GRAPHSELECT FOR BUDGETED REPRESENTATION SELECTION IN MULTIMODAL GRAPH INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal graph predictors combine text, images, and relations to classify connected entities. How much of this input is needed to preserve their predictions? We study budgeted representation selection, which chooses a subset of candidate text and image vectors under a separate capacity for each modality. Predictions from the complete candidate input define the classes to preserve. The challenge is that a representation's contribution depends on the other selected inputs, while graph propagation extends its effects across nodes. Our empirical study shows that candidate rankings change with the selected input, while predicted probabilities remain informative after the class stops changing. Updating scores improves selection, and exchanging inputs can improve a subset whose capacity is already filled. These findings lead to GRAPHSELECT, which starts from individual candidate gains and refines the subset through jointly evaluated exchanges. It screens promising removals and additions, accepts an exchange when it reduces the prediction loss, and updates the scores. Experiments on six graphs show higher mean objective recovery than six attribution and explanation methods adapted to the selection task. Across nine trained architectures on two graphs, retaining 20% of the candidate representations per modality gives a mean accuracy drop of 0.10 percentage points relative to full candidate input, preserving classification performance with substantially fewer text and image representations.

1 INTRODUCTION

Multimodal attributed graphs associate connected entities, such as products, artworks, and social media posts, with text and images (Ektefaie et al., 2023; Peng et al., 2024; Yan et al., 2025; Wan et al., 2026). By combining these attributes with relations, graph models support recommendation (Wei et al., 2019; Tao et al., 2020; Guo et al., 2024), cross-modal matching (Liu et al., 2020), and node analysis (Zhu et al., 2025; Guo et al., 2025; Zheng et al., 2026; Zhang et al., 2026). These predictors can draw on many representations whose contributions overlap. This raises a question about the input they need. Can a small subset of the available text and image representations preserve the predictions obtained with complete input?

We study this question through *budgeted representation selection for multimodal graph inference*. Given an encoded graph and a trained predictor, we select a prescribed number of candidate text and image vectors. For example, a 20% capacity retains 44 of 220 candidate vectors in each modality. The output is one joint subset, evaluated by how well it supports the classes predicted from complete candidate input. Full input supplies the reference predictions for selection. The graph and representations outside the candidate pool remain available, while unselected candidates use the mean training representation of their modality.

A natural approach scores candidates individually and keeps the highest-ranked inputs. The difficulty is that a useful input may duplicate information already selected, and a neighbor's representation may support predictions at several nodes. In the Grocery example in Fig. 1, a target product keeps its own text and image, yet loses confidence in its predicted category when a neighboring image is omitted. Exchanging another selected image for that neighbor's image restores most of the confidence at the same capacity. Individual scores favor the fern image, but their contributions within the selected subset favor this exchange.

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

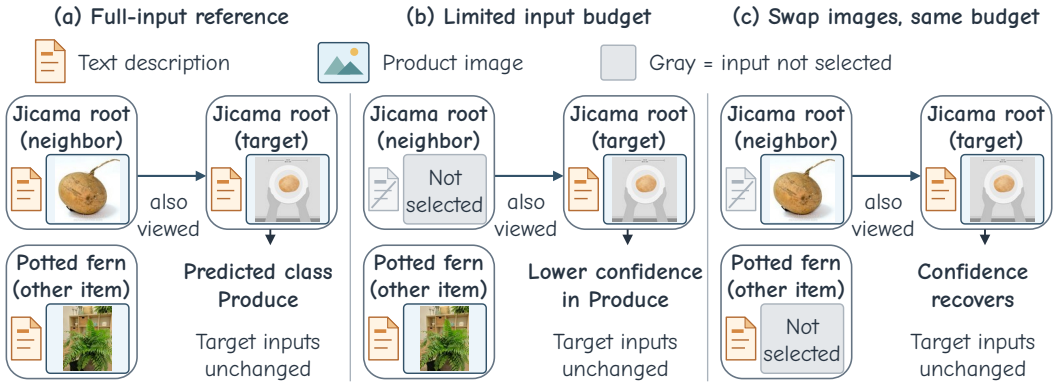

Figure 1: **A neighboring image can help preserve a graph prediction.** Each box represents a product. The two jicama products have an also-viewed relation. Full input provides the reference prediction. With a limited budget, omitting the neighbor's image lowers confidence in Produce. Exchanging the fern's image for the neighbor's restores most of that confidence at the same budget. The target's own inputs remain available. Gray inputs are not selected and use modality means.

To understand this difference, we examine what candidate scores should measure and when they should be updated. Deleting an input from a complete set can give a different ranking from adding it to a partial set. Hard class agreement can also stop distinguishing candidates while their predicted probabilities still change. Scoring the current subset with predicted-class cross-entropy addresses both issues, and recomputing scores improves selection as the subset grows. Once the capacity is filled, exchanges allow the search to revise earlier choices.

Motivated by these findings, GRAPHSELECT starts from individual candidate gains and refines the subset through joint removal and addition. It screens promising replacements, evaluates each pair with the trained predictor, and updates the scores after an improvement. Unlike FORWARD, which recomputes addition gains but keeps every earlier choice, this search can replace an input that has become less useful. For additive graph predictors, changes are evaluated exactly on the affected target nodes.

Our Contributions. (1) **In-depth Investigation.** We formulate representation selection under separate modality capacities and examine how the scoring input, prediction objective, and score updates affect selection quality in multimodal graphs. (2) **Novel Method.** We introduce GRAPHSELECT, which refines an initial subset through screened exchanges. Joint evaluation accounts for interactions within each replacement, while score updates adapt the search to the selected subset. (3) **Empirical Evaluation.** Across six graphs, GRAPHSELECT achieves higher mean recovery than six attribution and explanation methods adapted to the selection task. Experiments with nine trained architectures on two graphs show how increasing capacity improves objective recovery and reduces downstream prediction gaps.

2 RELATED WORKS

Multimodal graph learning. Multimodal attributed graphs combine relations with heterogeneous node attributes (Ektefaie et al., 2023; Peng et al., 2024; Yan et al., 2025; Wan et al., 2026; Zhu et al., 2025). Graph convolution and attention integrate these inputs for recommendation (Wei et al., 2019; Tao et al., 2020; Guo et al., 2024), while semantic matching, filtering, and dynamic graph construction capture interactions across modalities and nodes (Liu et al., 2020; Guo et al., 2025; Zheng et al., 2026; Zhang et al., 2026). Recent models improve how this context is used. LION couples modality alignment and fusion through graph propagation (Li et al., 2026), and RoleMAG assigns neighbors different propagation roles according to their contribution (Zuo et al., 2026). Other directions address generative tasks and cold-start nodes (Yoon et al., 2023; Hu et al., 2026), global transformers (Hu et al., 2025), and unified embeddings or language model integration (Ning et al.,

2025; Liu et al., 2025; Fang et al., 2025; Fan et al., 2025; He et al., 2025). GraphSelect studies how these trained predictors preserve their decisions when a subset of encoded node inputs is retained.

Modality selection and acquisition. Modality selection studies which information sources provide complementary predictive value. Greedy modality selection formalizes this problem through a utility function with approximate submodularity (Cheng et al., 2022). CAMA allocates the acquisition of an additional modality across a cohort by estimating the utility of the missing information (Rheude et al., 2025). Data selection instead chooses examples that contribute to model training, using quantities such as gradient matching or data value (Killamsetty et al., 2021; Wang et al., 2024). Our decision unit is one node's encoded text or image representation. Selection allocates separate capacities across these available representations and evaluates their joint effect on graph predictions.

Attribution and graph explanation. Feature attribution assigns contributions to model inputs through gradients, perturbations, or Shapley values (Sundararajan et al., 2017; Kokhlikyan et al., 2020; Shapley, 1953). GNNExplainer identifies a compact subgraph and feature subset for a prediction (Ying et al., 2019), while SubgraphX, GNNShap, and GOAt develop graph-specific contribution estimates (Yuan et al., 2021; Akkas & Azad, 2024; Lu et al., 2024). GraphFramEx emphasizes evaluating explanations according to their sufficiency and necessity for the prediction (Amara et al., 2022). Their scores provide a starting point for selecting a common subset across nodes and modalities. Our empirical study examines how the scored input, objective, and score updates affect this decision.

Prediction-preserving input subsets. Sufficient input subsets identify small sets of observed features that retain a model decision, using backward selection (Carter et al., 2019). Zorro constructs sparse graph explanations whose predictions remain stable when omitted inputs are randomly perturbed (Funke et al., 2023). Our setting evaluates a common subset of node-level representations over a target-node set, with a separate capacity for each modality. GraphSelect optimizes probability assigned to the full-input classes and revises the subset through jointly evaluated exchanges. Zorro is included in the comparison with its randomized class-retention objective, allowing the two selection criteria to be examined within the same candidate and capacity setup.

3 Problem Formulation and Empirical Study

3.1 Problem Formulation

Graph, predictor, and candidate pool. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}_G)$ be a multimodal attributed graph with adjacency matrix A, and let H contain its encoded text and image representations. A trained predictor $f_\theta(A, H)$ produces class logits at each node. Selection changes only the availability of candidate representations, leaving the graph, encodings, and predictor parameters unchanged. We designate N candidate nodes whose text and image vectors are subject to capacity limits, and a target set $\mathcal{T}$ on which predictions are evaluated. Candidate and target nodes may overlap. Representations outside the candidate pool provide context throughout selection.

Selectable representation and reference input. Each candidate node contributes one text vector and one image vector. One selectable representation is a complete vector together with its node and modality identity. With modality set $\mathcal{M} = \{\text{text}, \text{image}\}$, the candidates form

$$\mathcal{E} = \dot{\bigcup_{m \in \mathcal{M}}} \mathcal{E}_m, \qquad |\mathcal{E}_m| = N \quad \text{for every } m \in \mathcal{M}. \tag{1}$$

Candidate e has an observed representation h_e and a reference r_e, defined as the mean representation over training nodes for the same modality. A selected set $S \subseteq \mathcal{E}$ determines the predictor input

$$H_S(e) = \begin{cases} h_e, & e \in S, \\ r_e, & e \in \mathcal{E} \setminus S. \end{cases} \tag{2}$$

where noncandidate entries of H_S keep their observed values. Write $Z_S = f_\theta(A, H_S)$ and $P_S = \mathrm{softmax}(Z_S)$. The reference input $S = \varnothing$ replaces all candidates by their modality means, whereas full candidate input $S = \mathcal{E}$ uses all observed representations. Before selection, we run the predictor on full candidate input to obtain the target classes.

Output and objective. Let $P_{\mathcal{E}}$ be the full-input probabilities and $\widehat{y}_t = \arg\max_c P_{\mathcal{E},tc}$ the predicted class for target $t \in \mathcal{T}$. We measure how well a selected subset supports these classes by

$$R(S) = -\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \log P_{S,t\widehat{y}_t}. \tag{3}$$

Minimizing $R(S)$ increases the geometric mean probability assigned to the original classes, including when the most likely class stays unchanged. It also bounds their disagreement rate over $\mathcal{T}$ by $d(S) \leq \min\{1, R(S)/\log 2\}$ (Appendix B.3). This connects a continuous selection objective to prediction preservation. A subset can assign the original class greater probability than full input, so the loss rewards class support rather than matching the entire probability vector.

Separate modality capacities. Let $\mathbf{K} = (K_{\text{text}}, K_{\text{image}})$ denote the available text and image budgets, with $0 \leq K_m \leq |\mathcal{E}_m|$. Each selected representation consumes one unit of its modality budget. The feasible family and selection objective are

$$S^\star \in \arg\min_{S \in \mathcal{F}_{\mathbf{K}}} R(S), \qquad \mathcal{F}_{\mathbf{K}} = \{S \subseteq \mathcal{E} \mid |S \cap \mathcal{E}_m| = K_m,\ \forall m \in \mathcal{M}\}. \tag{4}$$

The selector returns $S \in \mathcal{F}_{\mathbf{K}}$, and the predictor evaluates H_S. Thus, the budget counts selected candidate vectors per modality, while the rest of the graph input remains available.

3.2 Empirical Study

The joint selection problem leaves three practical choices. Should a candidate be scored with all other inputs present or with the subset currently selected? Should its score track the predicted class or its probability? Should the scores be reused as selection proceeds? We examine these choices in turn to identify what the search needs to measure.

Study design. We vary one scoring choice at a time under the same setup. The deletion probe compares 6,600 candidate representations from five graphs, three seeds, and 220 candidate nodes (Yan et al., 2025; Wan et al., 2026). Cross-entropy deletion (CE deletion) replaces a candidate by its modality mean at full input and measures the increase in R. Total variation deletion (TV deletion) measures half the summed absolute probability change across classes, averaged over targets. We compare both rankings with exact first-addition gains.

We also compare Kullback-Leibler divergence (KL) from the full-input probabilities and prediction agreement, the fraction of unchanged predicted classes. The initial-gain ranking scores candidates once at the reference input, whereas FORWARD recomputes gains after every addition. These comparisons add EleFashion (Zhu et al., 2025), giving six graphs and 54 graph, seed, and budget settings. Objective recovery measures the loss reduction from reference to full input, with 0% and 100% marking the two endpoints. Figure 2b pools losses over affected targets before forming this ratio, while tables average the individual ratios.

Deletion and addition favor different candidates. CE deletion identifies the best first addition in 19 of 30 graph, seed, and modality groups, compared with 13 for TV deletion (Fig. 2a). Their mean Spearman correlations with addition gains are 0.950 and 0.791. CE deletion therefore provides a stronger initial ranking, yet misses the best addition in 11 groups. The distinction is the surrounding input. Deletion measures what a representation contributes alongside all other candidates, whereas construction asks what it contributes to the subset being assembled. Appendix B.2 gives a counterexample for both deletion scores.

Predicted-class probability remains informative after hard agreement saturates. At 5%, 10%, and 20% capacity, predicted-class cross-entropy achieves pooled recovery of 15.7%, 26.1%, and 42.3%. KL divergence reaches 11.9%, 20.6%, and 36.1%, while prediction agreement reaches 8.3%, 13.7%, and 23.6%. The difficulty with hard agreement is visible during construction. It gives zero gain at 1,877 of 2,772 selected steps, compared with none for predicted-class cross-entropy. Agreement is already near saturation, averaging 99.85% for FORWARD and 99.95% for the agreement control over affected targets and budgets. Cross-entropy continues to reward additional probability assigned to the target class. KL divergence also remains smooth, but requires matching probabilities for the other classes. These results favor predicted-class cross-entropy for the selection task.

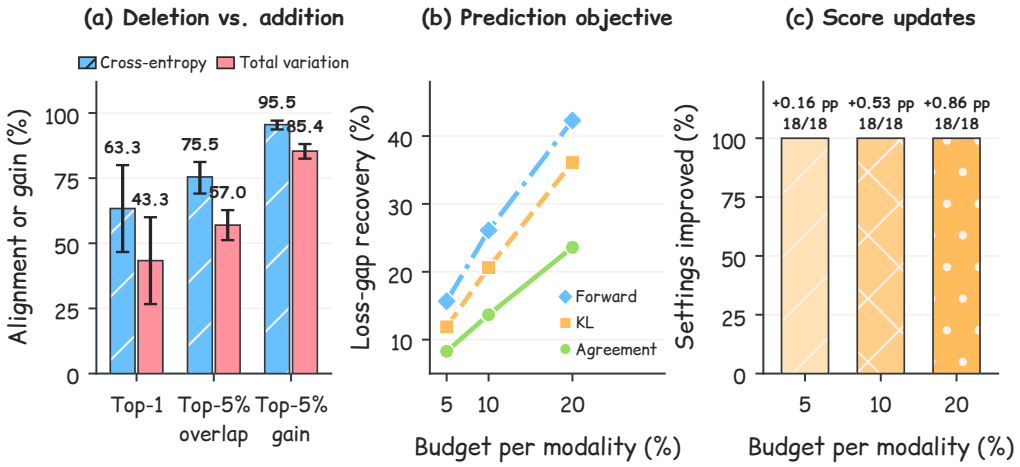

Figure 2: **Candidate scores depend on the input and prediction objective.** (a) Deletion rankings differ from first-addition rankings. (b) Predicted-class cross-entropy gives the highest recovery among the three objectives. (c) Updating addition scores lowers loss in all 18 settings at each capacity.

Updating scores improves selection consistently. Updating gains lowers the loss in all 54 comparisons with the initial-gain ranking (Fig. 2c). The mean recovery improvement grows from 0.2 percentage points at 5% capacity to 0.9 at 20%. An initial ranking therefore leaves improvements available as the subset grows. Rescoring captures these changes, while exchanges extend the same principle to a subset whose quotas are already filled.

Role of the graph. Removing feature and label propagation reduces mean full-input accuracy from 76.38% to 70.41%, with lower accuracy on five of six datasets. Propagation also expands a candidate representation's effect from its own node to an average of 127.2 target nodes. The graph thus supplies predictive context and determines which predictions an input change can affect. Appendix D.7 reports the controlled comparisons.

Design implication. The study identifies the selected subset as the relevant context for scoring. Predicted-class cross-entropy measures useful probability changes, and rescoring follows changes in that context. At full capacity, further improvement requires replacing an earlier choice. Jointly evaluating the removal and addition measures whether the replacement benefits the whole subset. These observations lead to the screened-exchange search in Sec. 4.

4 Methodology

4.1 Overview

GraphSelect evaluates each replacement in the subset where it will be used. Figure 3 summarizes the search. Individual gains initialize a subset that fills both quotas, current scores identify promising replacements, and joint evaluation selects an improving exchange. Accepted exchanges trigger new scores. Sec. 4.4 describes exact evaluation for additive graph predictors.

4.2 Initialization and Current State Screening

The gain from adding candidate e is $\Delta(e \mid S) = R(S) - R(S \cup \{e\})$. We initialize from these gains at the reference input,

$$S^{(0)} = \bigcup_{m \in \mathcal{M}} \mathrm{TopK}_{K_m}\{\Delta(e \mid \varnothing) \mid e \in \mathcal{E}_m\}. \tag{5}$$

where TopK_{K_m} returns the K_m highest-scoring candidate identities, with ties resolved by a predetermined order. Each candidate is scored once.

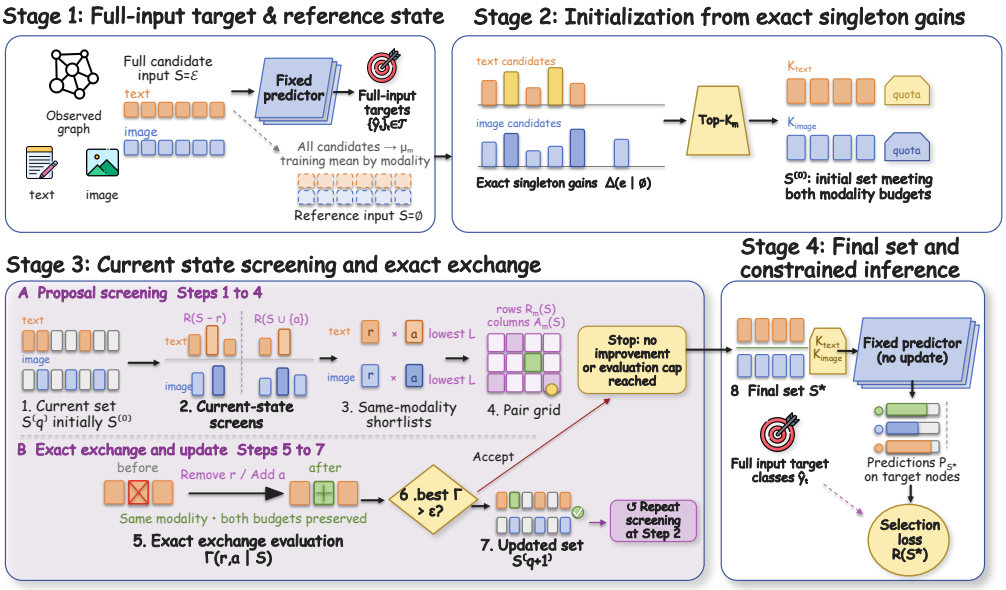

Figure 3: GRAPHSELECT overview. The graph predictor supplies the full-input target classes and evaluates each proposed subset. Singleton gains initialize selection, followed by screening and joint exchange evaluation. Accepted exchanges preserve both modality quotas and trigger score updates.

Screening identifies inputs the subset can give up and candidates it can use instead. For modality m, the removal shortlist $\mathcal{R}_m(S)$ contains up to L selected candidates with the lowest losses $R(S \setminus \{r\})$. The addition shortlist $\mathcal{A}_m(S)$ contains up to L unselected candidates with the lowest losses $R(S \cup \{a\})$. These rankings narrow the pairs that require joint evaluation.

4.3 EXACT SCREENED EXCHANGE

A useful addition can overlap with the contribution of the input being removed. GraphSelect therefore evaluates the two changes together. The candidate pairs form $\mathcal{P}(S) = \bigcup_{m \in \mathcal{M}} \mathcal{R}_m(S) \times \mathcal{A}_m(S)$, so each pair replaces one representation by another of the same modality. At iteration q, we choose

$$(r_q, a_q) \in \underset{(r,a) \in \mathcal{P}(S^{(q)})}{\arg\max} \left[R(S^{(q)}) - R\big((S^{(q)} \setminus \{r\}) \cup \{a\}\big) \right]. \tag{6}$$

Denote the bracketed loss reduction by $\Gamma(r, a \mid S^{(q)})$. If the best reduction exceeds ε, we update $S^{(q+1)} = (S^{(q)} \setminus \{r_q\}) \cup \{a_q\}$ and recompute both shortlists for the new subset. Otherwise, the search terminates.

Each accepted exchange preserves both quotas and strictly lowers the loss, so the returned subset improves on or matches initialization (Appendix B). We use $\varepsilon = 10^{-12}$ to reject numerical ties. In the additive comparison, $L \leq 8$ and the search receives $\lfloor 0.35 Q_{\text{forward}} \rfloor$ objective evaluations, including initialization. Here Q_{forward} is the count needed by Forward to fill both quotas. The search reduces L as this allowance is consumed and stops when it cannot evaluate another exchange.

4.4 EXACT EVALUATION FOR ADDITIVE PREDICTORS

For the additive graph predictor, let x_i^m be the log-probability vector for node i and modality m, with $U_i = \frac{1}{2} \sum_{m \in \mathcal{M}} x_i^m$. Row-normalized adjacency $\widetilde{A}$, smoothed label probabilities Q, term weights λ, and hop weights η give the logits

$$Z = \lambda_0 U + \lambda_g (\eta_1 \widetilde{A} U + \eta_2 \widetilde{A}^2 U) + \lambda_{\text{LP}} \log Q. \tag{7}$$

Let B_e be the logit change when candidate e replaces its reference. Then $Z_S = Z_\varnothing + \sum_{e \in S} B_e$. Cross-entropy couples these contributions, making candidate gains depend on the selected subset. Propagation determines which targets each B_e affects, extending to two hops here.

Both additions and exchanges can be evaluated through one logit change D. Let $\mathcal{T}_D$ contain its affected target rows and $\mathcal{L}_t(z) = -\log \mathrm{softmax}(z)_{\widehat{y}_t}$. The loss reduction is

$$g(D \mid S) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}_D} \left[\mathcal{L}_t(Z_{S,t}) - \mathcal{L}_t(Z_{S,t} + D_t) \right]. \tag{8}$$

Additions use $D = B_e$ and exchanges use $D = B_a - B_r$. Loss terms outside the affected rows cancel exactly, so this calculation avoids unchanged targets. Nonlinear predictors evaluate the same changes through complete forward passes. Appendix C details gain reuse and computational cost.

5 Experiments

We compare selection quality, examine exchanges and score updates, and measure prediction quality as the retained subset grows.

5.1 Experimental Settings

Datasets and protocol. The additive comparison uses six multimodal graph benchmarks (Yan et al., 2025; Wan et al., 2026; Zhu et al., 2025), three split seeds, $N = 220$ candidate nodes, and capacities b of 5%, 10%, and 20%, with $K_m = \lceil bN \rceil$. We also evaluate nine trained nonlinear architectures on Grocery and RedditS at 2.5%, 5%, 10%, 15%, 20%, and 30%. Test-set membership and test labels are unavailable to the selectors and used only for downstream evaluation.

Comparators. We adapt the six published methods in Table 1 to score the same candidate representations (Shapley, 1953; Kokhlikyan et al., 2020; Sundararajan et al., 2017; Ying et al., 2019; Lu et al., 2024; Funke et al., 2023). Each receives the same predictor, target nodes, and modality quotas. Sec. 3.2 defines the additional controls, while the Appendix explains the adaptation of each published method.

Metrics. We report selection loss $R(S)$ and objective recovery. Prediction agreement, total variation, accuracy, Macro-F1, and cross-entropy to ground-truth labels describe prediction changes. Tables give means and standard deviations. Hierarchical intervals resample datasets and then seeds.

5.2 Comparison Under the Shared Protocol

Table 1 compares objective recovery across the six graphs. Figure 4 shows the corresponding budget trends and differences from Forward.

Table 1: **Objective recovery (%).** Dataset columns summarize three seeds, and overall means summarize 18 graph and seed settings. Values are mean ± standard deviation. GraphSelect uses the budgeted search setting.

Method	Recovery at 20% capacity						Mean over 18 settings		
	SemArt	Grocery	Movies	Toys	RedditS	EleFashion	5%	10%	20%
Attribution and explanation adapters									
Shapley Sampling	44.7 ±1.2	37.3 ±1.5	53.7 ±1.1	36.1 ±0.2	32.1 ±1.3	47.5 ±4.2	14.3 ±3.4	24.9 ±5.1	41.9 ±7.7
Integrated Gradients	44.2 ±2.3	37.1 ±1.1	51.7 ±1.5	35.2 ±0.1	30.2 ±0.7	43.3 ±4.1	13.9 ±3.7	23.9 ±5.2	40.3 ±7.3
Feature Ablation	47.5 ±0.7	37.7 ±1.7	55.2 ±1.9	37.0 ±0.2	33.7 ±1.2	52.5 ±2.6	15.2 ±3.7	26.0 ±5.4	43.9 ±8.3
GNNExplainer	22.5 ±1.2	28.0 ±1.6	29.0 ±1.8	24.9 ±0.8	18.0 ±0.4	28.3 ±3.9	7.3 ±2.3	13.9 ±3.5	25.1 ±4.4
GOAt	39.7 ±2.0	37.6 ±1.6	57.4 ±1.9	36.7 ±0.3	29.8 ±1.3	38.3 ±4.9	13.1 ±4.2	22.9 ±6.2	39.9 ±8.8
Sufficient-subset adapter									
Zorro	43.7 ±1.5	31.2 ±1.8	44.1 ±3.2	28.6 ±0.7	31.5 ±0.3	44.2 ±2.6	13.9 ±3.7	23.0 ±5.5	37.2 ±7.1
Proposed method									
GraphSelect	**50.9** ±0.5	**42.2** ±2.3	**61.1** ±1.0	**40.5** ±0.4	**38.9** ±1.0	**56.1** ±1.8	**17.9** ±4.0	**29.9** ±6.1	**48.3** ±8.5

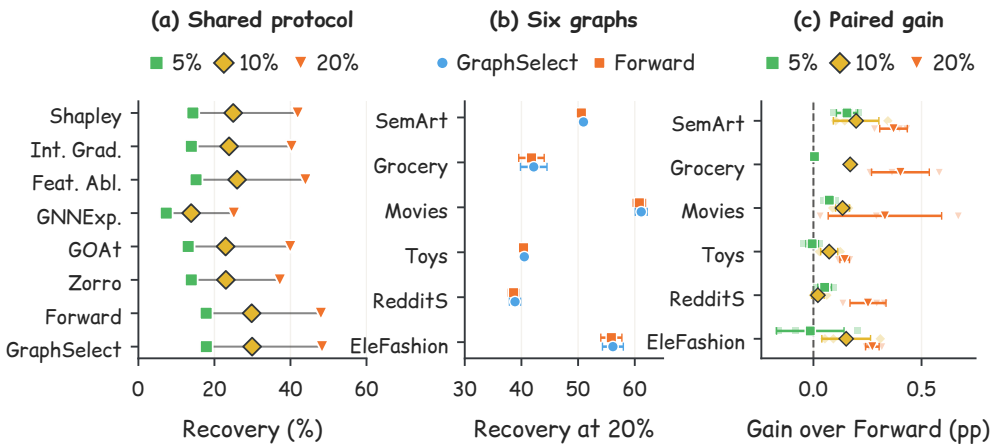

Figure 4: **Selection quality.** (a) Mean recovery at three capacities. (b) Dataset results at 20% capacity. (c) Paired recovery gains over Forward. All methods use the same candidate pool and modality quotas.

GraphSelect achieves the highest mean recovery. At 5%, 10%, and 20% capacity, GRAPHSELECT recovers 17.9%, 29.9%, and 48.3% of the objective gap on average. Feature Ablation, the strongest published comparator by mean recovery, reaches 15.2%, 26.0%, and 43.9%. The advantage holds across the three capacities and all six dataset means at 20%, showing that joint refinement improves on selecting candidates solely from attribution or explanation scores.

GraphSelect improves on the study controls. Compared with FORWARD, budgeted GRAPHSELECT gives lower loss in 46 of 54 settings, ties in five, and higher loss in three. Their mean recoveries reach 48.3% and 48.0% at 20% capacity. Refining the completed subset extends the improvements obtained by repeated addition rescoring. The initial-gain ranking and CE deletion also have lower mean recovery than GRAPHSELECT.

5.3 SEARCH ABLATIONS AND CAPACITY ANALYSIS

We tune the shortlist width on a predefined subset of Grocery settings and use the selected width in 126 disjoint evaluation settings. Appendix Table 7 reports the complete loss differences and win, tie, and loss counts.

We compare Forward with GraphSelect and its two ablations under matched evaluation budgets, treating loss differences within 10^{-6} as ties.

Exchange refinement and score updates improve selection. Removing exchange refinement increases the loss in 112 of 126 settings and ties in 14. Keeping the initial scores during exchange increases the loss in 62 settings, ties in 62, and improves it in two. Against Forward, GraphSelect improves 57 settings, ties in 68, and loses in one. The component controls show that both revising the initial subset and recomputing the screens contribute to selection quality. The Forward comparison further shows that revising earlier choices improves on repeated addition within the same evaluation budget.

Capacity and predictive performance. We vary the number of retained representations with the trained predictor and all other inputs unchanged. Accuracy and Macro-F1 show how selection affects test decisions, complementing probability recovery.

Larger subsets reduce the prediction gap. Across 54 architecture, graph, and seed settings, mean recovery rises from 37.6% at 5% capacity to 77.6% at 20%. At 20%, the mean accuracy and Macro-F1 drops are 0.100 and 0.065 percentage points. Recovery reaches 90.4% at 30%. Even at the smallest quota of six representations per modality, the mean accuracy gap is 0.237 percentage points. Table 2 reports all six capacities, and Appendix Figure 6 compares the four selectors.

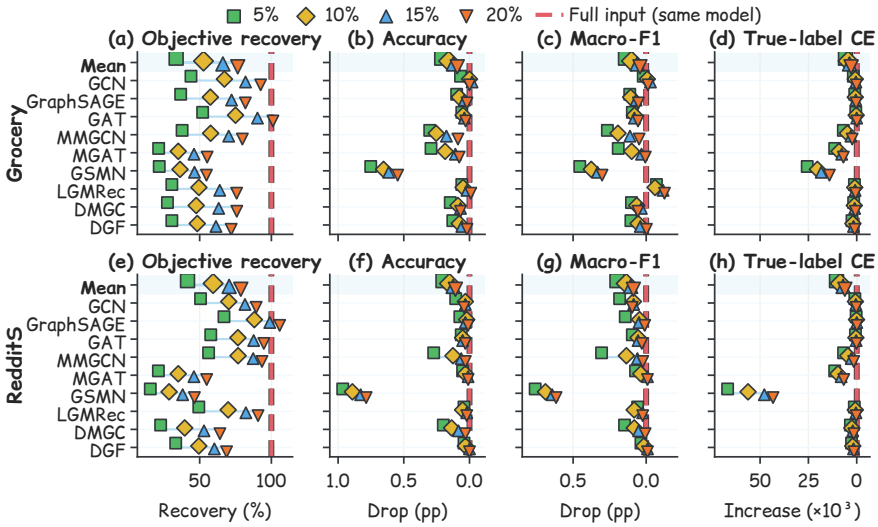

Figure 5: **Capacity and prediction quality.** Objective recovery and downstream performance gaps at four capacities on Grocery and RedditS. Red dashed lines indicate the corresponding full-input results.

Table 2: **GraphSelect across six capacities.** Values summarize 54 settings. Accuracy and Macro-F1 drops are in percentage points. The increase in cross-entropy to ground-truth labels is multiplied by 10^3.

Capacity (quota)	Objective recovery (%)	Accuracy drop	Macro-F1 drop	CE increase
2.5% (6)	24.81 ± 11.64	0.237 ± 0.270	0.202 ± 0.227	9.14 ± 16.80
5.0% (11)	37.56 ± 15.50	0.207 ± 0.256	0.171 ± 0.201	8.14 ± 15.62
10.0% (22)	56.02 ± 17.91	0.156 ± 0.237	0.119 ± 0.179	6.53 ± 13.12
15.0% (33)	68.38 ± 18.27	0.124 ± 0.225	0.084 ± 0.172	5.40 ± 11.31
20.0% (44)	77.61 ± 17.92	0.100 ± 0.212	0.065 ± 0.166	4.52 ± 10.15
30.0% (66)	90.39 ± 16.62	0.069 ± 0.176	0.040 ± 0.145	3.19 ± 7.50
Full input	**100.00** ± 0.00	**0.000** ± 0.000	**0.000** ± 0.000	**0.00** ± 0.00

At 20% capacity, the three graph architectures recover 91.0% to 97.9% of the objective gap, compared with 50.5% to 86.6% for the multimodal graph architectures. The mean accuracy gap is 0.667 percentage points for GSMN and 0.006 to 0.060 for the other eight architectures. These results separate two aspects of preservation. Changes in class probabilities need not change the most likely class, so a small accuracy gap can coexist with a larger objective gap. Increasing capacity improves probability recovery even when accuracy is already close to the full-input result. Appendix Table 6 reports the individual architecture results.

6 Conclusion

We presented GraphSelect for preserving multimodal graph predictions under separate text and image representation budgets. Its search follows the selected input, revising earlier choices through jointly evaluated exchanges and score updates. Graph propagation determines which target predictions a representation can support, and joint evaluation measures its contribution. GraphSelect achieves the highest mean objective recovery in the six-graph comparison. Across nine trained architectures, retaining 20% of the candidate representations per modality yields a mean accuracy gap of 0.10 percentage points to full input. These results show how joint selection preserves predictive performance with fewer candidate vectors, using the trained predictor and the remaining graph context.

AI use statement. Generative AI assisted with research framing, navigation of research materials, study design, code drafting, result interpretation, figure preparation, and language editing. The underlying data and the framework figure supplied by the authors were not altered by these tools. The authors verified the manuscript and take responsibility for its content.

Reproducibility statement. The Appendix provides the experimental protocol, implementation details, and proofs. The anonymous source package includes the selection methods, experiment configurations, individual results, and scripts for aggregation and figure generation.

REFERENCES

Ege Akkas and Ariful Azad. GNNShap: Scalable and accurate GNN explanation using shapley values. In *Proceedings of the ACM Web Conference*, pp. 827–838. ACM, 2024. doi: 10.1145/3589334.3645599. URL https://doi.org/10.1145/3589334.3645599.

Kenza Amara, Rex Ying, Zitao Zhang, Zhihao Han, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. GraphFramEx: Towards systematic evaluation of explainability methods for graph neural networks, 2022. URL https://arxiv.org/abs/2206.09677.

Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 567–576. PMLR, 2019. URL https://proceedings.mlr.press/v89/carter19a.html.

Runxiang Cheng, Gargi Balasubramaniam, Yifei He, Yao-Hung Hubert Tsai, and Han Zhao. Greedy modality selection via approximate submodular maximization. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 389–399. PMLR, 2022. URL https://proceedings.mlr.press/v180/cheng22a.html.

Yasha Ektefaie, George Dasoulas, Ayush Noori, Maha R. Farhat, and Marinka Zitnik. Multimodal learning with graphs. *Nature Machine Intelligence*, 5(4):340–350, 2023. doi: 10.1038/s42256-023-00624-6. URL https://doi.org/10.1038/s42256-023-00624-6.

Dongzhe Fan, Yi Fang, Jiajin Liu, Djellel Difallah, and Qiaoyu Tan. MLaGA: Multimodal large language and graph assistant, 2025. URL https://arxiv.org/abs/2506.02568.

Yi Fang, Bowen Jin, Jiacheng Shen, Sirui Ding, Qiaoyu Tan, and Jiawei Han. GraphGPT-O: Synergistic multimodal comprehension and generation on graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19467–19476, 2025. doi: 10.1109/CVPR52734.2025.01813. URL https://doi.org/10.1109/CVPR52734.2025.01813.

Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. Zorro: Valid, sparse, and stable explanations in graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8687–8698, 2023. doi: 10.1109/TKDE.2022.3201170. URL https://doi.org/10.1109/TKDE.2022.3201170.

Zhaochen Guo, Zhixiang Shen, Xuanting Xie, Liangjian Wen, and Zhao Kang. Disentangling homophily and heterophily in multimodal graph clustering, 2025. URL https://arxiv.org/abs/2507.15253.

Zhiqiang Guo, Jianjun Li, Guohui Li, Chaoyang Wang, Si Shi, and Bin Ruan. LGMRec: Local and global graph learning for multimodal recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8454–8462, 2024. doi: 10.1609/aaai.v38i8.28688. URL https://doi.org/10.1609/aaai.v38i8.28688.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1024–1034. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

Yufei He, Yuan Sui, Xiaoxin He, Yue Liu, Yifei Sun, and Bryan Hooi. UniGraph2: Learning a unified embedding space to bind multimodal graphs. In *Proceedings of the ACM Web Conference*, pp. 1759–1770. ACM, 2025. doi: 10.1145/3696410.3714818. URL https://doi.org/10.1145/3696410.3714818.

Jun Hu, Bryan Hooi, Bingsheng He, and Yinwei Wei. Modality-independent graph neural networks with global transformers for multimodal recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11790–11798, 2025. doi: 10.1609/aaai.v39i11.33283. URL https://doi.org/10.1609/aaai.v39i11.33283.

Jun Hu, Yufei He, Yuan Li, Bryan Hooi, and Bingsheng He. NTSFormer: A self-teaching graph transformer for multimodal isolated cold-start node classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(17):14856–14864, 2026. doi: 10.1609/aaai.v40i17.38506. URL https://doi.org/10.1609/aaai.v40i17.38506.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 2021. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, 2020. URL https://arxiv.org/abs/2009.07896.

Xunkai Li, Zhengyu Wu, Zekai Chen, Henan Sun, Daohan Su, Guang Zeng, Hongchao Qin, Rong-Hua Li, and Guoren Wang. LION: A clifford neural paradigm for multimodal-attributed graph learning, 2026. URL https://arxiv.org/abs/2601.21453.

Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10921–10930, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Liu_Graph_Structured_Network_for_Image-Text_Matching_CVPR_2020_paper.html.

Jiajin Liu, Dongzhe Fan, Jiacheng Shen, Chuanhao Ji, Daochen Zha, and Qiaoyu Tan. Graph-MLLM: Harnessing multimodal large language models for multimodal graph learning, 2025. URL https://arxiv.org/abs/2506.10282.

Shengyao Lu, Keith G. Mills, Jiao He, Bang Liu, and Di Niu. GOAt: Explaining graph neural networks via graph output attribution. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/92ee07d8a2c8f5ec08eff83f9eff0c1b-Abstract-Conference.html.

Xuying Ning, Dongqi Fu, Tianxin Wei, Wujiang Xu, and Jingrui He. Graph4MM: Weaving multimodal learning with structural information. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=FB2e8PV6qg.

Ciyuan Peng, Jiayuan He, and Feng Xia. Learning on multimodal graphs: A survey. *CoRR*, abs/2402.05322, 2024. URL https://arxiv.org/abs/2402.05322.

Tillmann Rheude, Roland Eils, and Benjamin Wild. Cohort-based active modality acquisition. *arXiv preprint arXiv:2505.16791*, 2025. URL https://arxiv.org/abs/2505.16791.

Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games (AM-28), Volume II*, pp. 307–318. Princeton University Press, 1953. doi: 10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Zhulin Tao, Yinwei Wei, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat-Seng Chua. MGAT: Multimodal graph attention network for recommendation. *Information Processing & Management*, 57(5):102277, 2020. doi: 10.1016/j.ipm.2020.102277. URL https://doi.org/10.1016/j.ipm.2020.102277.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Chenxi Wan, Xunkai Li, Yilong Zuo, Haokun Deng, Sihan Li, Bowen Fan, Hongchao Qin, Rong-Hua Li, and Guoren Wang. OpenMAG: A comprehensive benchmark for multimodal-attributed graph. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64650.

Jiachen T. Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. Rethinking data shapley for data selection tasks: Misleads and merits. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52033–52063. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24cg.html.

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

Hao Yan, Chaozhuo Li, Jun Yin, Zhigang Yu, Weihao Han, Mingzheng Li, Zhengxin Zeng, Hao Sun, and Senzhang Wang. When graph meets multimodal: Benchmarking and meditating on multimodal attributed graph learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 5842–5853. ACM, 2025. doi: 10.1145/3711896.3737404. URL https://doi.org/10.1145/3711896.3737404.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html.

Minji Yoon, Jing Yu Koh, Bryan Hooi, and Ruslan Salakhutdinov. Multimodal graph learning for generative tasks, 2023. URL https://arxiv.org/abs/2310.07478.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12241–12252. PMLR, 2021. URL https://proceedings.mlr.press/v139/yuan21c.html.

Sirui Zhang, Xu Wang, Zhengyu Wu, Xunkai Li, and Hongchao Qin. Context-aware modality-topology co-alignment for multimodal attributed graphs, 2026. URL https://arxiv.org/abs/2606.14172.

Haoran Zheng, Renchi Yang, Hongtao Wang, and Jianliang Xu. Cross-contrastive clustering for multimodal attributed graphs with dual graph filtering. In *KDD '26: Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 1, pp. 2020–2030. Association for Computing Machinery, 2026. doi: 10.1145/3770854.3780236. URL https://doi.org/10.1145/3770854.3780236.

Jing Zhu, Yuhang Zhou, Shengyi Qian, Zhongmou He, Tong Zhao, Neil Shah, and Danai Koutra. Mosaic of modalities: A comprehensive benchmark for multimodal graph learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14215–14224, 2025. doi: 10.1109/CVPR52734.2025.01326. URL https://doi.org/10.1109/CVPR52734.2025.01326.

Yilong Zuo, Xunkai Li, Zhihan Zhang, Ronghua Li, and Guoren Wang. RoleMAG: Learning neighbor roles in multimodal graphs, 2026. URL https://arxiv.org/abs/2604.12271.

This appendix extends the experiments, develops the theoretical results, and describes the implementation and evaluation of GRAPHSELECT. The additional comparisons examine how selection changes with the available capacity, predictor architecture, reference representation, and search procedure.

A ADDITIONAL EXPERIMENTS

A.1 COMPARISON METHODS AND CONTROLS

Table 3 summarizes the comparison methods and controls. Published attribution, explanation, and sufficient-subset methods provide alternative ways to score the candidate representations. The study controls test the input used for scoring and the prediction objective, while the method variants examine construction, exchange, and score updates.

Table 3: **Selection methods and controls.** The rows summarize how each method selects representations.

Entry	Operation in this paper
Published method adapters	
Shapley Sampling, Integrated Gradients, Feature Ablation, GNNExplainer, and GOAt	Published scoring or explanation methods adapted to rank the same candidate representations
Zorro	A published sufficient-subset method adapted to the same candidates and modality quotas
Study controls	
CE and TV deletion	Remove one candidate from the full input and measure cross entropy or total variation
KL divergence and prediction agreement	Build from the reference input by matching probabilities or predicted classes
Method variants	
Initial-gain ranking and Forward construction	Stop after the initial ranking or recompute exact addition gains after every selection
Exchange without score updates	Use the same initial set and evaluation cap as GRAPHSELECT while keeping reference gains unchanged
Proposed method	
GRAPHSELECT	Screen removals and additions at the current input and evaluate same-modality exchanges exactly

A.2 DELETION AND ADDITION ALIGNMENT

We compare deletion scores with exact first-addition gains for 6,600 candidate representations from five graphs, three split seeds, and two modalities. CE deletion identifies the best first addition in 19 of 30 graph, seed, and modality groups, compared with 13 for TV deletion. Their mean Spearman correlations with addition gains are 0.950 and 0.791. At the 5% ranking depth, the candidates selected by the two deletion rules recover 95.5% and 85.4% of the average gain obtained by the exact addition ranking. Thus, deletion scores retain useful information, but a candidate's value can change between full input and the reference input.

In the construction study, Forward achieves lower loss than the initial-gain ranking, KL divergence, prediction agreement, and both deletion controls in all 54 graph, seed, and budget settings. Table 4 reports the mean and standard deviation over the 18 graph and seed settings at each budget. These are means of individual recovery ratios. The pooled recovery in the main text is calculated after pooling the losses over affected target nodes, so the two summaries can differ.

Table 4: **Design controls.** Objective recovery is mean ± standard deviation.

Control	5%	10%	20%
Input state			
CE deletion	17.3 ± 3.9	28.9 ± 6.0	<u>47.0</u> ± 8.4
TV deletion	14.9 ± 3.4	25.1 ± 4.5	42.0 ± 6.4
Prediction target			
KL divergence	13.9 ± 3.2	23.9 ± 5.0	40.7 ± 6.6
Prediction agreement	9.9 ± 2.8	15.7 ± 4.2	25.3 ± 4.7
Score updates			
Initial-gain ranking	<u>17.5</u> ± 3.9	<u>29.1</u> ± 5.9	46.9 ± 8.2
FORWARD construction	**17.8** ± 4.0	**29.8** ± 6.0	**48.0** ± 8.5

A.3 COMPARISON WITH FULL INPUT AT SIX CAPACITIES

The capacity study evaluates GraphSelect, Forward construction, the initial-gain ranking, and CE deletion on Grocery and RedditS using nine trained architectures, three seeds, 220 candidate nodes,

and six quotas per modality. For each setting, the selectors share the predictor, candidates, data split, target nodes, and reference representations. We compare their predictions with full input on the same test nodes, giving 1,296 paired measurements.

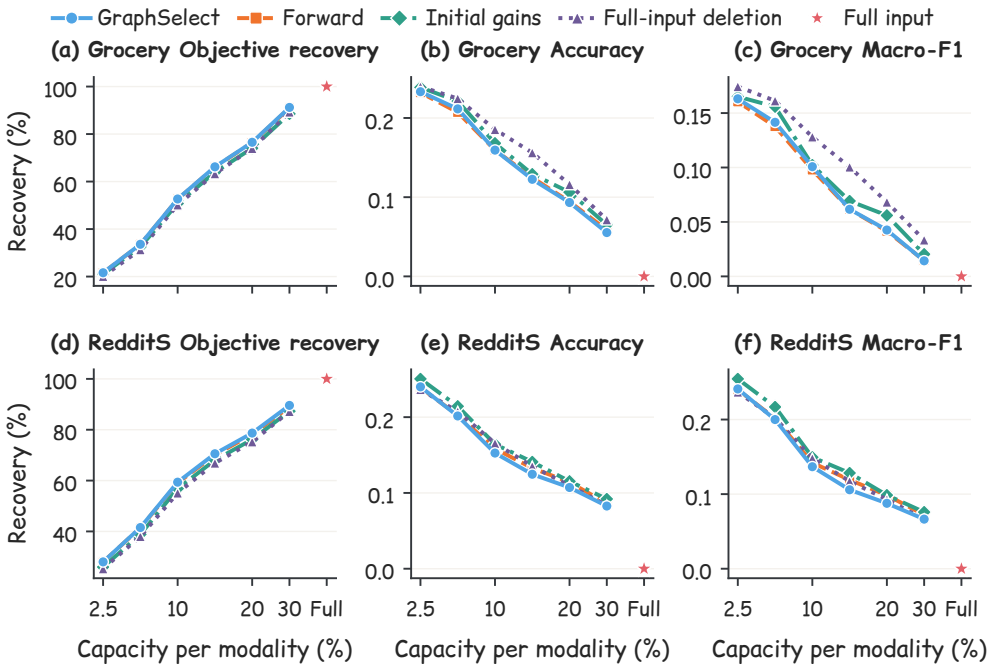

Figure 6: **Selection across capacities.** Mean objective recovery and downstream performance gaps relative to full input for four selectors on Grocery and RedditS at six capacities.

GraphSelect has the highest mean objective recovery at every capacity (Fig. 6). From 10% onward, it also has the smallest mean accuracy and Macro-F1 gaps. Forward is slightly closer to full input on these two downstream metrics at 2.5% and 5%. The difference shows that better recovery of predicted-class probability need not produce a closer match in classification performance. Table 2 reports the means and standard deviations.

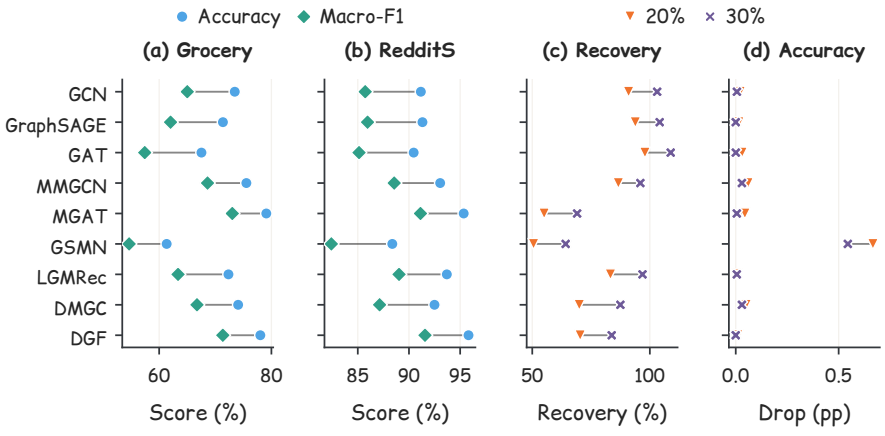

Figure 7: **Variation across architectures.** Full-input prediction quality and changes in recovery and accuracy gap from 20% to 30% capacity for nine architectures.

The architectures differ in both full-input prediction quality and their response to selection (Fig. 7). Increasing capacity from 20% to 30% brings most architectures closer to their full-input results.

Recovery varies more than the accuracy gap, partly reflecting differences in the reference-to-full-input loss reduction used to normalize recovery. We report every architecture evaluated in this comparison.

Tables 5 and 6 report the corresponding means and population standard deviations. MGAT has the highest full-input accuracy and Macro-F1 on Grocery, whereas DGF leads both metrics on RedditS. GAT has the highest recovery at 20% and 30%, but another architecture has the smallest accuracy gap. The architecture with the largest recovery therefore need not be the one with the closest downstream performance to full input.

Table 5: **Full input quality.** Values summarize three seeds for nine architectures.

Architecture	Grocery		RedditS	
	Accuracy	**Macro-F1**	**Accuracy**	**Macro-F1**
Graph architectures				
GCN	73.47 ± 2.15	65.01 ± 2.19	91.16 ± 1.00	85.73 ± 1.93
GraphSAGE	71.36 ± 2.23	62.03 ± 3.17	91.32 ± 0.41	85.96 ± 0.36
GAT	67.54 ± 2.17	57.43 ± 2.27	90.46 ± 0.32	85.12 ± 0.77
Multimodal graph architectures				
MMGCN	75.54 ± 1.99	68.62 ± 2.21	93.05 ± 1.05	88.56 ± 1.44
MGAT	**79.09** ± 0.30	**73.03** ± 0.42	95.34 ± 0.34	91.12 ± 0.88
GSMN	61.34 ± 0.55	54.65 ± 0.34	88.37 ± 0.44	82.42 ± 1.08
LGMRec	72.35 ± 0.79	63.37 ± 0.74	93.70 ± 0.10	89.03 ± 0.26
DMGC	74.08 ± 1.15	66.74 ± 1.49	92.49 ± 0.48	87.13 ± 0.62
DGF	78.03 ± 0.39	71.34 ± 0.98	**95.82** ± 0.33	**91.57** ± 1.04

Table 6: **Architecture details.** Values summarize two graphs and three seeds.

Architecture	20% capacity		30% capacity	
	Recovery	**Accuracy drop**	**Recovery**	**Accuracy drop**
Graph architectures				
GCN	91.0 ± 4.7	0.021 ± 0.034	103.1 ± 5.8	0.006 ± 0.028
GraphSAGE	93.8 ± 15.8	0.015 ± 0.029	104.2 ± 14.4	**0.000** ± 0.018
GAT	**97.9** ± 9.4	0.030 ± 0.043	**108.9** ± 10.7	**0.000** ± 0.046
Multimodal graph architectures				
MMGCN	86.6 ± 7.4	0.060 ± 0.061	96.0 ± 4.1	0.030 ± 0.038
MGAT	55.1 ± 3.5	0.044 ± 0.066	69.1 ± 3.3	0.005 ± 0.028
GSMN	50.5 ± 4.4	0.667 ± 0.145	64.2 ± 4.2	0.545 ± 0.121
LGMRec	83.2 ± 7.7	**0.006** ± 0.068	96.9 ± 4.6	0.005 ± 0.033
DMGC	70.0 ± 7.4	0.050 ± 0.032	87.5 ± 9.5	0.030 ± 0.034
DGF	70.4 ± 3.0	0.010 ± 0.040	83.8 ± 3.8	**0.000** ± 0.034

A.4 Reference and Search Controls

We repeat the comparison with four reference rules. They are the training mean, training median, nearest training representation to the mean, and zero vector. GraphSelect achieves lower mean loss than the initial-gain ranking under every rule. The mean Jaccard overlap with the training mean selection is 0.879 for the median, 0.348 for the nearest representation, and 0.538 for zero. The method ordering remains stable even though the selected sets vary across references.

The nonlinear search comparison uses 126 settings from the nine architectures, excluding the settings used to tune the shortlist width. GraphSelect has W/T/L counts of 57/68/1 against Forward construction and 112/14/0 against the version without exchange refinement. It uses 43.60% of the full forward passes available under the matched evaluation budget. Exchange without score updates starts from the same initial set and receives the same evaluation budget per setting, but uses the initial gains throughout refinement. GraphSelect has W/T/L counts of 62/62/2 against this control. Recomputing the scores improves 62 settings and preserves the loss within the tie tolerance in another 62. Table 7 gives the mean loss differences and their variation across settings.

Table 7: **Search ablations over 126 settings excluded from tuning.**

Variant	Modification	$R_{\text{variant}} - R_{\text{GraphSelect}}$ Loss $\times 10^4$	GraphSelect W/T/L
Construction alternative			
Forward construction	Repeated additions with rescoring	0.81 ± 3.75	57/68/1
Component removals			
w/o exchange refinement	Return the initial set	4.60 ± 10.67	112/14/0
w/o current-state updates	Keep initial scores during exchange	0.76 ± 3.85	62/62/2

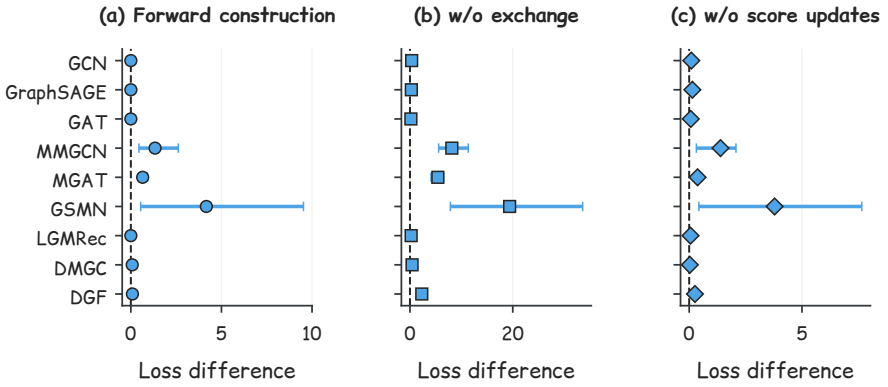

Figure 8: **Search ablations across architectures.** Loss differences between GraphSelect and its search controls, scaled by 10^4. Positive values favor GraphSelect.

Figure 8 shows how these differences vary across architectures. Score updates give the largest gains for MMGCN, MGAT, and GSMN, while several other architectures remain near a tie. A batch exchange control on the additive predictor provides a further comparison under the same evaluation budget. It improves several individual settings at 5% and 10% capacity. At 20%, screened exchange is better in 13 settings, tied in 5, and worse in none. Thus, the benefit of screening also depends on the capacity.

A.5 Algorithm Details

The larger search uses $L \leq \lceil \sqrt{N} \rceil$, giving at most 15 candidates per shortlist for $N = 220$.

GraphSelect takes a trained predictor, full-input target classes, candidate representations, modality references, quotas, shortlist width, objective tolerance, and evaluation budget. It returns a subset that satisfies both modality quotas. Initialization proceeds as follows.

1. Build the reference input and evaluate its loss.
2. Add each candidate alone and compute its exact gain.
3. Rank candidates within each modality using the predetermined tie order.
4. Select the first K_m candidates from every modality.
5. Evaluate the loss of the initialized subset.

The exchange stage repeats the following steps until it stops.

1. Evaluate the loss after removing each selected candidate.
2. Evaluate the loss after adding each unselected candidate.
3. Keep at most L candidates from each ranking within each modality.
4. Evaluate every feasible exchange between the resulting shortlists.

5. Accept the best exchange only when its gain exceeds ε.
6. Recompute both rankings after an accepted exchange or stop if no pair improves the loss.

Initialization, screening, and joint exchange evaluations all count toward the evaluation budget. A deterministic candidate order resolves exact ties. After each accepted exchange, the implementation evaluates the selected input again to verify the incremental update.

A.6 Exact Optima and Larger Candidate Pools

The exact oracle comparison uses five predefined eight-node pools for every graph and seed. At quotas one, two, and three per modality, Forward obtains 99.44%, 99.77%, and 99.46% of the reference-to-oracle improvement on average. Exact set matches occur in 88.9%, 91.1%, and 74.4% of the settings. These small pools allow comparison with the exact optimum and show that Forward often finds a nearly optimal subset. An exact optimum is not available for the larger $N = 220$ candidate pools.

With 1,000 candidate nodes and quotas of 50, 100, and 200 per modality, Forward achieves lower loss than the initial-gain ranking in all 54 settings. Reusing gains whose affected target rows have not changed reduces the number of exact marginal evaluations by 85.75% on average. This reduction measures the work avoided by sparse evaluation. End-to-end runtime was not measured in this test.

Direct recomputation of the reference logits, accepted exchanges, final subset, and full-input predictions agrees with the incremental implementation within the numerical tolerance. The nonlinear implementation uses a full forward pass for each objective evaluation.

B Theory

B.1 Task and Search Rule

Let P_S be the predictor probabilities when selected representations use their observed values and all other candidates use modality references. Let $\widehat{y}_t$ be the class predicted from the complete candidate pool. The selection loss is

$$R(S) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} - \log P_{S,t\widehat{y}_t}. \tag{9}$$

The constrained task keeps an exact quota for every modality.

$$S^\star \in \underset{S \subseteq \mathcal{E},\ |S \cap \mathcal{E}_m| = K_m\ \forall m}{\arg\min} R(S). \tag{10}$$

GraphSelect evaluates a same-modality exchange by jointly removing and adding a representation.

$$\Gamma(r, a \mid S) = R(S) - R\big((S \setminus \{r\}) \cup \{a\}\big). \tag{11}$$

The full-input predicted classes are determined before selection and supply the targets for R. Ground-truth labels and test membership are reserved for evaluation. Each exchange replaces a representation within the same modality.

Proposition 1. Every subset accepted by GraphSelect satisfies the modality quotas. Every accepted move strictly lowers R. The algorithm terminates under a finite objective evaluation cap.

Proof. The initial set contains exactly K_m representations from every modality. Each move removes and adds one representation from the same modality, so the counts remain unchanged. The acceptance rule requires $\Gamma(r, a \mid S) > \varepsilon$, which gives $R(S') < R(S)$. Every screen and exchange consumes a positive integer number of objective evaluations. A finite budget therefore permits only finitely many trials. □

The proposition establishes feasibility, loss reduction after each accepted exchange, and termination. Because screening can omit an improving exchange, it guarantees neither local nor global optimality.

B.2 A Deletion Ranking Counterexample

The direction of a transition can change a candidate ranking even without graph propagation. Consider a smooth binary predictor. Let $\mathcal{E} = \{a, b, c\}$, let $x_e \in [0, 1]$, and choose $0 < \varepsilon < 1$ and $0 < \delta < 1/2$. Define

$$\begin{aligned} q(x) &= x_a + x_b - x_a x_b - \varepsilon x_a x_b (1 - x_c), \\ p_x &= \big(\delta + (1 - 2\delta) q(x),\ 1 - \delta - (1 - 2\delta) q(x)\big). \end{aligned} \tag{12}$$

The full-input deletion score measures the total variation distance caused by removing a candidate.

$$s_{\mathrm{del}}(e) = \mathrm{TV}\big(p_{(1,1,1)}, p_{(1,1,1)-\mathbf{e}_e}\big). \tag{13}$$

The polynomial satisfies $0 \le q(x) \le 1$, so the probabilities are valid and bounded away from zero. At the complete input, deleting a or b leaves $q = 1$, while deleting c gives $q = 1 - \varepsilon$. The total variation rule therefore chooses c. From the reference input, adding c gives no loss reduction, while adding a or b gives

$$\Delta(a \mid \varnothing) = \Delta(b \mid \varnothing) = \log \frac{1 - \delta}{\delta} > 0. \tag{14}$$

Cross-entropy deletion makes the same choice. Removing a or b leaves the full-input class probability unchanged, while removing c lowers it. Both deletion rankings therefore choose a candidate with zero addition gain when the best addition has positive gain. This example isolates why the scoring input matters for subset construction.

B.3 Prediction Loss, Confidence, and Disagreement

For a sequence $S_0, S_1, \ldots, S_J$, write $p_t^j = P_{S_j, t\widehat{y}_t}$. The finite gain of one transition is

$$\delta_j = R(S_{j-1}) - R(S_j) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \log \frac{p_t^j}{p_t^{j-1}}. \tag{15}$$

Summing the transition gains cancels every intermediate loss.

$$\sum_{j=1}^{J} \delta_j = R(S_0) - R(S_J). \tag{16}$$

The loss is also the negative logarithm of the geometric mean probability assigned to the full-input predicted classes.

$$G(S) = \left(\prod_{t \in \mathcal{T}} P_{S, t\widehat{y}_t} \right)^{1/|\mathcal{T}|} = \exp[-R(S)]. \tag{17}$$

The identities apply to additions, removals, and exchanges.

For any threshold $\tau \in (0, 1)$, define the low-confidence rate $v_\tau(S) = |\mathcal{T}|^{-1} \sum_t \mathbf{1}[p_t^S \le \tau]$. Let $d(S)$ be the disagreement rate with the complete input classes. Then

$$v_\tau(S) \le \min\left\{1, \frac{R(S)}{-\log \tau}\right\}. \tag{18}$$

Disagreement with a full-input class requires its probability under the selected input to be at most one half.

$$d(S) \le v_{1/2}(S) \le \min\left\{1, \frac{R(S)}{\log 2}\right\}. \tag{19}$$

The selected-input error is therefore bounded by the full-input error plus the disagreement rate.

$$\mathrm{Err}(S) \le \mathrm{Err}_{\mathrm{full}} + d(S). \tag{20}$$

The first inequality follows because every term with $p_t^S \le \tau$ contributes at least $-\log \tau$ to the loss. A class disagreement implies $p_t^S \le 1/2$, which gives the second line. The last line follows from the pointwise union of full-input error and disagreement. These bounds relate the selection loss to confidence and prediction error over the declared target set. The bounds can be loose, particularly when $R(S) \ge \log 2$. They justify the objective as a surrogate for class preservation, while empirical agreement and classification metrics measure the observed prediction changes.

Table 8: **Multimodal graph datasets.** Numbers of nodes, nonzero adjacency entries, and classes in the processed graphs used for evaluation.

Dataset	Graph size		Classes
	Nodes	Adjacency nonzeros	
SemArt	21,382	1,173,014	10
Grocery	17,074	142,262	20
Movies	16,672	160,802	20
Toys	20,695	113,402	18
RedditS	15,894	283,080	20
EleFashion	97,766	399,172	11

C Exact Sparse Evaluation

For the additive predictor, each candidate induces a sparse change B_e in the logits of the target nodes. Its affected rows are $\mathcal{T}_e = \{t \in \mathcal{T} \mid B_{e,t} \neq 0\}$. The selected state is

$$Z_S = Z_\varnothing + \sum_{e \in S} B_e. \tag{21}$$

Addition and exchange gains can then be evaluated using only the affected rows.

$$\begin{aligned} \Delta(e \mid S) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}_e} \big[\ell_t(Z_{S,t}) - \ell_t(Z_{S,t} + B_{e,t})\big], \\ \Gamma(r, a \mid S) &= \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}_r \cup \mathcal{T}_a} \big[\ell_t(Z_{S,t}) - \ell_t(Z_{S,t} - B_{r,t} + B_{a,t})\big]. \end{aligned} \tag{22}$$

Rows outside these sets cancel exactly. An inverted index maps target rows to the candidates that affect them. An accepted update leaves a candidate's gain unchanged when their affected rows are disjoint, allowing that gain to be reused. We verify reused values by directly evaluating the objective.

Let $s_e = |\mathcal{T}_e|$ and let C be the class count. One candidate addition costs $O(Cs_e)$ after the current logits are available. One exchange costs $O(C|\mathcal{T}_r \cup \mathcal{T}_a|)$. These are arithmetic counts for the additive model. For a nonlinear predictor, each objective evaluation requires a full forward pass. We therefore distinguish objective evaluations, sparse row evaluations, and full forward passes when reporting computational cost.

C.1 Implementation and Computational Cost

The sparse implementation caches three quantities. It stores each contribution B_e on its affected rows, the current logits, and candidate gains with their dependent rows. After an accepted update, every gain that depends on a changed row must be recomputed before reuse. This dependence determines which gains can be reused without changing the result.

The nonlinear implementation constructs each candidate input, runs the trained predictor, and evaluates the loss on the target nodes. Initialization and screening count toward the budget of full forward passes. Loading the model and encoding the features are common to all selectors and are excluded from this count.

Both implementations follow the same selection rule. The additive structure allows gains to be evaluated from sparse changes in logits, whereas the nonlinear models require full forward passes. We evaluate selection quality for both and report the corresponding computational cost separately.

D Experimental Protocol

D.1 Datasets and Splits

We use the prescribed transductive graphs and three split seeds. Training labels are used to fit the predictor, while training-node representations define the modality references. Validation settings are

used to tune the search parameters. Test membership and labels are withheld during selection and used only to evaluate the resulting predictions. Candidate nodes follow a deterministic order. The main additive and nonlinear studies use the first 220 nodes, while the larger-scale comparison uses all 1,000 candidate nodes.

Each candidate node contributes one text representation and one image representation. The 220-node setting therefore has 440 selectable units. The 5%, 10%, and 20% budgets select 11, 22, and 44 units per modality. The full capacity sweep adds quotas 6, 33, and 66 for 2.5%, 15%, and 30%.

D.2 Predictors and References

The additive predictor combines modality logits, normalized one-hop and two-hop propagation, and smoothed label propagation. Its parameters do not change during selection. The nonlinear study uses GCN, GraphSAGE, GAT, MMGCN, MGAT, GSMN, LGMRec, DMGC, and DGF (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Wei et al., 2019; Tao et al., 2020; Liu et al., 2020; Guo et al., 2024; 2025; Zheng et al., 2026). Each architecture uses the stored text and image encodings. We evaluate all trained models without retraining them or choosing checkpoints from the subset results. Model evaluation is deterministic.

The main reference for each modality is the mean training representation. The reference comparison also uses the training median, the nearest training vector to the mean, and the zero vector. Each reference is determined before candidate scoring and remains unchanged throughout selection.

D.3 Selection and Evaluation Procedure

For each graph and seed, selection follows model training and precedes test evaluation.

1. Fit or load the predictor using the prescribed training and validation data.
2. Set the trained predictor to deterministic evaluation mode.
3. Prepare the candidate order and modality references.
4. Evaluate the predictor on full candidate input to determine the target classes.
5. Run each selector without test membership or test labels.
6. Evaluate the selected inputs on test nodes.

The complete candidate input is available before selection to determine the target classes. The representation quotas apply to the selected subset. Selection uses the model's predicted-class probabilities, and ground-truth test labels enter only the subsequent evaluation. It must be repeated if the predictor, candidate pool, reference, target set, or budget changes.

We tune the nonlinear shortlist width on a predefined subset of Grocery settings. The remaining Grocery settings and all RedditS settings provide a disjoint evaluation of the selected width. The downstream comparison then applies the same rule at all six capacities and compares the predictions with full input.

D.4 Comparators

We adapt Shapley Sampling, Integrated Gradients, Feature Ablation, GNNExplainer, GOAt, and Zorro to select whole representations. Each method uses the same predictor, candidate pool, full-input target classes, and separate modality quotas. Shapley Sampling uses a predefined sample budget. Integrated Gradients follows the straight path from the modality reference to the observed representation. Feature Ablation replaces one complete representation group. GNNExplainer learns a source-node mask and maps it to the two representation groups. GOAt aggregates its attribution score over the target rows. Zorro uses a predefined search width and fresh randomized complements to estimate its class-retention score.

The study controls change one selection choice at a time. CE deletion and TV deletion score a removal from full input. The initial-gain ranking scores each candidate once at the reference input. KL divergence compares the full class probability vector, while prediction agreement considers only whether the predicted class changes. Forward construction recomputes exact addition gains after

Table 9: Full-input test accuracy in the graph ablation study. Entries are mean ± population standard deviation over three seeds, in percent. The last column gives the mean number of target nodes affected by one candidate representation in the full graph predictor. Both ablations reduce this number to one.

Dataset	Full graph	No candidate propagation	No graph terms	Affected targets
SemArt	84.11 ± 0.09	84.09 ± 0.11	83.42 ± 0.91	277.3
Grocery	75.01 ± 0.61	75.01 ± 0.59	60.56 ± 5.47	183.9
Movies	47.55 ± 1.26	47.56 ± 1.24	38.33 ± 0.43	108.7
Toys	76.85 ± 0.64	76.86 ± 0.61	67.42 ± 0.97	55.4
RedditS	94.69 ± 0.30	94.67 ± 0.28	90.53 ± 0.36	17.4
EleFashion	80.06 ± 0.38	80.06 ± 0.38	82.21 ± 0.25	120.6
Mean	76.38	76.38	70.41	127.2

every selected representation. Exchange without score updates and GraphSelect start from the same initial set and receive the same budget of full forward passes.

D.5 Metrics and Statistical Units

The primary objective is the cross-entropy $R(S)$ to the full-input predicted classes. Objective recovery divides the loss reduction from the reference input by the reduction obtained with full input. Recovery can exceed 100% when a subset assigns greater probability to those classes than full input does, because $R(S)$ is not a distance between probability distributions. Accuracy, Macro-F1, cross-entropy to ground-truth labels, prediction agreement, and total variation describe other changes in the predictions. Comparisons with full input use the same trained model and target nodes.

Tables report mean ± population standard deviation whenever repeated settings form the displayed unit. Main additive tables use 18 graph and seed settings at each budget. The nonlinear tables use 54 graph, architecture, and seed settings at each budget. Hierarchical uncertainty intervals resample graphs and then seeds. Paired W/T/L counts compare losses within each setting, using the stated numerical tolerance to identify ties.

D.6 Reproducibility Details

The source package includes experiment configurations, commands, software environment details, individual results, and aggregation scripts. The scripts require one result for each expected setting and pair methods by predictor, candidate order, data split, and evaluation set. Figures and tables are generated from these paired results.

D.7 Graph Ablation Study

We separate the contribution of graph information to prediction from the effect of propagating candidate representations. The study uses the six additive datasets, three split seeds, 220 candidate nodes, and 5%, 10%, and 20% capacity per modality. We compare three predictor conditions. The full graph condition retains all terms. The second condition removes propagation of each candidate's deviation from its modality mean, retaining its within-node term and the original reference logits. Noncandidate graph context and label propagation remain available. The third condition removes both feature propagation and label propagation, retaining the original weight on the within-node modality logits.

Each condition defines its target classes using its own full candidate input. The encodings, probe training procedure, splits, and candidate sets are shared across conditions. The modality probes are reconstructed from the same recipe and seeds. This reconstruction changes no full-input predicted classes relative to the original saved probabilities. The within-node predictor uses the original parameters without additional tuning. These controlled ablations examine the additive model evaluated here.

Table 10: Selection losses in the graph ablation study. Each entry averages 18 graph and seed settings. The first two losses use each condition's own full-input target classes. The two rightmost columns report the control loss minus the GRAPHSELECT loss, multiplied by 10^5. Positive values favor GRAPHSELECT.

Predictor	Capacity	$R(\varnothing)$	$R(\mathcal{E})$	Initial ranking minus GRAPHSELECT	FORWARD minus GRAPHSELECT
Full graph	5%	0.522555	0.516625	1.599	0.352
	10%	0.522555	0.516625	4.026	0.776
	20%	0.522555	0.516625	6.567	1.756
No candidate propagation	5%	0.522529	0.517774	2.317	0.557
	10%	0.522529	0.517774	3.728	0.983
	20%	0.522529	0.517774	5.499	1.254
No graph terms	5%	1.555942	1.544582	1.185	0.268
	10%	1.555942	1.544582	3.055	0.693
	20%	1.555942	1.544582	6.140	2.535

Table 9 shows that graph terms improve mean full-input test accuracy by 5.97 percentage points. They improve accuracy on five datasets. On EleFashion, removing graph terms instead raises accuracy from 80.06% to 82.21%. Removing only candidate propagation leaves most of the graph context available, and the corresponding mean accuracy changes by less than 0.01 percentage points. Propagation also changes the reach of each candidate representation. Its mean number of affected targets ranges from 17.4 on RedditS to 277.3 on SemArt, compared with one in either ablation. Different candidate nodes share affected targets in 12.1% of pairs on average with propagation and in none of the pairs after it is removed.

For selection, FORWARD repeatedly evaluates all feasible additions until the quotas are filled. GRAPHSELECT uses a shortlist of at most eight candidates per modality and 35% of the analytical upper bound on these FORWARD evaluations. The initial-gain ranking evaluates candidates once. Each method keeps the same evaluation-limit rule across all three predictor conditions. This diagnostic uses a common analytical limit to compare conditions, while the main selection tables use their original matched FORWARD evaluation counts.

Table 10 reports the reference and full-input losses alongside raw selection-loss improvements. These losses are averaged over all target nodes. Recovery ratios have different denominators and target classes across conditions. Within the full graph condition, GRAPHSELECT improves on the initial-gain ranking in all 54 settings and on FORWARD in 47, with six ties and one loss. Without candidate propagation, the corresponding W/T/L counts are 53/1/0 and 47/6/1. Without either graph term, they are 51/3/0 and 45/8/1. Ties use a tolerance of 10^{-12}. Exchanges therefore improve input combinations in all three conditions. Graph propagation determines their cross-node effects, while modality fusion and the prediction loss also make a representation's gain depend on the selected subset.

Illustrative exchange. Figure 1 uses Grocery seed 44 at 20% capacity. The target is node 13190, which is outside the candidate pool. The neighboring jicama is node 13931, connected by the recorded also-viewed relation. The fern is node 8249, outside the target's two-hop neighborhood. The third accepted exchange replaces the fern image with the neighboring jicama image in the selected set, preserving 44 text and 44 image representations. Direct replay gives target-class probabilities of 47.5964% before and 53.0059% after, compared with 53.4605% under full input. The predicted category remains Produce throughout. Mean selection loss over all targets falls from 0.7554431450 to 0.7554343851. At the reference input, the fern and neighboring jicama images have addition gains of 5.0598×10^{-5} and 4.3646×10^{-5}, respectively. Immediately before the exchange, removing the fern image increases loss by 3.4887×10^{-5}, while adding the neighboring jicama image reduces it by 4.3646×10^{-5}. The figure shows the original product photographs and unchanged product identities. The neighbor's text uses its modality mean in both limited-input states, while the fern's text and both target inputs remain observed.

D.8 Scope and Limitations

Objective recovery measures improvement in the selection loss, while accuracy and Macro-F1 measure classification performance on test nodes. The architecture results show why both are needed. A high recovery ratio need not imply the smallest accuracy gap, and an aggregate mean can conceal this variation.

The evaluated workflow uses complete candidate representations during selection and a capacity-limited subset during subsequent inference. Each selection is associated with a trained predictor, candidate pool, reference, target set, and budget. A process that needs only previously computed predictions can cache those outputs directly. Extending the analysis to storage bytes, latency, or energy requires weighted costs and system measurements. Reconstruction of missing modalities and transfer to future models involve additional modeling choices.

The graph ablation study examines the additive predictor. Extending it to the nine nonlinear architectures would establish how architecture changes the relationship between graph propagation and selection quality.

The theory guarantees quota feasibility, lower loss after an accepted exchange, and termination under a finite evaluation budget. Screening can still miss an improving exchange. The downstream experiments measure gaps to full input and do not establish statistical equivalence. They support selection quality for the studied workflow, while global optimality and deployment savings require further investigation.