}} \in \mathbb{R}$).** Let $\mathbf{A}_{\mathcal{S}_B}$ be the adjacency of the sketch, define $\bar{\mathbf{A}} = \mathbf{A}_{\mathcal{S}_B} + \mathbf{I}$, and let $\bar{\mathbf{D}}$ be the diagonal degree matrix of $\bar{\mathbf{A}}$:

$$\bar{D}_{ii} = \sum_j \bar{A}_{ij}, \qquad \tilde{\mathbf{A}}_{\mathcal{S}_B} = \bar{\mathbf{D}}^{-1/2}\bar{\mathbf{A}}\bar{\mathbf{D}}^{-1/2}. \tag{30}$$

For a symmetric nonnegative sketch adjacency, the degree matrix is computed after adding unit self-loops; every node has $\bar{D}_{ii} \geq 1$ and an isolated node retains its own feature exactly. We perform a single non-parametric propagation step $\mathbf{H}^{(1)} = \tilde{\mathbf{A}}_{\mathcal{S}_B}\mathbf{H}$ and compute the relative change in class separation or retrieval ranking consistency:

$$\mathbf{x}_{\mathrm{prop}} = \frac{\mathrm{Tr}(\mathbf{S}_{\mathrm{between}}(\mathbf{H}^{(1)}))/\,\mathrm{Tr}(\mathbf{S}_{\mathrm{within}}(\mathbf{H}^{(1)}))}{\mathrm{Tr}(\mathbf{S}_{\mathrm{between}}(\mathbf{H}))/\,\mathrm{Tr}(\mathbf{S}_{\mathrm{within}}(\mathbf{H})) + \epsilon}, \tag{31}$$

where $\mathrm{Tr}(\cdot)$ is the trace, $\epsilon > 0$ is a numerical guard, and $\mathbf{S}_{\mathrm{between}}$ and $\mathbf{S}_{\mathrm{within}}$ have the standard Fisher discriminant form (Fisher, 1936) computed from pseudo-labels produced by the outer-training probe,

$$\mathbf{S}_{\mathrm{between}} = \sum_k n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top, \quad \mathbf{S}_{\mathrm{within}} = \sum_k \sum_{i \in \text{class } k} (\mathbf{h}_i - \boldsymbol{\mu}_k)(\mathbf{h}_i - \boldsymbol{\mu}_k)^\top, \tag{32}$$

with pseudo-class means $\boldsymbol{\mu}_k$, pseudo-class sizes n_k, and global mean $\boldsymbol{\mu}$. A ratio above 1 indicates increased separation under the probe's pseudo-labels and serves as a diagnostic signal rather than a downstream accuracy measurement. For link prediction and retrieval, a pair scorer fit on outer-training sources compares the ordering of bounded candidate pairs before and after sketch propagation. These probes provide target-free inputs to the utility branch, whose state is parameterized as

$$\mathbf{e}_U = \mathrm{MLP}_U([\,\mathbf{x}_{\mathrm{prop}} \,\|\, \mathbf{t} \,\|\, \mathbf{c}\,])\,. \tag{33}$$

The probes use outer-training labels or interactions and are kept fixed for outer-test evaluation. Node-classification pseudo-labels come from the probe; for link prediction and cross-modal re-

trieval, the scorer ranks candidate pairs generated from raw features and the bounded sketch. The realized gain is attached after the Build/Skip prediction is frozen.

Fusion and gates. The three states are combined into a graphification representation through learned scalar gates,

$$\mathbf{z}_G = f_\psi([\,\alpha_N \mathbf{e}_N \,\|\, \alpha_C \mathbf{e}_C \,\|\, \alpha_U \mathbf{e}_U \,\|\, \mathbf{t} \,\|\, \mathbf{c}\,]), \qquad \alpha_k = \sigma\big(\mathbf{w}_k^\top [\,\mathbf{e}_k \,\|\, \mathbf{t} \,\|\, \mathbf{c}\,]\big), \tag{34}$$

where σ is the logistic function and each $\mathbf{w}_k$ is a learned gate vector. The fused representation is then read out as the predicted value distribution

$$(\mu_V, s_V) = F_\omega(\mathbf{z}_G), \qquad \sigma_V = \mathrm{softplus}(s_V) + \epsilon_\sigma, \tag{35}$$

where F_ω is a two-layer MLP head with parameters ω, μ_V estimates the expected improvement over raw multimodal learning in percentage points, s_V is an unconstrained scale output, and σ_V is a positive Gaussian working-model scale in the same percentage-point units, with $\epsilon_\sigma > 0$. Equation (10) additionally uses a fitted multiplier $a_{\mathrm{cal}} > 0$ to obtain $\tilde{\sigma}_V$ for deployment. The scale is the residual dispersion in GraphGain points. Table 6 reports coverage and width after the multiplier is fit on the calibration sources.

B.4 Multi-Objective Training and Uncertainty Calibration

Episodes and data separation. An offline episode i contains pre-construction features $\mathbf{x}_i$, realized gain V_i^*, and measured cost C_i. Within each outer-training split, partition parent sources into fit/model-selection, scale-calibration ($\mathcal{I}_{\mathrm{cal}}$), and policy-validation groups, with each source kept intact. Hyperparameters are selected in the fit group; the predictor, calibration multiplier, and policy threshold are then fixed in sequence before outer-test evaluation.

Training loss. The objective in Equation (12) is

$$\mathcal{L} = \mathcal{L}_{\mathrm{gain}} + \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{unc}}\mathcal{L}_{\mathrm{unc}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{dec}}. \tag{36}$$

The three episode losses below are averaged over fitting episodes; the ranking loss is averaged over $\mathcal{R} = \{(i,j) : \mathcal{T}_i = \mathcal{T}_j,\ V_i^* > V_j^*\}$ within that split, and is zero if $\mathcal{R}$ is empty. Tied targets are excluded. All λ weights are nonnegative. **Gain Regression Loss.** The per-episode Huber loss, with transition $\delta > 0$ in gain units, is

$$\ell_{\mathrm{gain},i} = \begin{cases} \frac{1}{2}(\mu_{V_i} - V_i^*)^2, & |\mu_{V_i} - V_i^*| \le \delta, \\ \delta(|\mu_{V_i} - V_i^*| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \tag{37}$$

This is quadratic near the target and linear for large residuals. **Pairwise Ranking Loss.** For $(i,j) \in \mathcal{R}$,

$$\ell_{\mathrm{rank},ij} = -\log \mathrm{sigmoid}\left(\frac{\mu_{V_i} - \mu_{V_j}}{\tau_{\mathrm{rank}}}\right). \tag{38}$$

The positive temperature τ_{rank} has gain units, making the logistic argument dimensionless; pairing within task avoids ranking incompatible downstream metrics. **Uncertainty Loss.** Under the working model $V_i^* \mid \mathbf{x}_i \sim \mathcal{N}(\mu_{V_i}, \sigma_{V_i}^2)$, use

$$\ell_{\mathrm{unc},i} = \frac{1}{2}\left[\frac{(V_i^* - \mu_{V_i})^2}{\sigma_{V_i}^2} + \log(2\pi\sigma_{V_i}^2)\right]. \tag{39}$$

This supplies scale supervision; the separate calibration split fixes the deployed multiplier in Equation (41). **Decision Surrogate.** Oracle utility is $U_i^* = V_i^* - \lambda_c C_i$, the same definition as U^* in Section 2. The offline label is $y_i^* = \mathbb{I}[U_i^* \ge 0]$, with ties assigned to Build. During fitting, set $a_{\mathrm{cal}} = 1$ and use $U_i^{\mathrm{fit}} = \mu_{V_i} - \lambda_c \widehat{C}_{\mathrm{pre},i} - \lambda_u \sigma_{V_i}$, $\hat{p}_i = \mathrm{sigmoid}(U_i^{\mathrm{fit}}/T_{\mathrm{dec}})$ with $T_{\mathrm{dec}} > 0$ in gain units. The weighted per-episode loss is

$$\ell_{\mathrm{dec},i} = -w_+ y_i^* \log \hat{p}_i - w_-(1 - y_i^*)\log(1 - \hat{p}_i), \qquad w_+, w_- > 0. \tag{40}$$

Setting $w_+ > w_-$ prioritizes costly false skips. The final threshold is selected on separate policy-validation sources after scale calibration.

Post-hoc Scale Calibration. For each calibration source g with n_g episodes, assign episode weight $w_i = 1/(G_{\mathrm{cal}} n_g)$, where G_{cal} counts calibration sources. This gives each calibration source equal

total weight. With fixed predictions, fit one global scale multiplier by weighted Gaussian NLL:

$$a_{\mathrm{cal}}^2 = \max\left\{\epsilon_a^2, \sum_{i\in\mathcal{I}_{\mathrm{cal}}} w_i \frac{(V_i^* - \mu_{V_i})^2}{\sigma_{V_i}^2}\right\}, \qquad \tilde{\sigma}_{V_i} = a_{\mathrm{cal}}\sigma_{V_i}, \qquad \epsilon_a > 0. \tag{41}$$

Writing $b = a^2$, the terms depending on b are $\frac{1}{2}[\log b + b^{-1}\sum_i w_i(V_i^* - \mu_{V_i})^2/\sigma_{V_i}^2]$. Differentiation gives Equation (41), including its lower-bound constraint. When the bound is inactive, the weighted mean squared standardized calibration residual equals one. Source-held-out coverage is evaluated separately from this fitted residual moment.

Pre-construction cost and threshold. Fit a nonnegative cost regressor $\widehat{C}_{\mathrm{pre}} = g_\eta(\mathbf{x})$ to archived measured costs on the fitting sources, with features available before graph construction. Use the same cost definition and units for g_η targets and C_i. Then λ_c has units of gain per unit cost, while λ_u is dimensionless. On $\mathcal{I}_{\mathrm{pol}}$, choose τ_{thr} from a fixed grid to maximize saved work subject to the required retained positive-gain mass; break ties toward lower thresholds and use Always Build ($\tau_{\mathrm{thr}} = -\infty$) if no finite threshold is feasible. After calibration and threshold selection, the heads, cost regressor, a_{cal}, and τ_{thr} are fixed. Deployment uses U_{build} in Equation (13), replacing C_i by $\widehat{C}_{\mathrm{pre}}$ and subtracting $\lambda_u\tilde{\sigma}_V$. Final saved work and retained mass are evaluated on the outer-test sources.

Uncertainty evaluation protocol. For nominal Gaussian interval level $1-\alpha$, let $z_{1-\alpha/2}$ be a standard-normal quantile and form $[\mu_{V_i} - z_{1-\alpha/2}\tilde{\sigma}_{V_i}, \mu_{V_i} + z_{1-\alpha/2}\tilde{\sigma}_{V_i}]$. On outer-test sources, record parent-source empirical coverage, mean interval width in GraphGain percentage points, and Gaussian NLL before and after scale calibration. Table 6 reports task-macro coverage, mean width, and Gaussian NLL at the 80%, 90%, and 95% nominal levels. Mean width equals $2z_{1-\alpha/2}$ times the task-macro mean scale, so the three widths in a task are proportional to $z_{0.90}$, $z_{0.95}$, and $z_{0.975}$. One calibration multiplier rescales all three widths of that task. NLL does not depend on the nominal level. MAE assesses the predicted mean separately.

Table 6: Source-held-out uncertainty diagnostics before and after scale calibration. Coverage is reported at each nominal interval level; width is in GraphGain percentage points.

Task	Nominal	Raw coverage	Calibrated coverage	Raw width	Calibrated width	Raw / calibrated NLL
NC	80%	72.5	77.4	4.69	3.66	1.48 / 1.22
NC	90%	82.9	87.6	6.02	4.70	1.48 / 1.22
NC	95%	88.1	92.7	7.17	5.60	1.48 / 1.22
LP	80%	70.8	76.8	6.84	5.33	1.74 / 1.42
LP	90%	81.2	87.0	8.78	6.84	1.74 / 1.42
LP	95%	86.4	92.1	10.46	8.15	1.74 / 1.42
CMR	80%	72.0	77.2	4.42	3.45	1.42 / 1.16
CMR	90%	82.4	87.4	5.67	4.43	1.42 / 1.16
CMR	95%	87.6	92.5	6.76	5.28	1.42 / 1.16

Optimization and settings. Optimize the fitting objective with AdamW (Loshchilov & Hutter, 2019); treat δ, the temperatures, loss weights, w_+/w_-, λ_c, and λ_u as configuration parameters selected inside the fitting split. Record them with B, K, and all numerical guards.

Algorithm 1: Candidate Discovery and Regularized Volume Selection

Input: Raw multimodal entities $\mathcal{D} = (\mathcal{V}, \{\mathcal{X}^{(m)}\}_{m\in\mathcal{M}})$, Task $\mathcal{T}$, Constructor $\mathcal{C}$, Candidate pool budget K, and Sketch budget B.
Output: Relation index set $\mathcal{S}_B$ with attached embeddings $\{\mathbf{z}_e : e \in \mathcal{S}_B\}$.

1. Extract entity representations: $\mathbf{H} = E_{\mathrm{mm}}(\mathcal{D}) = \{\mathbf{h}_i\}_{i=1}^{|\mathcal{V}|}$.
2. Embed task and constructor: $\mathbf{t} = E_{\mathrm{task}}(\mathcal{T}) \in \mathbb{R}^d$, $\mathbf{c} = E_{\mathrm{con}}(\mathcal{C}) \in \mathbb{R}^d$.
3. **Inexpensive Candidate Pool Generation:**
   - Project and normalize $\mathbf{h}_i$ with a positive norm guard; retrieve approximate cosine neighbours.
   - Pool top-$\lceil K/N \rceil$ neighbours per entity, deduplicate pairs, and truncate deterministically to at most K candidates. Fix the resulting pool $\mathcal{P}$.
4. **Pairwise Relational Representation:**
   - For each candidate pair $(i, j) \in \mathcal{P}$, compute:
   $$\mathbf{z}_{ij} = \phi_\theta([\mathbf{h}_i \,\|\, \mathbf{h}_j \,\|\, \mathbf{h}_i \odot \mathbf{h}_j \,\|\, |\mathbf{h}_i - \mathbf{h}_j| \,\|\, \mathbf{t} \,\|\, \mathbf{c}]).$$
   - Compute $q_e = \mathrm{softplus}(\mathbf{u}^\top \mathbf{z}_e)$ and $\mathbf{v}_e = q_e \mathbf{z}_e / \max(\|\mathbf{z}_e\|_2, \epsilon_z)$; hold these vectors fixed during extraction.
5. **Greedy Regularized Volume Selection:**
   - Initialize $\mathcal{S} \leftarrow \varnothing$ and $\mathbf{P} \leftarrow \mathbf{I}_{d_z}$.
   - **While** $|\mathcal{S}| < \min(B, |\mathcal{P}|)$, choose (breaking ties by fixed candidate order):
   $$e^* \in \arg\max_{e\in\mathcal{P}\setminus\mathcal{S}} \log(1 + \mathbf{v}_e^\top \mathbf{P} \mathbf{v}_e).$$
   - Apply Equation (20) to $\mathbf{P}$ using e^*, then set $\mathcal{S} \leftarrow \mathcal{S} \cup \{e^*\}$ and repeat.
6. **Return** $\mathcal{S}_B = \mathcal{S}$ and its relation embeddings.

B.5 Theoretical Properties

The results in this section bound sketch-volume and decision-score changes for a fixed candidate pool under stated perturbation and margin conditions; they do not bound downstream GraphGain. We use $\|\cdot\|_2$ for the spectral norm and $\|\cdot\|_*$ for the trace (nuclear) norm.

Result 1 (sketch volume is robust to perturbed relations). Let the candidate pool $\mathcal{P}$ carry clean relation vectors $\{\mathbf{v}_e\}_{e\in\mathcal{P}}$ with $q = \max_e \|\mathbf{v}_e\|_2$, and let $\{\hat{\mathbf{v}}_e\}$ be perturbed vectors with $\|\hat{\mathbf{v}}_e - \mathbf{v}_e\|_2 \le \eta$ for every e. Write $f(\mathcal{S}) = \log\det\big(\mathbf{I} + \sum_{e\in\mathcal{S}} \mathbf{v}_e \mathbf{v}_e^\top\big)$ for the clean objective of Equation (18), $\hat{f}$ for the same objective on $\{\hat{\mathbf{v}}_e\}$, $\mathcal{S}^\star$ for the clean optimum, and $\mathcal{S}_B^{\mathrm{pert}}$ for the greedy solution returned by Algorithm 1 on the perturbed vectors. For $1 \le B \le |\mathcal{P}|$,

$$f(\mathcal{S}_B^{\mathrm{pert}}) \;\ge\; \Big(1 - \tfrac{1}{e}\Big) f(\mathcal{S}^\star) \;-\; \Big(2 - \tfrac{1}{e}\Big) B\eta(2q + \eta). \tag{42}$$

The bound applies to pointwise vector perturbations on a fixed candidate pool. Interventions that change the pool or relation types require separate analysis. Result 2 adds a margin condition on the score.

Proof. (a) $\log\det$ *is* 1*-Lipschitz in the trace norm on PSD inputs.* For $g(\mathbf{X}) = \log\det(\mathbf{I}+\mathbf{X})$ we have $\nabla g(\mathbf{X}) = (\mathbf{I}+\mathbf{X})^{-1}$, whose spectral norm is at most 1 for $\mathbf{X} \succeq 0$. By the fundamental theorem of calculus along the segment $\mathbf{X}(t) = (1-t)\mathbf{A} + t\mathbf{B}$, $|g(\mathbf{A}) - g(\mathbf{B})| \le \sup_t \|\nabla g(\mathbf{X}(t))\|_2 \, \|\mathbf{A}-\mathbf{B}\|_* \le \|\mathbf{A}-\mathbf{B}\|_*$. *(b) Uniform transfer between* f *and* $\hat{f}$. For $|\mathcal{S}| \le B$, $\big\| \sum_{e\in\mathcal{S}} (\mathbf{v}_e\mathbf{v}_e^\top - \hat{\mathbf{v}}_e\hat{\mathbf{v}}_e^\top) \big\|_* \le \sum_{e\in\mathcal{S}} \|\mathbf{v}_e - \hat{\mathbf{v}}_e\|_2 (\|\mathbf{v}_e\|_2 + \|\hat{\mathbf{v}}_e\|_2) \le B\eta(2q+\eta) =: \delta$, using $\|\mathbf{x}\mathbf{x}^\top - \mathbf{y}\mathbf{y}^\top\|_* \le \|\mathbf{x}-\mathbf{y}\|_2(\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2)$ and $\|\hat{\mathbf{v}}_e\|_2 \le q+\eta$. By (a), $|f(\mathcal{S}) - \hat{f}(\mathcal{S})| \le \delta$ for every $|\mathcal{S}| \le B$. *(c) Chain.* $\hat{f}$ is normalized, monotone, and submodular, so greedy on $\hat{f}$ gives $\hat{f}(\mathcal{S}_B^{\mathrm{pert}}) \ge (1-\frac{1}{e})\hat{f}(\mathcal{S}^\star)$. Then $f(\mathcal{S}_B^{\mathrm{pert}}) \ge \hat{f}(\mathcal{S}_B^{\mathrm{pert}}) - \delta \ge (1-\frac{1}{e})\hat{f}(\mathcal{S}^\star) - \delta \ge (1-\frac{1}{e})(f(\mathcal{S}^\star) - \delta) - \delta$, which rearranges to Equation (42). □

Result 2 (score-error decomposition under a margin condition). For a source g, let e_b be the greedy relation selected at step b from $\mathcal{S}_{b-1}$ using the clean marginal in Equation (19). Write γ_b for its margin over the best unselected competitor, and set $\delta_B = B\eta_v(2q+\eta_v)$ for the vector perturbation bound η_v of Result 1. If $\gamma_b > 4\delta_B$ for every $b \le B$, clean and perturbed greedy selection return the same relation indices. Suppose in addition that the selected embeddings change by at most η_z each, the cost features and frozen threshold τ do not change, and the deployed score is L_g-Lipschitz in the concatenated selected embeddings. Then

$$|\widehat{U}_g^{\mathrm{pert}} - \widehat{U}_g^{\mathrm{clean}}| \le L_g\sqrt{B}\,\eta_z, \qquad |\widehat{U}_g^{\mathrm{clean}} - \tau| > L_g\sqrt{B}\,\eta_z \implies a_g^{\mathrm{pert}} = a_g^{\mathrm{clean}}. \tag{43}$$

The score's Lipschitz factor combines the mean head, calibrated scale, and λ_u in Equation (13); the same-decision guarantee holds when both margin conditions are met.

Proof. Result 1 gives $|f(S) - \hat{f}(S)| \le \delta_B$ whenever $|S| \le B$. Each marginal is a difference of two such values, so its error is at most $2\delta_B$. A clean winner ahead by more than $4\delta_B$ remains ahead after both marginals move; induction on b gives the same indices. For these indices, concatenating the B embedding perturbations has Euclidean norm at most $\sqrt{B}\eta_z$. Lipschitz continuity gives the first inequality in Equation (43). The second inequality keeps both scores on the same side of the frozen threshold. □

Result 3 (uncertainty-based lower bound on retained gain). Consider a target source g with episodes $i \in g$, and write $[x]_+ = \max(0,x)$. Suppose the calibrated scale satisfies the simultaneous residual condition $|V_i^* - \mu_{V_i}| \le \beta\tilde{\sigma}_{V_i}$ for every episode of that source, where $\beta > 0$ is fixed before evaluating g. The constant β is the maximum of $|V_i^* - \mu_{V_i}|/\tilde{\sigma}_{V_i}$ on calibration sources, excluding g. For any frozen Build/Skip policy $a_i \in \{0,1\}$, define the quantities available before construction:

$$\begin{aligned} L_g &= \sum_{i\in g} [\mu_{V_i} - \beta\tilde{\sigma}_{V_i}]_+, \\ M_g &= \sum_{i\in g} (1-a_i)[\mu_{V_i} + \beta\tilde{\sigma}_{V_i}]_+, \\ R_g &\ge \max\{0, 1 - M_g/L_g\} \qquad (L_g > 0). \end{aligned} \tag{44}$$

Here β and the threshold are fixed before source g is scored. Q3 instead chooses τ_{thr} by nested search on the outer-training split and reports empirical retention using Equation (49). For $L_g = 0$, the bound is vacuous; averaging source-level bounds requires the residual condition to hold for every included source.

Proof. Under the stated residual condition, $[V_i^*]_+ \ge [\mu_{V_i} - \beta\tilde{\sigma}_{V_i}]_+$ and $[V_i^*]_+ \le [\mu_{V_i} + \beta\tilde{\sigma}_{V_i}]_+$. The total positive-gain mass on g is at least L_g, while the mass lost on Skip episodes is at most M_g. Since retained mass equals one minus the ratio of lost to total positive mass, Equation (44) follows; its lower bound is clipped at zero. □

C EXPERIMENTS IN DETAIL

C.1 DATASETS, TASKS, AND SPLITS

The evaluation covers six multimodal benchmarks catalogued by OpenMAG (Wan et al., 2026), spanning node classification, link prediction, and cross-modal retrieval. Table 7 separates published source-graph sizes from feature dimensions and evaluation splits.

Table 7: Source benchmark statistics for the six datasets. Source edges are published graph counts. Feature dimensions correspond to frozen text and visual encoders; split ratios specify the evaluation protocol, including a held-out-edge split for link prediction.

Benchmark	Downstream Task	Entities ($\lvert\mathcal{V}\rvert$)	Source Edges	Modalities	Dims (T/V)	Classes / Targets	Split (Train/Val/Test)
Toys	Node Classification	20,695	126,886	Text, Visual	768 / 768	18 categories	60% / 20% / 20%
Grocery	Node Classification	17,074	171,340	Text, Visual	768 / 768	20 categories	60% / 20% / 20%
Bili Music	Link Prediction	6,038	21,592	Text, Visual	768 / 768	Binary interaction	70% / 10% / 20%
DY	Link Prediction	8,299	35,627	Text, Visual	768 / 768	Binary interaction	70% / 10% / 20%
QB	Cross-modal Retrieval	6,121	24,145	Text, Visual	768 / 768	Query-Item Pairs	60% / 20% / 20%
Bili Cartoon	Cross-modal Retrieval	4,724	18,660	Text, Visual	768 / 768	Query-Video Pairs	60% / 20% / 20%

Toys & Grocery (Ni et al., 2019). Derived from the Amazon Review collection, each entity corresponds to a commercial product with textual metadata and a product image. The evaluation uses frozen 768-dimensional CLIP text and visual features (Radford et al., 2021). The task is multi-class product categorization evaluated via Accuracy (ACC %). **Bili Music & DY (Zhang et al., 2025a).** Multimodal video interaction datasets containing video entities with multi-attribute text titles, descriptive tags, and video frame representations. The task is link prediction between user-item or item-item co-occurrence graphs, evaluated via the Area Under the ROC Curve (AUC %). **QB & Bili Cartoon (Zhang et al., 2025a).** E-commerce and anime entertainment platforms containing query-item and cover-video multimodal pairs. Downstream cross-modal retrieval is evaluated using Recall at Rank 5 among 100 candidate pools (R@5@100 %). The registry contains 13 parent sources: the six benchmarks above, plus *Movies*, *Reddit-S*, *Reddit-M*, the *books* collection, *cloth-copurchase*, *ele-fashion*, and *sports-copurchase*. The books parent is evaluated once for link prediction (*books-lp*) and once for node classification (*books-nc*); both evaluations stay in the same outer fold and count as one parent source. Q1a uses the six named benchmarks; Q1b, Q3, and Q4 use the 13-source registry. Q2 uses a separate rebuilt 11-source bank (Appendix D.1). Table 7 reports published source graph sizes. RawOnly uses the graph-free path. Node-classification and retrieval splits use $60/20/20$, and link prediction uses a $70/10/20$ train/validation/test interaction split. Text and visual encoders remain frozen.

C.2 EVALUATION METRICS

Signed GraphGain (percentage points). For benchmark d, constructor c, and seed s:

$$\mathrm{GraphGain}_{d,c,s} = 100 \cdot (y^{\mathrm{graph}}_{d,c,s} - y^{\mathrm{raw}}_{d,s}). \tag{45}$$

Point-prediction error and predictive uncertainty are evaluated separately. MAE measures gain error; it is not the confidence-calibration metric studied by Guo et al. (2017). Appendix B.4 specifies coverage, interval width, and NLL for the uncertainty model. Build/Skip is related to selective prediction (Geifman & El-Yaniv, 2019), with Skip selecting the raw-feature path rather than withholding an output. **Spearman Rank Correlation Coefficient (ρ).**

$$\rho = \frac{\sum_i (r_i - \bar{r})(\hat{r}_i - \bar{\hat{r}})}{\sqrt{\sum_i (r_i - \bar{r})^2 \sum_i (\hat{r}_i - \bar{\hat{r}})^2}}, \tag{46}$$

where r_i and $\hat{r}_i$ are average ranks of the realized and predicted gain, respectively, over paired episodes; bars are their sample means. This Pearson-on-ranks form handles ties (Spearman, 1904). **Kendall Rank Correlation Coefficient (τ_b).**

$$\tau_b = \frac{C - D}{\sqrt{(C + D + T_r)(C + D + T_{\hat{r}})}}, \tag{47}$$

where C and D count concordant and discordant pairs, T_r counts pairs tied only in realized gain, and $T_{\hat{r}}$ counts pairs tied only in predicted gain (Kendall, 1938). Pairs tied in both are excluded. If a rank sequence is constant, its correlation is undefined and the corresponding episode group is

excluded from macro averaging. **Mean Absolute Prediction Error (MAE) in percentage points.**

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{V}_i - V_i^*|. \tag{48}$$

Here $\hat{V}_i = \mu_{V_i}$ is the predicted value and V_i^* the realized value for episode i; both are expressed in percentage points so that MAE shares the unit of GraphGain. **Retained Positive Gain Mass.** For an outer-test parent source g, let I_g index its episodes, a_i be the Build indicator, and $G_i = [V_i^*]_+$. When $\sum_{i\in I_g} G_i > 0$, define

$$R_g(\pi) = \frac{\sum_{i\in I_g} a_i G_i}{\sum_{i\in I_g} G_i}, \qquad R_{\mathrm{tm}}(\pi) = \frac{1}{3}\sum_{t\in\{\mathrm{NC,LP,CMR}\}} \frac{1}{|\mathcal{G}_t^+|}\sum_{g\in\mathcal{G}_t^+} R_g(\pi), \tag{49}$$

where $\mathcal{G}_t^+$ contains task-t sources with defined positive-gain mass. The Q3 tables and Figure 4 report task-macro retention, averaging source ratios within each task before averaging the three task families.

Saved Graph Work. Let $C_i > 0$ be the graphification cost $C(\mathcal{D}, \mathcal{C})$ of episode i, in the same units as the cost term of U^*. With Build indicator a_i,

$$S_g(\pi) = 1 - \frac{\sum_{i\in I_g} a_i C_i}{\sum_{i\in I_g} C_i}, \qquad S_{\mathrm{tm}}(\pi) = \frac{1}{3}\sum_{t} \frac{1}{|\mathcal{G}_t|}\sum_{g\in\mathcal{G}_t} S_g(\pi). \tag{50}$$

Always-Build sets every $a_i = 1$ and saves no graphification cost. A Skip decision sets $a_i = 0$ and removes C_i from the numerator. Q3 reports S_{tm}.

Fixed-zero-threshold sensitivity. The Q4 sensitivity grid holds the decision threshold at zero, so its action is $a_i(0) = \mathbb{I}[\widehat{U}_i \geq 0]$, whereas Q3 selects a threshold inside each outer-training split. Both use the source-level numerator and denominator in Equation (49). Table 4 reports this fixed-zero-threshold reference and each factor's grid maximum. Q3's 96.7% is measured at a separately selected nested threshold.

C.3 Graph Constructors in Detail

The constructor comparison evaluates nine constructors. Each graph topology uses a shared two-layer GCN backbone (Kipf & Welling, 2017) with hidden dimension 128, dropout 0.5, and learning rate 0.01. The backbone follows the message-passing design surveyed in Wu et al. (2021) and Zhou et al. (2020). Table 1 averages all nine. **RawOnly (Reference).** A graph-free multilayer perceptron baseline trained strictly on fused raw multimodal features without message passing. Defines the mathematical zero gain reference ($\Delta \equiv 0$). **kNN-GCN (Kipf & Welling, 2017).** Single-modal k-nearest neighbor graph constructed by computing pairwise Euclidean distances over normalized text embeddings, retaining $k = 10$ nearest neighbors per entity. **MM-kNN-GCN.** Multimodal k-nearest neighbor graph constructed over fused multimodal features with equal modality weights: $d_{ij} = 0.5d_{\mathrm{text}}(i,j) + 0.5d_{\mathrm{vis}}(i,j)$, with $k = 10$. **AutoG (Chen et al., 2025).** Formulates graph construction from tabular data and uses an LLM-based solution to generate high-quality graph schemas without human intervention. **AutoGraph (Shan et al., 2025).** LLM-based construction framework that infers entity knowledge as semantic vectors, extracts latent factors via vector quantization, and links entities through the resulting latent-factor nodes. **auGraph (Cucumides & Geerts, 2025).** Task-aware graph augmentation that promotes predictive non-key attributes into nodes via relevance scoring functions, preserving the original schema while injecting task-relevant structure. **LLaTA** (Zhang et al., 2025b) uses tree-guided LLM inference for graph structure learning. **GAugLLM** (Fang et al., 2024) generates graph views for contrastive learning. **LAGA** (Wang et al., 2026) adapts a frozen language backbone for text-attributed graphs. **GraphMaster** (Du et al., 2025) uses LLM agents for graph synthesis when labeled data are limited. These four constructors are trained and scored with the same protocol as the five constructors above. **Related directions.** Our design is informed by the broader literature on graph structure learning and augmentation (Zhu et al., 2021; Zhao et al., 2021), including recent work on LLM-assisted structure discovery and adaptation (Jin et al., 2024; Ren et al., 2024; Chen et al., 2024b; He et al., 2024; Chen et al., 2024a).

C.4 Predictor Baselines

We compare MAG-Scout with four pre-construction prediction and model-selection baselines. **NetInfoF (Lee et al., 2024).** Computes network usable information over sampled candidate relations to estimate whether relational topologies convey task-relevant mutual information. **WDGH (When Do GNNs Help) (Luan et al., 2023).** Analyzes entity homophily and distinguishability metrics to predict whether graph neural networks outperform graph-free baselines. **GLEMOS-S2 (Park et al., 2023).** A meta-learning benchmark method that extracts structural and feature meta-statistics to predict graph model performance across transferable domains. **MetaGL (Park et al., 2022).** An evaluation-free meta-learning framework that trains a surrogate predictor over historical graph learning episodes for model recommendation. All four baselines use the same outer folds and prediction inputs as MAG-Scout, including task and constructor identifiers and bounded relational evidence before graph construction.

C.5 Implementation Details

For Q4 sensitivity, the reference predictor is a two-layer MLP with utility-head width 32. It is trained for 40 epochs with AdamW (learning rate 2×10^{-3}, weight decay 10^{-4}), smooth-ℓ_1, and BCE weight 0.5. The one-factor grid varies learning rate, BCE weight, relation budget, or utility-head width in $\{16, 32, 48\}$; corruption runs use the same reference model.

The graphification predictor uses the four-term objective in Equation (36), regularized volume selection, source-separated scale calibration, and independently selected policy thresholds. Constructor comparisons use a shared two-layer GCN backbone (hidden dimension 128, dropout 0.5, learning rate 0.01) and matched tuning budgets for graph-based and graph-free encoders. Downstream models use Adam (learning rate 10^{-3}, weight decay 10^{-5}), batch size 512, up to 200 epochs, early stopping with patience 30, dropout 0.3, and a 512-dimensional retrieval head for cross-modal tasks.

The deployed decision score combines predicted gain, calibrated uncertainty, and pre-construction cost (Equation (13)); saved graph work is computed using Equation (50). Thresholds are selected on inner policy-validation sources and fixed before outer evaluation. Model runs use the five matched seeds listed in Appendix A. Confidence intervals use 10,000 parent-source bootstrap resamples with generator state `20260916`; perturbation thresholds are fixed using the clean run. Experiments use the hardware and software environment described in Appendix A.

C.6 Source Provenance and Audit Coverage

Q1b, Q3, and Q4 use the 13 parent sources. Eleven parents have raw text features, visual representations, adjacency records, and evaluation splits verified against official SHA-256 receipts. Reddit-M and the books collection are represented by registry records and receipt hashes. Reddit-M and books-lp are absent from the public OpenMAG release used for the bit-level audit. For books-nc, the scorecards use the raw-feature audit accuracy of 0.82; the sealed-registry record lists 0.99 for that audit. books-lp and books-nc are two task evaluations of one parent. On the 11 receipt-verified parents, MAG-Scout retains 97.1% of positive-gain mass and saves 24.1% of graph work. The Q2 ablation uses a separately rebuilt 11-source bank.

D Additional Experimental Results and Extended Analyses

D.1 Task-Level Ablation Results

Table 8 gives each variant's change from the full model on the rebuilt 11-source bank (5 NC, 4 LP, 2 CMR). Absolute scores and seed-level standard deviations appear in Table 3; interval-calibration results are in Table 6.

D.2 Build/Skip Operating Points

Section 6.4 reports held-out saved graph work, retained positive-gain mass, and build rate at the nested validation-selected threshold. The next subsection gives the results for five retention targets.

Table 8: Component ablations on the rebuilt 11-source bank, reported as differences from Full MAG-Scout in Table 3. Rank correlations are subtracted from Full. MAE differences are positive when absolute error increases. The bank has 5 node-classification, 4 link-prediction, and 2 cross-modal sources, with three model seeds.

Ablation Variant	**Δ Spearman ρ**			**Δ Kendall τ**			**Δ MAE**		
	NC	**LP**	**CMR**	**NC**	**LP**	**CMR**	**NC**	**LP**	**CMR**
Frozen Heuristic Sketch	-0.022	-0.028	-0.035	-0.019	-0.027	-0.029	$+0.177$	$+0.416$	$+0.254$
w/o Necessity ($\mathbf{e}_N$)	-0.057	-0.038	-0.068	-0.049	-0.038	-0.064	$+0.447$	$+0.766$	$+0.617$
w/o Constructability ($\mathbf{e}_C$)	-0.071	-0.051	-0.037	-0.060	-0.050	-0.032	$+0.371$	$+0.880$	$+0.479$
w/o Utility ($\mathbf{e}_U$)	-0.079	-0.093	-0.062	-0.074	-0.083	-0.055	$+0.742$	$+1.499$	$+0.952$
Uniform Bottleneck Gates	-0.137	-0.105	-0.085	-0.115	-0.095	-0.071	$+0.625$	$+1.158$	$+0.765$
w/o Pairwise Interaction	-0.037	-0.037	-0.039	-0.033	-0.033	-0.034	$+0.208$	$+0.489$	$+0.342$
w/o Ranking Loss ($\mathcal{L}_{\text{rank}}$)	-0.066	-0.061	-0.052	-0.063	-0.053	-0.045	$+0.131$	$+0.338$	$+0.168$
w/o Decision Loss ($\mathcal{L}_{\text{dec}}$)	-0.017	-0.017	-0.018	-0.014	-0.015	-0.015	$+0.558$	$+1.105$	$+0.673$
w/o One-time Refinement	-0.106	-0.078	-0.059	-0.093	-0.068	-0.054	$+0.402$	$+0.644$	$+0.453$
Condition Only	-0.20	-0.19	-0.14	-0.17	-0.16	-0.12	$+1.24$	$+2.37$	$+1.82$

D.3 Decision Boundary and Threshold Sensitivity

At the selected 95% target, MAG-Scout avoids 23.6% of graph work while retaining 96.7% of positive-gain mass, averaged over tasks. Table 9 compares five targets: $\{90\%, 92.5\%, 95\%, 97.5\%, 99\%\}$. For each target, nested validation selects a threshold as described in Section 6.4.

Table 9: Build/Skip policies at five retention floors on 13 parent sources. An asterisk (*) marks a held-out retention below the floor named in that column.

Method	**Held-out Retained Gain Mass (%)**					**Avoided Graph Work (%)**				
	90%	**92.5%**	**95%**	**97.5%**	**99%**	**90%**	**92.5%**	**95%**	**97.5%**	**99%**
MAG-Scout	**96.7**	**96.7**	**96.7**	96.8*	96.8*	24.2	24.2	**23.6**	19.6	18.4
NetInfoF	96.3	96.4	96.4	96.4*	97.6*	24.3	23.5	23.2	22.6	21.8
WDGH	85.4*	90.5*	90.5*	90.5*	90.5*	25.0	23.8	23.5	22.3	22.3
GLEMOS-S2	60.0*	60.2*	60.5*	60.9*	64.2*	34.8	32.9	30.3	28.6	24.1
MetaGL	80.2*	80.2*	80.2*	80.4*	83.5*	32.6	32.6	32.6	29.7	25.9

Note: Values are task-macro source averages. A separate inner policy-validation threshold is selected for each floor and fixed for outer evaluation.

D.4 Relational Evidence Corruption and Stress-Testing Details

Section 6.5 tests four perturbations of $\mathcal{S}_B$ using seed `20260813`; the predictor, threshold, and weights are fixed from the clean run before perturbed inputs are scored. **Rewiring.** We reassign the content rows for candidate relation types `C_same` and `C_different` by random derangement. Each affected row receives another candidate row's content, while the relation-type inventory stays fixed. Sources without pair-relation rows use the labelled full-bank derangement fallback. **Relation Collapse.** We replace the content of pair-relation rows with their mean vector, leaving one distinct pair-relation content. Sources without these rows use the labelled all-bank collapse fallback. **Single-Modality Dominance.** We retain only multimodal (`M`) relations and suppress the typed auxiliary objectives for other relation types. **Relation-Bank Partition.** We retain a deterministic half-partition of the relation bank. Because relation rows have no global entity identities, the partition is applied to relation rows. Across the four perturbations, retained-mass changes range from -1.6 to -0.1 points, the largest build-rate change is -14.3 points, and the largest number of changed parent decisions is 3 of 13.

D.5 Hyperparameter Sensitivity and Corruption Summaries

Table 10 reports one-factor settings and the budget by loss-weight interaction grid at a fixed zero threshold. Table 11 reports frozen-policy changes by task. Table 4 summarizes these protocols; Q3 uses its separately selected 95%-floor threshold.

Table 10: One-factor sensitivity at a fixed zero threshold, followed by the budget × loss-weight grid. Retained mass is task-macro positive-gain mass. One-factor differences use that factor's reference: 81.9, 96.2, 87.8, and 93.0. Interaction differences use the 90.5 grid reference.

Configuration	Grid Points	Retained (%)	Δref (percentage points)
Selected setting in Figure 5(a)	4	81.9 / 96.2 / 87.8 / 93.0	0
Learning Rate ($0.001 \rightarrow 0.004$)	3	83.6	+1.7
BCE Loss Weight ($0.0 \rightarrow 1.0$)	3	96.5	+0.3
Relation Budget ($0.50 \rightarrow 0.75$)	3	88.2	+0.4
Utility Head Width ($16 \rightarrow 48$)	3	93.4	+0.4
Retained-fraction × loss-weight interaction grid (all four cells)			
Budget 0.50, BCE 0.0	1	86.4	−4.1
Budget 0.50, BCE 1.0	1	91.7	+1.2
Budget 0.75, BCE 0.0	1	89.3	−1.2
Budget 0.75, BCE 1.0	1	94.8	+4.3

The selected-setting cell lists the learning-rate, BCE-weight, relation-budget, and head-width references in that order. Each following row is that factor's grid maximum. The interaction rows are measured from the 90.5 reference. Q3 operating points use the nested 95% floor in Appendix D.2.

Table 11: Frozen-threshold evidence corruption on 13 parent sources. The predictor and threshold are frozen before perturbation. ΔRet. is the task-macro change in retained positive-gain mass; ΔBuild is the build-rate change in percentage points. Macro flips count parent sources. Task rows count evaluations: the books parent contributes one node-classification evaluation and one link-prediction evaluation.

Perturbation Family	Scope	ΔRet.	ΔBuild	Flips
Rewiring	Macro (13)	−0.4	−7.1	2/13
	Node classification (5)	−0.7	−13.4	1/5
	Link prediction (5)	−0.3	−4.7	1/5
	Cross-modal retrieval (4)	−0.2	−2.2	0/4
Relation Collapse	Macro (13)	−0.8	−7.1	2/13
	Node classification (5)	−1.4	−12.1	1/5
	Link prediction (5)	−0.6	−5.4	1/5
	Cross-modal retrieval (4)	−0.3	−2.8	0/4
Single-Modal Dominant	Macro (13)	−1.6	−14.3	3/13
	Node classification (5)	−2.3	−18.6	1/5
	Link prediction (5)	−1.6	−16.4	1/5
	Cross-modal retrieval (4)	−0.7	−6.2	1/4
Relation-Bank Partition	Macro (13)	−0.1	−0.2	0/13
	Node classification (5)	−0.2	−0.6	0/5
	Link prediction (5)	−0.1	+0.3	0/5
	Cross-modal retrieval (4)	+0.1	−0.4	0/4

Macro rows match Table 4. Retention averages sources with positive-gain mass within each task. Build rate and flips use the 13 parent sources. Appendix D.4 defines each perturbation.