SMaRS-GNN: A Scalable and Robust Graph Neural Network Integrating Higher-Order Information

Anonymous authors
Paper under double-blind review

Abstract

Growing concerns about the computational and energy demands of artificial intelligence motivate more efficient learning methods. For graph classification, the challenge is to reduce computational cost while retaining information across higher-order neighborhoods and maintaining robustness to perturbations. In this paper, we introduce a new method called SMaRS-GNN to address this problem. Under this method, a graph representation is constructed for each actor from the dominant right singular vector of its node embedding matrix, and a critic model provides weights that combine these representations for classification. The training step uses randomly sampled subgraphs instead of the full graph, which leads to computational scalability. We establish finite sample bounds on the perturbations in normalized feature Gram matrices caused by subsampling, accounting for node sampling and recomputed message passing. Under standard regularity conditions, we establish theoretical bounds that yield stability guarantees for pooled representations and predicted labels. We further prove a risk transfer bound that preserves generalization guarantees up to explicit subsampling and optimization terms, as well as a bound that establishes actor-level robustness to feature perturbations. Experiments on simulated and benchmark datasets demonstrate substantial reductions in runtime and characterize the tradeoffs among computational efficiency, predictive accuracy, and robustness across various model settings and algorithmic hyperparameters.

1 Introduction

Growing concerns about the computational and energy demands of artificial intelligence have made efficiency an increasingly important objective in machine learning. Training and tuning modern models can require substantial computational resources, with associated financial and environmental costs (Strubell et al., 2019). The Green AI perspective therefore advocates evaluating efficiency alongside predictive performance (Schwartz et al., 2019). These concerns also have a broader infrastructure dimension. The International Energy Agency identifies AI as an important driver of growing electricity demand from data centers and highlights the potential contribution of more efficient hardware and models (International Energy Agency, 2025). These developments motivate learning methods that use available computation more effectively.

For graph learning, this objective creates a specific methodological challenge. A labeled observation can contain thousands of vertices and many more edges, and extracting information from increasingly distant neighborhoods requires repeated propagation over that structure. The resulting cost is incurred across training graphs, optimization iterations, and model-selection runs. Reducing the graph processed during training can therefore improve scalability and support the broader goal of resource-efficient AI. The statistical question is how much graph information can be removed without substantially changing the learned representation or its predictive usefulness.

Graph classification requires learning a label for an entire network, such as a molecular structure, protein interaction network, or social discussion graph. Graph neural networks (GNNs) combine node features with topology through message passing and then pool the node embeddings into a

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.

graph representation (Scarselli et al., 2009; Kipf & Welling, 2017; Gilmer et al., 2017; Xu et al., 2019). Here, three issues arise together:

1. First, the information relevant to graph classification may occur at different distances within a graph. Local neighborhoods capture interactions among nearby vertices, whereas higher-order neighborhoods provide access to more distant relationships. Relying on a single neighborhood order can therefore restrict the information available to the classifier. Increasing message-passing depth extends the receptive field, but repeated propagation can lead to over-smoothing, making node representations increasingly similar and potentially obscuring useful information (Li et al., 2018; Oono & Suzuki, 2020). This motivates architectures that retain and combine representations from multiple neighborhood orders.
2. Second, the resulting graph representation should remain stable under perturbations. Observed graphs may contain noisy node attributes, missing edges, or spurious connections, and GNN predictions can be sensitive to changes in features or topology (Dai et al., 2018; Zügner et al., 2018; Bojchevski & Günnemann, 2019). Such changes can affect both the node embeddings produced by message passing and the graph summary constructed through pooling. For graph classification, a useful pooling mechanism should preserve task-relevant information while limiting sensitivity to perturbations in its input embeddings.
3. Third, extracting these representations must remain computationally feasible. Processing all nodes and edges across multiple neighborhood orders increases the computational cost and can place substantial demands on memory. Subsampling offers a way to reduce this burden, but it also removes information and changes the neighborhoods over which messages are propagated.

   The three requirements are therefore closely connected. Reducing training cost is useful *only when* the sampled graphs retain sufficient information for prediction, and incorporating higher-order information is useful only when the resulting representations remain stable and affordable to compute.

Recent methodological advances have addressed individual aspects of this problem. MaGNet (Zhou et al., 2025) constructs separate actors for different neighborhood orders and uses critic-derived weights to fuse their graph representations. Robust singular pooling (RS-Pool) (Ennadir et al., 2025) uses a dominant right singular vector as the readout, whose stability depends on spectral separation. Vertex subsampling based methods for graph learning are known to reduce computational cost while retaining inferential accuracy (Hamilton et al., 2017; Chen et al., 2018; Chiang et al., 2019; Zeng et al., 2020; Dasgupta & Sengupta, 2022; Komolafe et al., 2022; Bhadra et al., 2026; Chakrabarty et al., 2025; Chakraborty et al., 2025). We adapt these ingredients to graph classification by placing singular pooling inside each actor and training on induced subgraphs of each labeled observation.

The three components have distinct but complementary roles. Multi-order actors expose information from several propagation distances to the classifier. Singular pooling summarizes the geometry of the node embeddings at each order, while subgraph sampling reduces the amount of graph data used to fit the model. Their combination is useful only if the sampled views retain relevant information, since an efficient training procedure is of limited value when it discards the structures that distinguish the classes. We therefore study both computational savings and the conditions under which subgraph representations approximate their full-graph counterparts.

This distinction between training and prediction raises the following theoretical question: when can a representation computed on an induced subgraph approximate the corresponding full graph representation? Deleting nodes changes both the messages and the degree normalization. Moreover, a singular vector readout depends on the feature geometry after propagation, so preservation of adjacency eigenvalues alone is insufficient. We analyze normalized feature Gram matrices, which have the same dimension for the graph and subgraph and determine the singular pooling directions.

Our contributions are threefold.

- First, we formulate a graph classification framework combining multi-order actor representations, singular pooling, and subgraph sampling.
- Second, we derive finite sample Gram matrix bounds that separate sampling and propagation errors, give sufficient degree conditions for the latter, and translate them into repre-

sentation, logit, and prediction stability. We also state the additional conditions needed to transfer a subgraph training objective to full-graph risk.

- Third, our simulations and benchmark experiments quantify the tradeoffs and interplay between accuracy, robustness, and runtime across sampling ratios.

Relation to prior work. Sampling-based GNN methods provide the computational foundation for this approach. GraphSAGE samples local neighborhoods (Hamilton et al., 2017), FastGCN uses importance sampling in graph convolutions (Chen et al., 2018), and Cluster-GCN builds training batches from graph clusters (Chiang et al., 2019). GraphSAINT samples subgraphs before constructing the GNN computation and introduces normalization and variance-reduction procedures (Zeng et al., 2020). Our setting consists of multiple labeled graphs, with each sampled view carrying its parent graph's label. The sampling objective is consequently tied to preserving a graph-level readout, and the normalization in our analysis is recomputed directly from subgraph degrees. Unbiasedness properties of other sampling schemes cannot simply be assumed for this operator.

In particular, our proposed GNN framework is based on two recent methodological advancements: MaGNet and RS-Pool Zhou et al. (2025); Ennadir et al. (2025). MaGNet provides the multi-order actor-critic organization (Zhou et al., 2025), and RS-Pool provides the singular vector readout (Ennadir et al., 2025).

2 SMARS-GNN

We observe labeled graphs $\{(G_i, y_i)\}_{i=1}^n$, with adjacency A_i, node features $X_i \in \mathbb{R}^{N_i \times p}$, and class $y_i \in \{1, \dots, C\}$. Each graph is an observation. Sampled views of the same graph are dependent observations with the same parent label.

2.1 MULTI-ORDER ACTORS AND SINGULAR POOLING

For an undirected graph, define $\mathcal{L}_i = D_i^{-1/2} A_i D_i^{-1/2}$, setting inverse square roots of zero degrees to zero. For $k = 1, \dots, K$, the actor embedding and readout are

$$H_i^{(k)} = \mathcal{L}_i^k X_i W_A, \qquad g_i^{(k)} = \tau v_1(H_i^{(k)}), \tag{1}$$

where $W_A \in \mathbb{R}^{p \times h}$ is shared across orders, $\tau > 0$ is a fixed scale, and $v_1(H)$ is a unit dominant right singular vector. There are no intervening nonlinear activations in these propagation steps. Separate readouts retain access to lower orders while adding higher-order information. A dominant vector can be approximated by power iteration without a full singular value decomposition.

For $H = U\Sigma V^\top$, the vector $v_1(H)$ lies in the latent space $\mathbb{R}^h$, so the readout dimension does not depend on the graph size. Equivalently, it solves

$$v_1(H) \in \arg\max_{\|v\|_2 = 1} v^\top H^\top H v. \tag{2}$$

Thus singular pooling selects a direction with large aggregate squared projection of the node embeddings. If Π permutes the vertices, then $(\Pi H)^\top (\Pi H) = H^\top H$, so a consistently oriented, uniquely defined leading direction is permutation invariant. The fixed scale τ controls the readout norm. The leading vector alone does not retain every aspect of the embedding matrix, such as the entire singular-value spectrum, so robustness and predictive usefulness must be assessed separately.

A critic MLP assesses the predictive quality of each actor representation and supplies an order-specific reliability weight $\alpha^{(k)}$. The weights are fitted from training data and held fixed for new inputs. The fused representation, logits, and probabilities are

$$z_i = \sum_{k=1}^{K} \alpha^{(k)} g_i^{(k)}, \qquad s(G_i) = W_o z_i + b_o, \qquad p_i = \mathrm{softmax}(s(G_i)). \tag{3}$$

The critic is used to evaluate actor representations during supervised training; the actor–critic terminology does not require an environment or a reinforcement-learning reward. Singular pooling is applied separately within each actor, before fusion, so its stability depends on that order's embedding geometry. The theory below requires a consistent singular vector sign convention and a nonzero leading gap; arbitrary numerical singular value decomposition (SVD) signs alone do not provide a signed-vector stability guarantee.

Algorithm 1 SMaRS-GNN training and prediction: overview

Require: Labeled training graphs, orders $1{:}K$, sampling ratio ρ, views R, scale τ.
1: Sample R induced subgraphs per training graph; inherit features and labels.
2: Recompute each subgraph's degrees and normalized adjacency.
3: **repeat**
4: **for** $k = 1, \ldots, K$ **do**
5: Compute order-k embeddings and singular readouts for the sampled views.
6: Update the critic and the order-specific fusion weight $\alpha^{(k)}$.
7: **end for**
8: Fuse readouts within each view and compute class probabilities.
9: Average probabilities within each parent; update parameters using the graph labels.
10: **until** the stopping criterion is met
11: Predict on each full test graph using the fitted parameters and actor weights.

Critic-derived weights. For the binary tasks considered here, write $q_j^{(k)}$ for the critic's class probabilities on training input j and let $\beta_j^{(k)} \geq 0$ denote its sample weight, with a positive total weight. Here j indexes a training graph (or subgraph). The MaGNet weighting rule uses

$$\epsilon^{(k)} = \frac{\sum_j \beta_j^{(k)} \mathbf{1}\{y_j \neq \arg\max_c q_{j,c}^{(k)}\}}{\sum_j \beta_j^{(k)}}, \qquad \alpha^{(k)} = \frac{1}{2} \log \frac{1 - \epsilon^{(k)}}{\epsilon^{(k)}}. \tag{4}$$

An actor with smaller weighted classification error receives a larger fusion coefficient. These coefficients are not necessarily nonnegative or normalized to sum to one; the theory therefore uses their absolute values. The formula requires $0 < \epsilon^{(k)} < 1$, and endpoint clipping must be specified in an implementation. The learned coefficients are global order-specific weights, not functions of the test graph. Appendix A gives the detailed training procedure and numerical considerations.

2.2 Subgraph training and full-graph prediction

For each training graph, sample R subsets $S_i^{(r)}$ of size $m_i = \lfloor \rho N_i \rfloor$ uniformly without replacement, with $\rho \in (0, 1]$. Form the induced adjacency $A_i^{(r)}$, inherit the selected rows of X_i, and recompute the degree normalization $\mathcal{L}_i^{(r)}$ within each subgraph. Each view is assigned the parent label y_i. Replacing $(\mathcal{L}_i, X_i)$ in equations (1)–(3) by $(\mathcal{L}_i^{(r)}, X_i^{(r)})$ gives $p_i^{(r)}$. We aggregate the probabilities,

$$\bar{p}_i = \frac{1}{R} \sum_{r=1}^{R} p_i^{(r)}, \tag{5}$$

and fit the classifier using $(\bar{p}_i, y_i)$. For inference, the fitted parameters and actor weights act on the full observed graph, and $\widehat{y}(G) = \arg\max_c p_c(G)$. Figure 1 summarizes the framework.

The sampling ratio is chosen so that every training graph has at least one sampled vertex. Graph-derived features are computed before sampling and inherited by each view. Recomputing these features on the subgraph would change the input as well as the propagation operator and would require another error term. The training objective for cross-entropy is

$$\widehat{R}_{s,R}^{p}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \log \bar{p}_{i,y_i}(\theta). \tag{6}$$

Averaging probabilities before taking the loss differs from averaging the losses of separate views. Our training-risk discussion uses the former objective. Algorithm 1 summarizes the workflow. The full procedure is Algorithm 2 in the appendix.

2.3 Computational considerations

For a graph with N vertices and edge set E, uniform fixed-size sampling retains each edge with probability $m(m-1)/\{N(N-1)\}$. Thus the expected number of sampled edges is approximately

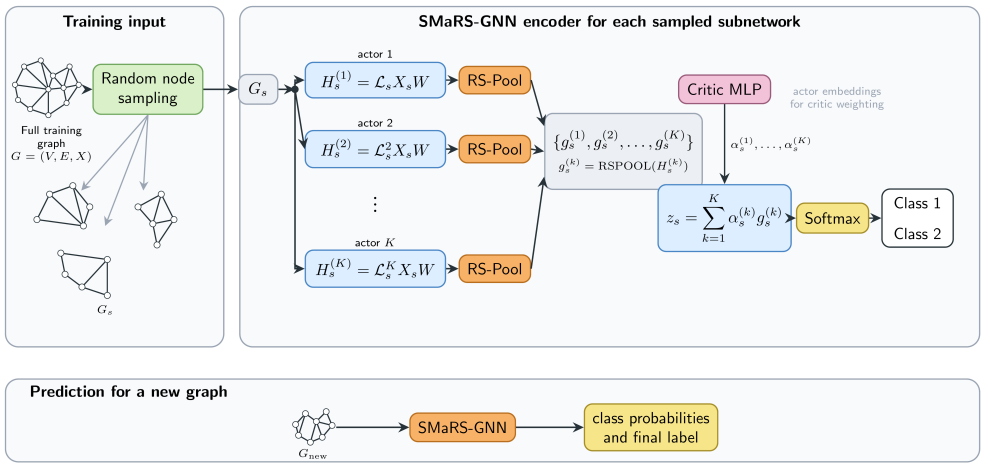

Figure 1: SMaRS-GNN trains on induced subgraphs and predicts on the full test graph. Singular pooling produces a representation for each neighborhood order. The critic-derived fusion weights are fitted during training and fixed at inference.

$\rho^2|E|$. This explains a potential reduction in message-passing work; actual training time also depends on the number of views, pooling, optimization, and implementation overhead. We evaluate the resulting runtime tradeoff empirically rather than claiming a universal speedup factor.

The linear actors admit a recursive computation: set $H^{(0)} = XW_A$ and $H^{(k)} = \mathcal{L}H^{(k-1)}$. This avoids explicitly forming powers of the adjacency operator or repeating all earlier propagation steps for each actor. With sparse adjacency, one view requires $O(Nph + K|E|h)$ arithmetic for feature projection and propagation. If singular pooling uses J power iterations, multiplication by $H^\top(Hv)$ costs $O(Nh)$ per iteration, giving an additional $O(KJNh)$ term. These counts describe a possible forward implementation and exclude critic fitting, optimizer steps, and data preparation. For R sampled views, the expected propagation work is proportional to $R\rho^2|E|$ and the nodewise work to $R\rho N$, up to the finite-size correction above. This explains why a smaller sampling ratio can reduce training cost, while additional views can offset some of the savings. Full-graph inference retains the cost of evaluating one full graph.

3 Theoretical Results

We distinguish evaluation of a shared fitted model on two graph inputs from comparison of training procedures. The former follows from Gram-matrix preservation and a margin condition. The latter requires uniform control over the parameter class and an additional estimation bound. All proofs and complete versions of the results appear in Appendix B.

3.1 A graph characteristic adapted to singular pooling

Fix a simple undirected graph G with $N \geq 3$ and a uniform subset S of m vertices, $2 \leq m \leq N$. Let P_S select the sampled rows and let $\mathcal{L}_s$ be normalized using the subgraph degrees. Set

$$F_k = \mathcal{L}^k X, \quad F_{k,s} = \mathcal{L}_s^k P_S X, \qquad Q_k = \frac{F_k^\top F_k}{N}, \quad Q_{k,s} = \frac{F_{k,s}^\top F_{k,s}}{m}. \tag{7}$$

Features are fixed before sampling and inherited on the subgraph. The hidden Gram matrices are $C_k = W_A^\top Q_k W_A$ and $C_{k,s} = W_A^\top Q_{k,s} W_A$. Their leading eigenvectors are the right singular vectors used in pooling. Normalization by graph size does not change those vectors.

Theorem 3.1 (Sampling and propagation errors). *Suppose* $\max_i \|(F_k)_{i,:}\|_2 \leq R_k$, *and deterministic* $r_k \geq 0$ *satisfy*

$$\mathbb{P}_S\big\{\|F_{k,s} - P_S F_k\|_{\mathrm{op}}/\sqrt{m} \leq r_k \text{ for all } k \leq K\big\} \geq 1 - \eta_p.$$

For $0 < \eta_g < 1 - \eta_p$, define

$$\varepsilon_k = R_k^2 \sqrt{\frac{K(N-m)}{\eta_g m(N-1)}} + 2R_k r_k + r_k^2. \tag{8}$$

With probability at least $1 - \eta_g - \eta_p$, simultaneously for all $k \le K$, $\|Q_{k,s} - Q_k\|_{\mathrm{op}} \le \varepsilon_k$. On that same event, $\|C_{k,s} - C_k\|_{\mathrm{op}} \le B^2 \varepsilon_k$ for every $\|W_A\|_{\mathrm{op}} \le B$.

Remark. To see why the bound separates two errors, let $\bar{F}_k = P_S F_k$, $E_k = F_{k,s} - \bar{F}_k$, and $\widetilde{Q}_k = \bar{F}_k^\top \bar{F}_k / m$. Then

$$Q_{k,s} - Q_k = (\widetilde{Q}_k - Q_k) + \frac{\bar{F}_k^\top E_k + E_k^\top \bar{F}_k + E_k^\top E_k}{m}. \tag{9}$$

The first difference samples fixed outer products of full-graph feature vectors. Its second moment has the finite-population factor $(N-m)/\{m(N-1)\}$, so Markov's inequality and a union bound control all actor orders. For the remaining terms, $\|\bar{F}_k\|_{\mathrm{op}} \le \sqrt{m} R_k$ and the propagation assumption give $2R_k r_k + r_k^2$. No independence across actors is required. This also explains why the conclusion is uniform in bounded W_A: the event is determined before multiplication by the learned feature weights. The complete proof is in Appendix B.2.

The propagation assumption can be verified under explicit graph conditions. Suppose $d_{\min} > 0$, $d_{\max}/d_{\min} \le \kappa$, and $\max_i \|X_{i,:}\|_2 \le M$. Put $q = (m-1)/(N-1)$, $c_N = (N-1)/(N-2)$, and $R_* = M\sqrt{\kappa}$. If $m < N$ and $q d_{\min} \ge 12 \log(2m/\eta_d)$, Proposition B.4 gives $R_k = R_*$, $\eta_p = \eta_d + \eta_z$, and

$$r_k = R_* \left(3k \sqrt{\frac{3\log(2m/\eta_d)}{q d_{\min}}} + \sqrt{\frac{k K c_N (1-q)}{\eta_z q d_{\min}}} \right). \tag{10}$$

Here $\eta_d, \eta_z > 0$ and $\eta_g + \eta_d + \eta_z < 1$. This result allows heterogeneous degrees. For fixed K, M, κ and confidence level, the resulting ε_k is $O(m^{-1/2} + \sqrt{\log m/(q d_{\min})})$ when $q d_{\min}/\log m \to \infty$; it is $O(\sqrt{\log m/m})$ if $d_{\min} \ge cN$.

Connection to graph spectra. If $\mathcal{L} = U \Lambda U^\top$, with eigenpairs (λ_j, u_j), then

$$Q_k = \frac{X^\top \mathcal{L}^{2k} X}{N} = \sum_{j=1}^{N} \lambda_j^{2k} a_j a_j^\top, \qquad a_j = N^{-1/2} X^\top u_j. \tag{11}$$

Thus, the graph characteristics relevant to the readout are the eigenvalues and the projections of the node features onto their eigenspaces. Two graphs with similar eigenvalues can still have different feature Gram matrices. Conversely, controlling these Gram matrices suffices for our analysis without separately matching every graph eigenvector. Repeated eigenvalues can be grouped using spectral projectors, making the expression independent of a choice of basis within each eigenspace.

3.2 From feature geometry to predicted labels

For hidden dimension $h \ge 2$, let $\delta_k = \lambda_1(C_k) - \lambda_2(C_k) > 0$. Fix the network parameters, pooling scales, and fusion weights across G and G_s. A fixed unit anchor a_k chooses the sign of each leading vector by positive inner product. Suppose the full-graph vector obeys $a_k^\top v_k \ge \zeta_k > 0$.

Theorem 3.2 (Pooling and prediction stability)**.** *On the event of Theorem 3.1, write $b_k = B^2 \varepsilon_k$ and assume $b_k \le \delta_k/4$ and $\xi_k = 2\sqrt{2} b_k/\delta_k < \zeta_k$ for every k. Then*

$$\|g_{k,s} - g_k\|_2 \le \tau \xi_k, \qquad \|s(G_s) - s(G)\|_\infty \le t(G) := \|W_o\|_{2\to\infty} \sum_{k=1}^{K} |\alpha^{(k)}| \tau \xi_k. \tag{12}$$

If the gap $\gamma(G)$ between the two largest full-graph logits exceeds $2t(G)$, the predicted labels agree with probability at least $1 - \eta$, where $\eta = \eta_g + \eta_p$.

The proof uses eigenvector perturbation and then a classification-margin argument (Theorem B.7).

For a random labeled graph satisfying these conditions almost surely, conditioning on the fitted network and using the parent label for both inputs gives

$$|\operatorname{Acc}_{\mathrm{sub}} - \operatorname{Acc}_{\mathrm{full}}| \le \mathbb{P}\{\gamma(G) \le 2t(G)\} + \eta \tag{13}$$

(Corollary B.9). Thus preservation of the relevant Gram matrices connects a graph characteristic to accuracy through the margin distribution.

The theorem gives a concrete sequence of requirements. Sampling must preserve feature geometry well enough relative to each actor's latent gap; the resulting readout error, after fusion and classification, must then be small relative to the prediction margin. A large graph by itself is not sufficient: small gaps or near-tied logits can make the guarantee weak. If $t(G) \le t_m$ almost surely and the fitted classifier has a margin distribution satisfying $\mathbb{P}\{\gamma(G) \le u\} \le C_\gamma u^\beta$, then equation (13) is at most $C_\gamma(2t_m)^\beta + \eta$ whenever $2t_m$ lies in the stated margin range. This is a preservation result for the same fitted classifier, with probability also taken over the sampled view.

3.3 Generalization and actor-level robustness

We complement the subsampling results with a generalization-rate statement and a robustness result for the full-graph actors. Supporting conditions and proofs are given in Appendices B.5 and B.6.

Theorem 3.3 (Conditional generalization rate)**.** *Let $\widehat{f}_n$ be the binary prediction score of SMaRS-GNN trained using n independent labeled parent graphs and a fixed number of sampled views per graph. Under conditions (G1)–(G4) in Appendix B.5, including $O(n^{-1/2})$ estimation, approximation, subsampling, and optimization terms, for each fixed $\delta \in (0,1)$, with probability at least $1-\delta$ over training,*

$$\mathbb{P}\{Y\widehat{f}_n(G) \le 0\} = O(n^{-1/2}). \tag{14}$$

The probability inside the bound is over an independent full test graph and its label.

Remark (Connection to MaGNet). The rate in Theorem 3.3 agrees with the regular-graph generalization rate reported following Theorem 5.3 of Zhou et al. (2025). In SMaRS-GNN, the training-to-test comparison additionally accounts for the use of induced subgraphs during training and full graphs at prediction time. The estimation condition concerns the singular-pooling model class; n counts independent parent graphs, with their sampled views grouped together.

Theorem 3.4 (Full-graph actor robustness)**.** *Fix a full graph G and actor $H_k = \mathcal{L}^k X W_A$, and let $\widetilde{H}_k = \mathcal{L}^k \widetilde{X} W_A$ under a feature perturbation satisfying $\max_i \|\widetilde{X}_{i,:} - X_{i,:}\|_2 \le \epsilon$. Write $\delta_k > 0$ for the leading gap of $H_k^\top H_k/N$, and let b_k^X be the Gram-perturbation bound in Appendix B.6. Under the small-perturbation and orientation conditions of Proposition B.12, the expected actor readout discrepancy satisfies*

$$R_\epsilon^{(k)} := \mathbb{E}\|\tau v_1(\widetilde{H}_k) - \tau v_1(H_k)\|_2 \le \min\left\{2\tau, \frac{2\sqrt{2}\tau b_k^X}{\delta_k}\right\}. \tag{15}$$

The conditions are required for every realized perturbation; the bound 2τ holds without the small-perturbation conditions.

Remark (Connecting RS-Pool and MaGNet). Theorem 3.4 places the singular-pooling stability mechanism of Ennadir et al. (2025) within each order-specific actor of MaGNet. The clean actor gap and pooling scale govern the stability of the summaries supplied to the critic. This result concerns feature perturbations on a fixed full graph, independently of the subsampling guarantees in Sections 3.1–3.2.

4 Experiments

We compare a two-layer GCN with average pooling (Av-GCN), a two-layer GCN with singular pooling (RS-GCN), full-graph MaGNet with singular pooling (MaRS-GNN), and SMaRS-GNN

Model and signal	Nodes	Clean (%)		PGD (%)		Time (min)	
		MaRS	SMaRS	MaRS	SMaRS	MaRS	SMaRS
RDPG, $\delta = .01$	1200–1500	61.1	58.2	43.0	44.7	30.5	9.9
RDPG, $\delta = .03$	1200–1500	93.2	85.6	61.3	64.8	30.5	9.9
RDPG, $\delta = .05$	1200–1500	100.0	100.0	77.3	70.4	30.5	9.9
RDPG, $\delta = .03$	1700–2000	90.5	88.0	50.9	47.8	93.3	26.6
RDPG, $\delta = .05$	1700–2000	100.0	100.0	100.0	100.0	93.3	26.6
RDPG, $\delta = .05$	2500–3000	100.0	100.0	100.0	100.0	258.7	62.4
NW-ER, $\beta = .10$	1200–1500	75.3	63.2	55.6	49.8	31.5	9.4
NW-ER, $\beta = .25$	1200–1500	100.0	100.0	68.4	67.9	31.5	9.4
NW-ER, $\beta = .25$	2500–3000	100.0	100.0	64.9	65.3	279.3	64.0

Table 1: Summary of all simulation settings comparing full-graph MaRS-GNN with SMaRS-GNN at $\rho = 0.6$. Complete results, including random perturbations, both GCN baselines, and all sampling ratios, are in Appendix C.

with $\rho \in \{0.2, 0.4, 0.6\}$. Methods share data splits within each experiment. We report clean accuracy, accuracy under random perturbations and projected-gradient-descent (PGD) attacks with the reported 20% perturbation budget, and training time in minutes. Simulation accuracies and times are averages over five runs. Full specifications and all reported simulation results appear in Appendix C.

The comparisons address complementary questions. Av-GCN versus RS-GCN examines the effect of replacing average pooling in a two-layer baseline. MaRS-GNN versus SMaRS-GNN examines the computational and predictive effect of subsampling within the multi-order singular-pooling architecture. Varying ρ then shows how this tradeoff changes with the retained fraction of vertices.

4.1 Simulated graphs

The first experiment distinguishes two random dot product graph (RDPG) distributions, with class separation controlled by δ and graph sizes from $1{,}200$ to $3{,}000$ nodes. Each setting has 500 graphs per class. The second compares a Newman–Watts Erdős–Rényi (NW-ER) alternative with an Erdős–Rényi baseline, using β to control small-world structure (Lovekar et al., 2025). Both use six-dimensional node features constructed from a constant, noisy normalized degree, noisy clustering coefficient, and three noise coordinates. Specifically,

$$X_{iv} = \left(1, \frac{d_{iv}}{N_i - 1} + \eta_{iv}^{(1)}, \mathrm{clust}_{iv} + \eta_{iv}^{(2)}, \xi_{iv}^{(1)}, \xi_{iv}^{(2)}, \xi_{iv}^{(3)}\right), \tag{16}$$

where $\eta_{iv}^{(j)} \sim N(0, 0.1^2)$ and $\xi_{iv}^{(j)} \sim N(0, 0.4^2)$. The structural features describe observable properties of each graph, while the remaining coordinates are noise variables. Latent positions and class labels are not supplied as node features. Features are inherited when sampling training views.

Table 1 summarizes all nine simulation settings at $\rho = 0.6$; Tables 3–4 retain every original simulation entry. Increasing ρ generally recovers more information at increased cost. In the largest RDPG setting, $\rho = 0.6$ matches MaRS-GNN's reported clean and attacked accuracies while reducing training time from 258.7 to 62.4 minutes ($4.15\times$ faster). At moderate signal, there is an accuracy cost: for $\delta = 0.03$ and $1{,}200$–$1{,}500$ nodes, clean accuracy decreases from 93.2% to 85.6%, while PGD accuracy increases from 61.3% to 64.8%. These results illustrate a tradeoff across model settings and varying ρ.

The complete tables also show why clean and perturbed accuracy should be considered together. For RDPG graphs with $1{,}200$–$1{,}500$ nodes and $\delta = 0.05$, sampling at $\rho = 0.4$ or 0.6 retains the reported 100% clean accuracy, yet PGD accuracy remains below the full-graph result. For NW-ER graphs of the same size and $\beta = 0.25$, all methods reach 100% clean accuracy, while perturbed accuracies vary substantially. In the larger NW-ER setting, $\rho = 0.6$ gives 65.3% PGD accuracy compared with 64.9% for MaRS-GNN, but its random-perturbation accuracy is lower (93.0% versus 98.2%).

Across the RDPG settings, SMaRS-GNN at $\rho = 0.6$ is approximately 3.08–4.15 times faster than MaRS-GNN. The corresponding NW-ER range is 3.35–4.36. Smaller sampling ratios give further time savings but can lose substantial signal: in the moderate-signal RDPG setting with $1{,}200$–$1{,}500$ nodes, clean accuracy rises from 72.8% to 78.0% to 85.6% as ρ increases from 0.2 to 0.4 to 0.6,

		Av-GCN	RS-GCN	MaRS	SMaRS $\rho = 0.2$	SMaRS $\rho = 0.4$	SMaRS $\rho = 0.6$
Reddit-B	Clean	73.0	74.5	88.1	66.3	69.8	76.6
	Random	71.3	74.2	78.9	67.3	68.3	73.2
	PGD	66.1	71.0	72.5	57.8	62.7	67.3
	Time	24.7	28.3	58.7	6.0	12.2	18.1
D&D	Clean	75.1	74.2	82.6	63.7	67.9	72.6
	Random	62.0	68.3	71.0	58.8	61.3	63.8
	PGD	7.3	30.4	32.7	27.4	29.4	29.6
	Time	14.2	16.9	31.5	7.1	13.6	19.9
PROTEINS	Clean	74.3	73.7	84.9	66.3	68.1	75.6
	Random	68.8	70.4	70.2	61.5	64.0	68.8
	PGD	44.7	51.2	52.5	48.3	50.8	53.4
	Time	6.4	8.5	14.8	1.6	3.7	4.9

Table 2: Benchmark accuracy (percent) and training time (minutes). MaRS denotes full-graph MaRS-GNN and SMaRS denotes induced-subgraph training. All reported benchmark measurements are retained.

with times of 3.2, 6.0, and 9.9 minutes. The computational benefit is therefore most useful when the chosen ratio retains enough task-relevant information.

4.2 Benchmark graph classification

We evaluate Reddit-B, D&D, and PROTEINS from TUDataset (Morris et al., 2020). Reddit-B contains discussion-thread graphs, while D&D and PROTEINS contain protein-structure graphs. Table 2 reports all methods and sampling ratios. MaRS-GNN has the highest clean accuracy on these datasets. SMaRS-GNN at $\rho = 0.6$ reduces training time by factors of 3.24, 1.58, and 3.02, respectively, with clean-accuracy decreases of 11.5, 10.0, and 9.3 percentage points. Relative to RS-GCN, it is faster on Reddit-B and PROTEINS but slower on D&D. PGD accuracy exceeds MaRS-GNN on PROTEINS at $\rho = 0.6$, but is lower on Reddit-B and D&D. Choosing the sampling ratio therefore depends on the acceptable loss in accuracy and the perturbations of interest.

Sampling ratio and practical interpretation. Clean accuracy increases with ρ on all three benchmarks, together with training time. On Reddit-B, moving from $\rho = 0.2$ to 0.6 recovers 10.3 percentage points at a cost of 12.1 additional minutes. On PROTEINS, $\rho = 0.6$ exceeds both two-layer baselines in clean accuracy while taking less time, although it remains below MaRS-GNN. On D&D, this ratio is slower and less accurate on clean graphs than either two-layer baseline. The preferred method therefore depends on the application and compute budget.

These results motivate selecting ρ and actor order jointly using validation data and a stated accuracy or runtime requirement. The theory identifies obstacles to transfer rather than an optimal sampling ratio. The tables compare separately trained methods, whereas the perturbation bounds in Section 3 compare inputs to the same fitted model.

5 Discussion

SMaRS-GNN combines order-specific singular pooling with subgraph training and full-graph inference. The theory identifies a feature-dependent Gram matrix that connects sampling to representation and prediction stability, and separates these guarantees from the stronger requirements for training-risk transfer. The experiments show substantial training-time reductions with setting-dependent accuracy and robustness costs.

There are several open directions for future research. The explicit propagation bound requires sufficiently large sampled minimum degrees, and an interesting problem is to extend these ideas to sparse graphs. Adaptive sampling, joint selection of ρ and K, and broader robustness guarantees are also important directions for future work.

AI USE STATEMENT

AI assistance was used to polish writing.

REPRODUCIBILITY STATEMENT

We have provided source code in the supplementary materials to support reproduction of our experimental results. The assumptions underlying our theoretical results are stated explicitly, and complete proofs and theoretical details are provided in the appendix. References to the datasets used in our experiments are provided to enable access to the original data sources.

REFERENCES

Subhankar Bhadra, Marianna Pensky, and Srijan Sengupta. Scalable community detection in massive networks via predictive assignment. *Journal of the American Statistical Association*, pp. 1–14, 2026.

Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, Cameron Musco, and Archan Ray. Sublinear time eigenvalue approximation via random sampling. *Algorithmica*, 86:1764–1829, 2024. doi: 10.1007/s00453-024-01208-5.

Aleksandar Bojchevski and Stephan Günnemann. Adversarial attacks on node embeddings via graph poisoning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 695–704. PMLR, 09–15 Jun 2019.

Sayan Chakrabarty, Srijan Sengupta, and Yuguo Chen. Subsampling based community detection for large networks. *Statistica Sinica*, 35(3):1–42, 2025. doi: 10.5705/ss.202022.0108.

Kaustav Chakraborty, Srijan Sengupta, and Yuguo Chen. Scalable estimation and two-sample testing for large networks via subsampling. *Journal of Computational and Graphical Statistics*, 34(3): 1127–1139, 2025.

Jie Chen, Tengfei Ma, and Cao Xiao. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rytstxWAW.

Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 257–266, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330925. URL https://doi.org/10.1145/3292500.3330925.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1115–1124. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/dai18b.html.

Anirban Dasgupta and Srijan Sengupta. Scalable estimation of epidemic thresholds via node sampling. *Sankhya A*, 84:321–344, 2022.

Sofiane Ennadir, Oleg Smirnov, Yassine Abbahaddou, Lele Cao, and Johannes F Lutzeyer. Enhancing graph classification robustness with singular pooling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4666.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

International Energy Agency. Energy and AI. Technical report, International Energy Agency, Paris, 2025. URL https://www.iea.org/reports/energy-and-ai.

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.

Tomilayo Komolafe, Allan Fong, and Srijan Sengupta. Scalable community extraction of text networks for automated grouping in medical databases. *Journal of Data Science*, 21(3):470–489, 2022. ISSN 1680-743X. doi: 10.6339/22-JDS1038.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. doi: https://doi.org/10.1609/aaai.v32i1.11604.

Kartik Lovekar, Srijan Sengupta, and Subhadeep Paul. Testing for the network small-world property. *Electronic Journal of Statistics*, 19(1):1453–1506, 2025.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *arXiv preprint arXiv:1907.10597*, 2019. URL https://arxiv.org/abs/1907.10597.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355/.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015. doi: 10.1093/biomet/asv008.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor K. Prasanna. GraphSAINT: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJe8pkHFwS.

Wenzhuo Zhou, Annie Qu, Keiland W Cooper, Norbert Fortin, and Babak Shahbaba. A model-agnostic graph neural network for integrating local and global information. *Journal of the American Statistical Association*, 120(550):1225–1238, 2025.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 2847–2856, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220078. URL https://doi.org/10.1145/3219819.3220078.

A Additional method details

A.1 Message passing and graph classification

Most GNNs are built from neural message passing layers. The basic idea is to update the representation of each node by combining its current representation with information aggregated from its neighbors. Let the initial layer information $h_{iv}^{(0)} = x_{iv}$ be the feature vector of node v in graph G_i. At layer ℓ, a generic message passing update can be written as

$$m_{iv}^{(\ell)} = \mathrm{AGG}^{(\ell)}\left(\left\{h_{iu}^{(\ell)} : u \in \mathcal{N}_i(v)\right\}\right), \qquad h_{iv}^{(\ell+1)} = \mathrm{UPD}^{(\ell)}\left(h_{iv}^{(\ell)}, m_{iv}^{(\ell)}\right),$$

where $\mathcal{N}_i(v)$ is the neighborhood of node v, $\mathrm{AGG}^{(\ell)}(\cdot)$ is a permutation-invariant aggregation function, and $\mathrm{UPD}^{(\ell)}(\cdot)$ is a trainable update map. After L message passing layers, the node embeddings are obtained as $h_{iv}^{(L)}$, and a graph-level pooling function produces a fixed-dimensional embedding

$$g_i = \mathrm{POOL}\left(\left\{h_{iv}^{(L)} : v \in V_i\right\}\right).$$

This graph embedding is then passed to a classifier to predict y_i.

A common example is the graph convolutional network (GCN) (Kipf & Welling, 2017), where a normalized graph operator propagates node features before applying a trainable linear transformation and nonlinear activation function. Let $\tilde{A}_i = A_i + I$ and $\tilde{\mathcal{L}}_i = \tilde{D}_i^{-1/2}\tilde{A}_i\tilde{D}_i^{-1/2}$, where $\tilde{D}_i$ is the degree matrix of $\tilde{A}_i$, a GCN layer update can be written as

$$H_i^{(\ell+1)} = \sigma\left(\tilde{\mathcal{L}}_i H_i^{(\ell)} W^{(\ell)}\right),$$

where $H_i^{(0)} = X_i$, $W^{(\ell)}$ is a trainable weight matrix for layer l, and $\sigma(\cdot)$ is an activation function. Stacking more layers increases the neighborhood range represented by each node, but it can also make node embeddings increasingly similar. Thus, graph classification requires a balance between capturing higher-order graph structure and preserving informative lower-order representations.

A.2 Critic-based actor weighting

Equation (4) gives the weighting rule. Let $g_i^{(k)}$ denote the graph-level embedding obtained from the kth actor. A critic network, implemented as an l_0-layer multilayer perceptron (MLP) followed by a softmax output, maps $g_i^{(k)}$ to a class probability vector $q_i^{(k)}$. The critic therefore measures how predictive the kth actor embedding is. In the binary MaGNet formulation, this quality can be summarized through the weighted classification error

$$\epsilon^{(k)} = \frac{\sum_{i=1}^{n} \beta_i^{(k)} \mathbf{1}\{y_i \neq \arg\max_c q_{i,c}^{(k)}\}}{\sum_{i=1}^{n} \beta_i^{(k)}},$$

where $\beta_i^{(k)}$ is the sample weight used at the kth actor step. The corresponding actor weight is then

$$\alpha^{(k)} = \frac{1}{2}\log\left(\frac{1-\epsilon^{(k)}}{\epsilon^{(k)}}\right).$$

This construction is similar to a reliability score used in boosting. Actor embeddings that lead to lower classification error receive larger fusion weights. For multi-class classification, the same idea can be implemented using the critic's cross-entropy or classification accuracy as the order-specific reliability score.

A.3 Training and inference algorithm

The detailed procedure below describes the training and prediction steps of the proposed method. The sampling ratio is chosen so that $\lfloor \rho N_i \rfloor \geq 1$ for every training graph. Features are constructed before sampling and inherited by each subgraph.

Algorithm 2 SMaRS-GNN: subsampled training and full-graph prediction

Input: Training graphs $\mathcal{D} = \{(G_i, y_i)\}_{i=1}^n$; maximum actor order K; sampling ratio ρ; replications R; RS-Pool scale $\tau > 0$; test graph G_{new}.
Output: Fitted model; class probabilities $\hat{p}_{new}$ and predicted label $\hat{y}_{new}$.

procedure TRAIN($\mathcal{D}, K, \rho, R, \tau$)
  For each i and r, sample $\lfloor \rho N_i \rfloor$ nodes uniformly without replacement.
  Form the induced subnetwork $G_i^{(r)}$ with adjacency $A_i^{(r)}$, features $X_i^{(r)}$, and label y_i.
  Compute $\mathcal{L}_i^{(r)} = (D_i^{(r)})^{-1/2} A_i^{(r)} (D_i^{(r)})^{-1/2}$ for all i, r.
  Initialize W_A, the critic parameters, and the output parameters W_o, b_o.
  repeat
    for $k = 1, \ldots, K$ **do**
      Compute $H_i^{(r,k)} = (\mathcal{L}_i^{(r)})^k X_i^{(r)} W_A$ for all i, r.
      Apply singular pooling: $g_i^{(r,k)} \leftarrow \tau v_1(H_i^{(r,k)})$, approximating v_1 by power iteration.
      Fit/update the critic on $\{(g_i^{(r,k)}, y_i)\}_{i,r}$ and obtain its class probabilities.
      Set $\alpha^{(k)}$ from the critic's prediction quality using the weighting rule described above.
    end for
    Fuse actor embeddings: $z_i^{(r)} \leftarrow \sum_{k=1}^K \alpha^{(k)} g_i^{(r,k)}$.
    Compute $p_i^{(r)} \leftarrow \mathrm{softmax}(W_o z_i^{(r)} + b_o)$ and $\bar{p}_i \leftarrow R^{-1} \sum_{r=1}^R p_i^{(r)}$.
    Update the trainable actor and output parameters using $\{(\bar{p}_i, y_i)\}_{i=1}^n$.
  until the training stopping criterion is met
  Save the fitted parameters and actor weights $\{\hat{\alpha}^{(k)}\}_{k=1}^K$.
end procedure
procedure PREDICT(G_{new}, fitted model)
  Normalize the full test adjacency: $\mathcal{L}_{new} \leftarrow D_{new}^{-1/2} A_{new} D_{new}^{-1/2}$.
  for $k = 1, \ldots, K$ **do**
    Compute $H_{new}^{(k)} = \mathcal{L}_{new}^k X_{new} \hat{W}_A$ and $g_{new}^{(k)} \leftarrow \tau v_1(H_{new}^{(k)})$.
  end for
  Fuse $z_{new} \leftarrow \sum_{k=1}^K \hat{\alpha}^{(k)} g_{new}^{(k)}$.
  Compute $\hat{p}_{new} \leftarrow \mathrm{softmax}(\hat{W}_o z_{new} + \hat{b}_o)$.
  return $\hat{p}_{new}$ and $\hat{y}_{new} \leftarrow \arg\max_{1 \le c \le C} \hat{p}_{new,c}$.
end procedure

Pooling implementation and orientation. Power iteration approximates the dominant right singular vector of H. The signed-vector theory below specifies a fixed anchor and a positive alignment margin; a raw SVD sign or an arbitrary initialization does not by itself verify that condition. The statements concern exact singular directions. If the computed unit vectors have errors at most e_k and $e_{k,s}$ relative to the consistently oriented exact vectors, the actor readout discrepancy acquires the additional term $\tau(e_k + e_{k,s})$.

B FULL SUBSAMPLING RESULTS AND PROOFS

Theorem 3.1 is Theorem B.2 below, and Theorem 3.2 is the common-scale case $\tau_k = \tau$ of Theorem B.7. The full statements make the feature, normalization, gap, and sign assumptions explicit. Corollary B.11 extends the conditional risk-transfer argument to the probability aggregation used in the manuscript.

B.1 SETTING AND SCOPE

Let G be a fixed undirected simple graph on $[N] = \{1, \dots, N\}$, with $N \geq 3$, adjacency matrix A, and features $X \in \mathbb{R}^{N \times p}$. The entries of A are zero or one and $A_{ii} = 0$. Write $D = \operatorname{diag}(d_1, \dots, d_N)$ and $\mathcal{L} = D^{-1/2} A D^{-1/2}$. At an isolated vertex we set the corresponding entry of $D^{-1/2}$ to zero. The features are fixed before subsampling; features computed from the original graph are allowed, but they are inherited, rather than recomputed, on the subgraph.

Let S be a uniformly distributed subset of $[N]$ of size m, where $2 \leq m \leq N$, and let $P_S \in \mathbb{R}^{m \times N}$ be its row-selection matrix, with a fixed ordering of the selected vertices. Define

$$A_s = P_S A P_S^\top, \qquad X_s = P_S X, \qquad \mathcal{L}_s = D_s^{-1/2} A_s D_s^{-1/2},$$

where D_s uses the degrees *in the induced subgraph*, with the same zero-degree convention. For $k = 0, \dots, K$, set

$$F_k = \mathcal{L}^k X, \qquad F_{k,s} = \mathcal{L}_s^k P_S X.$$

For $k \geq 1$, the actor matrices are $H_k = F_k W_A$ and $H_{k,s} = F_{k,s} W_A$, where $W_A \in \mathbb{R}^{p \times h}$. There is no activation between these linear propagation steps. All matrix operator norms are induced by the Euclidean norm. Frobenius norms are denoted by $\| \cdot \|_F$.

The central quantities are the $p \times p$ matrices

$$Q_k = \frac{F_k^\top F_k}{N} = \frac{X^\top \mathcal{L}^{2k} X}{N}, \qquad Q_{k,s} = \frac{F_{k,s}^\top F_{k,s}}{m}. \tag{17}$$

Both have the same dimension, despite the different graph sizes. Moreover,

$$C_k := \frac{H_k^\top H_k}{N} = W_A^\top Q_k W_A, \qquad C_{k,s} := \frac{H_{k,s}^\top H_{k,s}}{m} = W_A^\top Q_{k,s} W_A. \tag{18}$$

Their leading eigenvectors are precisely the dominant right singular vectors used by RS-Pool; division by graph size leaves those vectors unchanged.

The sampling argument is related to the fixed-feature Gram-matrix concentration result from Lemma 9 in Appendix A of Bhattacharjee et al. (2024). Here sampling has fixed cardinality, and the features used by the GNN are recomputed through $\mathcal{L}_s$, which requires a separate propagation argument. An eigenvalue-approximation theorem alone does not control this propagation discrepancy. The sampling calculation is proved below, together with the additional propagation and prediction arguments.

B.2 A GENERAL GRAM-MATRIX PRESERVATION THEOREM

Lemma B.1 (Sampling a finite population)**.** *Let $Y_1, \dots, Y_N$ be fixed vectors in a finite-dimensional real inner product space, and let $\bar{Y} = N^{-1} \sum_i Y_i$. For a uniform subset S of size m, put $\bar{Y}_S = m^{-1} \sum_{i \in S} Y_i$. Then*

$$\mathbb{E}_S \|\bar{Y}_S - \bar{Y}\|^2 = \frac{N - m}{m(N-1)} \frac{1}{N} \sum_{i=1}^{N} \|Y_i - \bar{Y}\|^2. \tag{19}$$

In particular, if $f_i \in \mathbb{R}^p$, $\|f_i\|_2 \leq R$, and $Y_i = f_i f_i^\top$ with the Frobenius inner product, the right side is at most $R^4(N-m)/\{m(N-1)\}$.

Proof. Write $Z_i = Y_i - \bar{Y}$ and $I_i = \mathbf{1}\{i \in S\}$. Since $\sum_i Z_i = 0$, $\mathbb{E}I_i = m/N$, and $\mathbb{E}I_iI_j = m(m-1)/\{N(N-1)\}$ for $i \neq j$,

$$\mathbb{E}\Big\|\frac{1}{m}\sum_i I_i Z_i\Big\|^2 = \frac{1}{m^2}\left\{\frac{m}{N}\sum_i \|Z_i\|^2 + \frac{m(m-1)}{N(N-1)}\sum_{i\neq j}\langle Z_i, Z_j\rangle\right\}$$
$$= \frac{N-m}{mN(N-1)}\sum_i \|Z_i\|^2.$$

For the final assertion, use $N^{-1}\sum_i \|Y_i - \bar{Y}\|_F^2 = N^{-1}\sum_i \|Y_i\|_F^2 - \|\bar{Y}\|_F^2 \le R^4$. □

Theorem B.2 (Sampling plus propagation)**.** *Assume the setting of Appendix B.1. Suppose deterministic constants $R_k \ge 0$ satisfy $\max_i \|(F_k)_{i,:}\|_2 \le R_k$ for $k = 1, \dots, K$. Suppose also that deterministic $r_k \ge 0$ satisfy*

$$\mathbb{P}_S\left\{\frac{\|F_{k,s} - P_S F_k\|_{\mathrm{op}}}{\sqrt{m}} \le r_k \text{ for all } k = 1, \dots, K\right\} \ge 1 - \eta_p. \tag{20}$$

Here $0 \le \eta_p < 1$, and choose $0 < \eta_g < 1 - \eta_p$. Define

$$a_k = R_k^2\sqrt{\frac{K(N-m)}{\eta_g m(N-1)}}, \qquad \varepsilon_k = a_k + 2R_k r_k + r_k^2. \tag{21}$$

Then, with probability at least $1 - \eta_g - \eta_p$, simultaneously for all k,

$$\|Q_{k,s} - Q_k\|_{\mathrm{op}} \le \varepsilon_k. \tag{22}$$

On this same event, for every *matrix W_A satisfying $\|W_A\|_{\mathrm{op}} \le B$,*

$$\|C_{k,s} - C_k\|_{\mathrm{op}} \le B^2\varepsilon_k. \tag{23}$$

The event in (22) *does not depend on W_A.*

Proof. Let $\bar{F}_k = P_S F_k$ and $\widetilde{Q}_k = \bar{F}_k^\top \bar{F}_k/m$. Lemma B.1, $\|M\|_{\mathrm{op}} \le \|M\|_F$, and Markov's inequality give $\|\widetilde{Q}_k - Q_k\|_{\mathrm{op}} \le a_k$ with failure probability at most η_g/K. If $R_k = 0$ or $m = N$, this bound holds deterministically. A union bound gives all K inequalities with failure probability at most η_g; independence across actor orders is not required.

Write $E_k = F_{k,s} - \bar{F}_k$. Direct expansion gives

$$\|Q_{k,s} - \widetilde{Q}_k\|_{\mathrm{op}} \le \frac{2\|\bar{F}_k\|_{\mathrm{op}}\|E_k\|_{\mathrm{op}} + \|E_k\|_{\mathrm{op}}^2}{m}$$
$$\le 2R_k r_k + r_k^2$$

on the event in (20), since $\|\bar{F}_k\|_{\mathrm{op}} \le \|\bar{F}_k\|_F \le \sqrt{m}R_k$. The triangle inequality and a union bound prove (22). Equation (23) follows from $C_{k,s} - C_k = W_A^\top(Q_{k,s} - Q_k)W_A$ and submultiplicativity. □

B.3 Explicit propagation control for induced subgraphs

The next proposition supplies (20) using observable graph quantities. It does not assume that the sampled graph is regular. All probabilities remain conditional on the original graph and its features.

Lemma B.3 (A hypergeometric tail bound)**.** *Let Z count the successes in a uniform sample of size t without replacement from a population of size n containing d successes. Put $\mu = td/n$. For $0 < u < 1$,*

$$\mathbb{P}\{|Z - \mu| \ge u\mu\} \le 2\exp(-u^2\mu/3).$$

Proof. For any real λ, averaging products of t distinct elements of the nonnegative population $e^{\lambda x_1}, \dots, e^{\lambda x_n}$, where $x_i \in \{0, 1\}$, yields

$$\mathbb{E}e^{\lambda Z} \le \left(1 + \frac{d}{n}(e^\lambda - 1)\right)^t \le \exp\{\mu(e^\lambda - 1)\}.$$

The first inequality is the elementary symmetric-mean inequality: for a fixed sum of nonnegative entries, averaging any two entries cannot decrease their elementary symmetric polynomial, so its maximum occurs when all entries are equal. Chernoff's argument, with $\lambda = \log(1+u) > 0$ for the upper tail and $\lambda = \log(1-u) < 0$ for the lower tail, gives exponents $-\mu\{(1+u)\log(1+u) - u\}$ and $-\mu\{u + (1-u)\log(1-u)\}$, respectively. Elementary differentiation shows that the bracketed terms are at least $u^2/3$ and $u^2/2$ on $0 < u < 1$. Adding the two bounds proves the claim. □

Proposition B.4 (Propagation with recomputed degree normalization)**.** *Under the setup of Appendix B.1, assume additionally that:*

(P1) $d_{\min} := \min_i d_i > 0$*, and* $d_{\max}/d_{\min} \le \kappa$ *for a specified* $\kappa \ge 1$*;*

(P2) $\max_i \|X_{i,:}\|_2 \le M$*;*

(P3) $2 \le m < N$*. Let* $q = (m-1)/(N-1)$ *and choose* $\eta_d, \eta_z > 0$ *with* $\eta_d + \eta_z < 1$*. Assume*

$$qd_{\min} \ge 12\log(2m/\eta_d). \tag{24}$$

Set $R = M\sqrt{\kappa}$*,* $c_N = (N-1)/(N-2)$*, and*

$$u = \sqrt{\frac{3\log(2m/\eta_d)}{qd_{\min}}}, \qquad r_k = R\left\{3ku + \sqrt{\frac{kKc_N(1-q)}{\eta_z qd_{\min}}}\right\}, \quad 1 \le k \le K. \tag{25}$$

Then $\max_i \|(F_l)_{i,:}\|_2 \le R$ *for every* $l \ge 0$*, and* (20) *holds with* $\eta_p = \eta_d + \eta_z$ *and these* r_k*. If* $m = N$*, propagation is exact and one may instead set* $r_k = 0$ *and* $\eta_p = 0$*, without* (24)*.*

Proof. We separate feature bounds, degree normalization, and message sampling.

Feature bounds. The random-walk matrix $P = D^{-1}A$ is row stochastic and $\mathcal{L}^l = D^{1/2}P^lD^{-1/2}$. Therefore

$$\|(F_l)_{i,:}\|_2 \le M\sqrt{d_i}\sum_j (P^l)_{ij}/\sqrt{d_j} \le M\sqrt{d_i/d_{\min}} \le R.$$

This also holds for $l = 0$. Both $\mathcal{L}$ and $\mathcal{L}_s$ have operator norm at most one. For completeness, for a normalized adjacency matrix L and any x, the inequality $2|ab| \le a^2 + b^2$, summed over edges, gives $|x^\top Lx| \le \|x\|_2^2$; isolated rows contribute zero.

Degree normalization. Conditional on $i \in S$, the subgraph degree d_i^s is hypergeometric with population size $N-1$, sample size $m-1$, and d_i successes. Its conditional mean is qd_i. Lemma B.3 and a union bound over the original vertices, retaining their inclusion probabilities, give

$$\begin{aligned}\mathbb{P}\{\exists i \in S:\ |d_i^s - qd_i| \ge uqd_i\} &\le \sum_{i=1}^N \frac{m}{N}\, 2e^{-u^2qd_i/3} \\ &\le 2me^{-u^2qd_{\min}/3} = \eta_d.\end{aligned}$$

On the complementary event $\mathcal{D}$, all sampled degrees are positive, and $u \le 1/2$ by (24). Define

$$\widehat{\mathcal{L}}_s = q^{-1}P_S\mathcal{L}P_S^\top, \qquad T = \mathrm{diag}_{i\in S}\sqrt{d_i^s/(qd_i)}.$$

Entrywise calculation gives $\widehat{\mathcal{L}}_s = T\mathcal{L}_sT$. On $\mathcal{D}$, $\|T - I\|_{\mathrm{op}} \le u$ and $\|T\|_{\mathrm{op}} \le \sqrt{1+u}$, hence

$$\|\mathcal{L}_s - \widehat{\mathcal{L}}_s\|_{\mathrm{op}} \le \|T-I\|_{\mathrm{op}}(1 + \|T\|_{\mathrm{op}})\|\mathcal{L}_s\|_{\mathrm{op}} \le 3u. \tag{26}$$

Sampling the messages. For $l = 0, \dots, K-1$, put

$$Z_l = \widehat{\mathcal{L}}_sP_SF_l - P_S\mathcal{L}F_l.$$

For an original vertex $i \in S$, its row in Z_l is

$$(Z_l)_{i,:} = \sum_{j\ne i}\left(\frac{\mathbf{1}\{j \in S\}}{q} - 1\right)\mathcal{L}_{ij}(F_l)_{j,:},$$

where the row is indexed by its original vertex label. This identity uses the zero diagonal of A. Conditional on $i \in S$, the set $S \setminus \{i\}$ is a uniform sample of $m-1$ vertices from $[N] \setminus \{i\}$. For each $j \neq i$, define the fixed vector

$$w_j = \mathcal{L}_{ij}(F_l)_{j,:},$$

where $(F_l)_{j,:}$ denotes the jth row of F_l. Lemma B.1, applied to these $N-1$ vectors, shows that $q^{-1}\sum_{j\in S\setminus\{i\}} w_j$ is an unbiased estimator of $\sum_{j\neq i} w_j$. Its mean squared error is bounded by

$$\begin{aligned}\mathbb{E}\{\|(Z_l)_{i,:}\|_2^2 \mid i \in S\} &\leq \frac{c_N(1-q)}{q}\sum_{j\neq i}\|w_j\|_2^2 \\ &\leq \frac{c_N(1-q)R^2}{q}\sum_j \mathcal{L}_{ij}^2 \leq \frac{c_N(1-q)R^2}{qd_{\min}} =: V.\end{aligned}$$

The last inequality follows from $\sum_j \mathcal{L}_{ij}^2 = \sum_{j:A_{ij}=1}(d_i d_j)^{-1} \leq d_{\min}^{-1}$. Since $\mathbb{P}(i \in S) = m/N$, summing over i gives $\mathbb{E}\{\|Z_l\|_F^2/m\} \leq V$. Consequently, with probability at least $1-\eta_z$,

$$\sum_{l=0}^{K-1}\frac{\|Z_l\|_{\mathrm{op}}^2}{m} \leq \sum_{l=0}^{K-1}\frac{\|Z_l\|_F^2}{m} \leq \frac{KV}{\eta_z}. \tag{27}$$

This is Markov's inequality applied to the sum; no independence of its terms, or of the sampled vertices' messages, is assumed.

Propagation across orders. Let $E_l = F_{l,s} - P_S F_l$, so $E_0 = 0$. The exact recurrence is

$$E_{l+1} = \mathcal{L}_s E_l + (\mathcal{L}_s - \widehat{\mathcal{L}}_s)P_S F_l + Z_l.$$

Using $\|\mathcal{L}_s\|_{\mathrm{op}} \leq 1$, (26), and $\|P_S F_l\|_{\mathrm{op}}/\sqrt{m} \leq R$, on $\mathcal{D}$ and (27) we obtain, simultaneously for $1 \leq k \leq K$,

$$\begin{aligned}\frac{\|E_k\|_{\mathrm{op}}}{\sqrt{m}} &\leq 3kuR + \sum_{l=0}^{k-1}\frac{\|Z_l\|_{\mathrm{op}}}{\sqrt{m}} \\ &\leq 3kuR + \sqrt{\frac{kKV}{\eta_z}} = r_k.\end{aligned}$$

The second inequality follows from the Cauchy–Schwarz inequality. A union bound over the two events completes the proof. If $m = N$, all operators and features agree up to their common vertex ordering, so $E_k = 0$. □

Corollary B.5 (An explicit Gram-matrix bound)**.** *Under Proposition B.4, take $\eta_g+\eta_d+\eta_z = \eta < 1$, with all three terms positive. For $1 \leq k \leq K$, let r_k be as in* (25) *and set*

$$\varepsilon_k = R^2\sqrt{\frac{K(N-m)}{\eta_g m(N-1)}} + 2Rr_k + r_k^2. \tag{28}$$

Then (22) *and* (23) *hold simultaneously with probability at least $1-\eta$.*

Proof. The conclusion follows from Theorem B.2 with $R_k = R$ and $\eta_p = \eta_d + \eta_z$, using the propagation bounds established in Proposition B.4. □

Remark B.6 (Interpretation of the sufficient conditions). For fixed K, M, κ and fixed positive failure probabilities, along a sequence with $m \to \infty$ and $qd_{\min}/\log m \to \infty$,

$$\varepsilon_k = O\left(m^{-1/2} + \sqrt{\frac{\log m}{qd_{\min}}}\right).$$

This is an order statement for the displayed deterministic error bounds, at the chosen confidence level. When $d_{\min} \geq cN$ for a fixed $c > 0$, it is $O(\sqrt{\log m/m})$. The constants are conservative. Graphs with bounded degrees need not satisfy these sufficient conditions; the result makes no vanishing-error claim for that regime. Bounded features also do not automatically cover unbounded Gaussian features without a separate high-probability truncation or tail argument.

B.4 From Gram matrices to singular pooling and classification

We now hold all network parameters fixed. They may be learned from an independent training sample, in which case the statements are conditional on that fitted network. We compare that same network on G and G_s.

For $h \geq 2$, let v_k and $v_{k,s}$ be unit leading eigenvectors of C_k and $C_{k,s}$. The readouts and logits are

$$g_k = \tau_k v_k, \quad g_{k,s} = \tau_k v_{k,s}, \qquad s(G) = W_o \sum_{k=1}^{K} \alpha_k g_k + b_o, \quad s(G_s) = W_o \sum_{k=1}^{K} \alpha_k g_{k,s} + b_o. \tag{29}$$

Here $W_o \in \mathbb{R}^{C \times h}$, $C \geq 2$, $\tau_k > 0$, and $\alpha_k \in \mathbb{R}$ are finite, shared across the two inputs, and held fixed. Set $A_o = \|W_o\|_{2\to\infty}$, the maximum Euclidean row norm of W_o. Graph-dependent fusion weights, adaptive pooling scales, or numerical singular-vector errors would require additional error terms.

Theorem B.7 (RS-Pool and prediction stability)**.** *Suppose deterministic ε_k satisfy* (22) *jointly with probability at least $1-\eta$, and $\|W_A\|_{\mathrm{op}} \leq B$. Assume the following for each k:*

(S1) The full-graph hidden Gram matrix has leading gap

$$\delta_k = \lambda_1(C_k) - \lambda_2(C_k) > 0.$$

(S2) Write $b_k = B^2\varepsilon_k$. The perturbation is small enough that $b_k \leq \delta_k/4$.

(S3) A fixed unit anchor $a_k^{\mathrm{anc}} \in \mathbb{R}^h$ determines the sign of each leading eigenvector by requiring a positive inner product with the anchor. The full-graph vector satisfies $(a_k^{\mathrm{anc}})^\top v_k \geq \zeta_k > 0$, and

$$\xi_k := \frac{2\sqrt{2}b_k}{\delta_k} < \zeta_k.$$

On inputs with a nonunique leading vector or zero anchor inner product, any fixed measurable fallback rule defines the classifier; such inputs do not occur on the success event considered below.

Then with probability at least $1-\eta$, for every k,

$$\|v_{k,s} - v_k\|_2 \leq \xi_k, \qquad \|g_{k,s} - g_k\|_2 \leq \tau_k \xi_k, \tag{30}$$

and

$$\|s(G_s) - s(G)\|_\infty \leq t(G) := A_o \sum_{k=1}^{K} |\alpha_k| \tau_k \xi_k. \tag{31}$$

Let $\gamma(G) = s_{(1)}(G) - s_{(2)}(G)$ be the difference between the largest and second-largest full-graph logits. With a common deterministic tie-breaking rule for prediction, if $\gamma(G) > 2t(G)$, then

$$\mathbb{P}_S\{\arg\max_c s_c(G_s) \neq \arg\max_c s_c(G)\} \leq \eta.$$

The same predicted labels result from applying softmax first.

Proof. We prove the bounds on the event where (22) holds simultaneously for all k. By assumption, this event has probability at least $1-\eta$. For a fixed k, we suppress its index and write $C_s = C + E$, where $\|E\|_{\mathrm{op}} \leq b$. The variational characterization of eigenvalues gives $|\lambda_j(C_s) - \lambda_j(C)| \leq b$. Hence the sampled leading gap is at least $\delta - 2b \geq \delta/2 > 0$.

Let w be its leading unit eigenvector, initially oriented so that $v^\top w \geq 0$, and put $u_\perp = (I - vv^\top)w$. Projecting $(C+E)w = \lambda_1(C_s)w$ onto $v^\perp$ gives

$$(\lambda_1(C_s)I - C)u_\perp = (I - vv^\top)Ew.$$

On $v^\perp$ the operator on the left has smallest eigenvalue at least $\delta - b$, so $\|u_\perp\|_2 \leq b/(\delta - b) \leq 2b/\delta$. Since $v^\top w \geq 0$,

$$\|w - v\|_2^2 = 2(1 - v^\top w) \leq 2\{1 - (v^\top w)^2\} = 2\|u_\perp\|_2^2.$$

Thus $\|w - v\|_2 \le \xi$. The anchor condition implies $(a^{\mathrm{anc}})^\top w \ge \zeta - \xi > 0$, so this aligned vector is exactly the vector selected by the stated sign convention. This proves (30). This is a direct one-dimensional eigenvector perturbation argument of the Davis–Kahan type; see also Yu et al. (2015).

The triangle inequality in (29) gives (31). If c_* maximizes the full-graph logits, then for every $c \neq c_*$ on the same event,

$$s_{c_*}(G_s) - s_c(G_s) \ge s_{c_*}(G) - s_c(G) - 2t(G) \ge \gamma(G) - 2t(G) > 0.$$

Therefore the predicted class cannot change on this event. Softmax preserves the ordering of logits. □

Remark B.8 (Which gap and which sign?). The relevant gap is $\delta_k = \{\sigma_1(H_k)^2 - \sigma_2(H_k)^2\}/N$; it is a hidden-feature gap, not an adjacency eigenvalue gap. An arbitrary numerical SVD sign does not satisfy (S3) automatically. Alignment by an unobservable pairwise sign is sufficient for a subspace statement, but not for the signed-vector classifier in (29). An implementation using an anchor must satisfy the stated nonzero alignment condition; otherwise only the bound modulo sign follows from the proof.

Corollary B.9 (Population accuracy discrepancy)**.** *Let (G, Y) be random. Given G, sample S uniformly among vertex subsets of the prescribed size. The same fixed network parameters are used for G and G_s. Suppose the assumptions of Theorem B.7 hold for almost every G, with conditional failure probability at most η, where η does not depend on G. The threshold $t(G)$ in* (31) *is deterministic once G and the network parameters are fixed. Define*

$$\mathrm{Acc}_{\mathrm{full}} = \mathbb{P}\{\widehat{y}(G) = Y\}, \qquad \mathrm{Acc}_{\mathrm{sub}} = \mathbb{P}\{\widehat{y}(G_s) = Y\},$$

using the parent graph's label *for both predictions. Then*

$$|\,\mathrm{Acc}_{\mathrm{sub}} - \mathrm{Acc}_{\mathrm{full}}\,| \le \mathbb{P}\{\gamma(G) \le 2t(G)\} + \eta. \tag{32}$$

If $t(G) \le t_m$ almost surely and $\mathbb{P}\{\gamma(G) \le u\} \le C_\gamma u^\beta$ for $0 < u \le u_0$, where $\beta > 0$, then for $0 < 2t_m \le u_0$ the right side is at most $C_\gamma(2t_m)^\beta + \eta$.

Proof. The absolute difference of the two correctness indicators is bounded by the indicator that the predictions disagree. Conditional on each G, disagreement requires either failure of the preservation event or $\gamma(G) \le 2t(G)$. Integrate this conditional bound over (G, Y). The final assertion follows by the assumed margin bound. □

B.5 Conditional transfer of a training objective

The preceding results do not compare independently optimized parameter vectors. The following elementary proposition makes the extra requirements explicit. It concerns surrogate risk, not directly zero–one accuracy.

Proposition B.10 (Conditional transfer of a training objective)**.** *Let Θ be a parameter class, and let $\ell(s, y) \in [0, M_\ell]$ be L_ℓ-Lipschitz in s with respect to $\|\cdot\|_\infty$. Assume that a single event $\mathcal{E}$ in the joint distribution of (G, Y, S), with $\mathbb{P}(\mathcal{E}^c) \le \eta$, satisfies*

$$\sup_{\theta \in \Theta} \|s_\theta(G_s) - s_\theta(G)\|_\infty \le T \quad \text{on } \mathcal{E},$$

where T is a deterministic constant. Define

$$R_f(\theta) = \mathbb{E}\ell(s_\theta(G), Y), \qquad R_s(\theta) = \mathbb{E}\ell(s_\theta(G_s), Y), \qquad a = L_\ell T + M_\ell \eta.$$

Let $\widehat{R}_s$ be the empirical subgraph objective. On any training-data event where $\sup_{\theta\in\Theta} |\widehat{R}_s(\theta) - R_s(\theta)| \le b_n$, if $\widehat{R}_s(\widehat{\theta}_s) \le \inf_{\theta\in\Theta} \widehat{R}_s(\theta) + \epsilon_{\mathrm{opt}}$, then

$$R_f(\widehat{\theta}_s) - \inf_{\theta \in \Theta} R_f(\theta) \le 2a + 2b_n + \epsilon_{\mathrm{opt}}. \tag{33}$$

Proof. For each θ, split the expectation of the loss difference over $\mathcal{E}$ and its complement. This gives $\sup_\theta |R_s(\theta) - R_f(\theta)| \le a$. For any $\theta \in \Theta$, on the asserted training-data event,

$$\begin{aligned} R_f(\widehat{\theta}_s) &\le R_s(\widehat{\theta}_s) + a \le \widehat{R}_s(\widehat{\theta}_s) + b_n + a \\ &\le \widehat{R}_s(\theta) + \epsilon_{\mathrm{opt}} + b_n + a \\ &\le R_s(\theta) + \epsilon_{\mathrm{opt}} + 2b_n + a \\ &\le R_f(\theta) + \epsilon_{\mathrm{opt}} + 2b_n + 2a. \end{aligned}$$

Since this inequality holds for every $\theta \in \Theta$, taking the infimum of its right-hand side yields (33). □

To obtain the event in Proposition B.10 from the earlier theorems, weight bounds, hidden-feature gaps, anchor alignments, scales, and fusion weights must be controlled *uniformly over* Θ. This is substantially stronger than verifying them for one fitted network. The quantity b_n is an additional uniform estimation bound; no rate for it is asserted here. With n independent parent graphs and multiple views per parent, a natural empirical objective averages within each parent and then across parents. Views sharing a parent must not be counted as independent graph observations. Corollary B.11 handles averaging probabilities before applying the loss, as in equation (5).

Corollary B.11 (Transfer for averaged view probabilities)**.** *Let $S^{(1)}, \dots, S^{(R)}$ be sampled views of a parent graph, and set*

$$p_\theta(G) = \mathrm{softmax}(s_\theta(G)), \qquad \bar{p}_\theta = \frac{1}{R}\sum_{r=1}^{R} \mathrm{softmax}(s_\theta(G_{S^{(r)}})).$$

Suppose an event $\mathcal{E}_R$ of probability at least $1-\eta_R$ satisfies

$$\sup_{\theta\in\Theta} \max_{1\le r\le R} \|s_\theta(G_{S^{(r)}}) - s_\theta(G)\|_\infty \le T.$$

If $\ell_p(p,y) \in [0, M_\ell]$ is L_p-Lipschitz in probability vectors under $\|\cdot\|_\infty$, define

$$R_f^p(\theta) = \mathbb{E}\ell_p(p_\theta(G), Y), \quad R_{s,R}^p(\theta) = \mathbb{E}\ell_p(\bar{p}_\theta, Y), \quad a_R = \tfrac{1}{2}L_p T + M_\ell \eta_R.$$

Assume n independent parent graphs (with their associated view collections) and an event on which $\sup_\theta |\widehat{R}_{s,R}^p(\theta) - R_{s,R}^p(\theta)| \le b_n$. An ϵ_{opt}-approximate empirical minimizer $\widehat{\theta}$ then satisfies

$$R_f^p(\widehat{\theta}) - \inf_{\theta\in\Theta} R_f^p(\theta) \le 2a_R + 2b_n + \epsilon_{\mathrm{opt}}.$$

The views within a parent need not be independent. If the single-view uniform event fails with probability at most η, a union bound gives $\eta_R \le R\eta$.

Proof. For $p = \mathrm{softmax}(s)$, the absolute row sum of the Jacobian in row j is $2p_j(1-p_j) \le 1/2$. Integrating along the line segment between two logit vectors proves that softmax is $1/2$-Lipschitz from ℓ_∞ to ℓ_∞. Thus on $\mathcal{E}_R$, uniformly over θ,

$$\|\bar{p}_\theta - p_\theta(G)\|_\infty \le \frac{1}{R}\sum_r \tfrac{1}{2}\|s_\theta(G_{S^{(r)}}) - s_\theta(G)\|_\infty \le T/2.$$

Splitting expected loss differences over $\mathcal{E}_R$ and its complement yields $\sup_\theta |R_{s,R}^p(\theta) - R_f^p(\theta)| \le a_R$. The claimed excess-risk bound follows by repeating the empirical-minimization comparison in the proof of Proposition B.10. □

Cross-entropy on averaged probabilities. Cross-entropy is not bounded and Lipschitz on the entire probability simplex. A sufficient restriction is a uniform logit bound $\|s_\theta(\cdot)\|_\infty \le Z$ on both graphs and views. Then each class probability is at least e^{-2Z}/C, so the loss is bounded by $M_\ell = 2Z + \log C$. More directly, on $\mathcal{E}_R$ each view's class probability is between e^{-2T} and e^{2T} times the full-graph probability. The average obeys the same ratio bound, and hence the two cross-entropies differ by at most $2T$. The corollary therefore also holds for cross-entropy with $a_R = 2T + (2Z + \log C)\eta_R$. This condition concerns bounded logits and does not assert that it has been verified for the reported experiments.

Conditions and proof for Theorem 3.3. The following conditions give a precise classification-rate consequence of Corollary B.11. They specify the estimation and training requirements in addition to the graph-subsampling assumptions. Fix a margin level $\nu \in (0,1)$ and a binary classifier with probability vector $p_\theta(G)$, and write

$$f_\theta(G) = p_{\theta,+}(G) - p_{\theta,-}(G), \qquad \ell_\nu(p,y) = \min\{1, \max\{0, 1 - y(p_+ - p_-)/\nu\}\}.$$

The loss lies in $[0,1]$, is $2/\nu$-Lipschitz in p under the infinity norm, and upper bounds $\mathbf{1}\{yf_\theta(G) \le 0\}$. For R sampled views, let $\bar{p}_\theta = R^{-1}\sum_{r=1}^R p_\theta(G_{S^{(r)}})$, as in equation (5). Define

$$R_f^\nu(\theta) = \mathbb{E}\ell_\nu(p_\theta(G), Y), \qquad R_{s,R}^\nu(\theta) = \mathbb{E}\ell_\nu(\bar{p}_\theta, Y).$$

The empirical risk $\widehat{R}_{s,R}^\nu$ averages the latter loss over the n observed parent graphs. Assume:

(G1) The labeled parent graphs and their associated view collections are independent and identically distributed. The number R of views and the margin ν are fixed. The parameter class Θ_n is specified independently of the training sample, and the same parameter vector is used on a parent graph and its views.

(G2) On an event of probability at least $1 - \eta_n$ for a fresh parent graph and its view collection,

$$\sup_{\theta\in\Theta_n} \max_{1\le r\le R} \|s_\theta(G_{S^{(r)}}) - s_\theta(G)\|_\infty \le T_n, \qquad a_n := T_n/\nu + \eta_n = O(n^{-1/2}).$$

The norm, gap, orientation, and fusion conditions must hold uniformly over Θ_n when this event is obtained from the preceding spectral results.

(G3) For each fixed confidence level $\delta \in (0, 1)$, there is a training-data event of probability at least $1 - \delta$ on which

$$\sup_{\theta\in\Theta_n} |\widehat{R}^\nu_{s,R}(\theta) - R^\nu_{s,R}(\theta)| \le b_n, \qquad b_n = O(n^{-1/2}).$$

This is an estimation assumption for the actual singular-pooling class. Establishing this condition requires a separate capacity analysis of that class; the subsampling results control the distinct term a_n.

(G4) The optimal full-graph margin risk $r^*_n := \inf_{\theta\in\Theta_n} R^\nu_f(\theta)$ is $O(n^{-1/2})$, including the realizable case $r^*_n = 0$. On the same training-data event, the returned parameter satisfies

$$\widehat{R}^\nu_{s,R}(\widehat{\theta}_n) \le \inf_{\theta\in\Theta_n} \widehat{R}^\nu_{s,R}(\theta) + \epsilon_{\mathrm{opt},n}, \qquad \epsilon_{\mathrm{opt},n} = O(n^{-1/2}).$$

This condition is on the attained margin loss; it is not automatically implied by minimizing cross-entropy.

All rates hold along the specified sequence of graph distributions, sampling designs, and parameter classes; constants may depend on the fixed confidence level and model bounds. Condition (G2) states explicitly the rate required of the sampling discrepancy. It does not follow merely by increasing the number of training graphs at a fixed graph size and sampling ratio.

Proof of Theorem 3.3. Apply Corollary B.11 with $L_p = 2/\nu$, $M_\ell = 1$, $T = T_n$, and $\eta_R = \eta_n$. Its population-risk comparison gives $\sup_\theta |R^\nu_{s,R}(\theta) - R^\nu_f(\theta)| \le a_n$. On the event in (G3)–(G4), the empirical-minimization comparison therefore yields

$$\mathbb{P}\{Y f_{\widehat{\theta}_n}(G) \le 0\} \le R^\nu_f(\widehat{\theta}_n) \le r^*_n + 2a_n + 2b_n + \epsilon_{\mathrm{opt},n}. \tag{34}$$

Each term on the right is $O(n^{-1/2})$ under (G2)–(G4), proving the claimed rate. The expectation in the risk is over an independent test graph conditional on the fitted parameters. The empirical risk treats each parent and its view collection as one observation, so independence of views within a parent is not required. □

B.6 Actor-level robustness to feature perturbations

This section gives the supporting conditions and proof for the full-graph actor robustness result in Theorem 3.4. Singular pooling is applied separately to each MaGNet actor, with the graph and fitted parameters held fixed while the node features are perturbed.

Proposition B.12 (A conditional feature-perturbation bound)**.** *Fix a graph with nonnegative normalized adjacency $\mathcal{L}$ and actor $H_k = \mathcal{L}^k X W_A$. Suppose $\|W_A\|_{\mathrm{op}} \le B$ and the perturbed features satisfy $\max_i \|\widetilde{X}_{i,:} - X_{i,:}\|_2 \le \epsilon$. Let $\widetilde{H}_k = \mathcal{L}^k \widetilde{X} W_A$, $w_k = \mathcal{L}^k \mathbf{1}$, and*

$$e_k = B\epsilon\|w_k\|_2, \qquad b^X_k = \frac{2\|H_k\|_{\mathrm{op}} e_k + e_k^2}{N}.$$

Let $\delta_k > 0$ be the leading gap of $H_k^\top H_k/N$ and use the fixed anchor from Theorem B.7, with full-graph alignment at least $\zeta_k > 0$. If $b^X_k \le \delta_k/4$ and $2\sqrt{2}b^X_k/\delta_k < \zeta_k$, then

$$\|\tau_k v_1(\widetilde{H}_k) - \tau_k v_1(H_k)\|_2 \le \frac{2\sqrt{2}\tau_k b^X_k}{\delta_k}.$$

Without these small-perturbation conditions, the universal bound is $2\tau_k$. The same bounds hold for expected readout discrepancy when every realized perturbation satisfies the stated conditions.

Proof. Nonnegativity of $\mathcal{L}^k$ and the rowwise perturbation budget imply

$$\|\widetilde{H}_{k,i,:} - H_{k,i,:}\|_2 \le B\epsilon \sum_j (\mathcal{L}^k)_{ij}.$$

Summing squared row bounds gives $\|\widetilde{H}_k - H_k\|_{\mathrm{op}} \le \|\widetilde{H}_k - H_k\|_F \le e_k$. Writing $E = \widetilde{H}_k - H_k$ and expanding $(H_k + E)^\top (H_k + E) - H_k^\top H_k$ bounds the normalized Gram difference by b_k^X. The eigenvector and anchor argument in the proof of Theorem B.7 gives the displayed bound. The bound $2\tau_k$ follows from the triangle inequality for two unit vectors. An almost-sure deterministic bound also bounds its expectation. □

Proof of Theorem 3.4. Set $\tau_k = \tau$ in Proposition B.12. The displayed spectral bound and the universal 2τ bound hold simultaneously under its hypotheses, so their minimum also bounds each realized discrepancy. Taking expectations over any perturbation distribution supported on the stated set gives the theorem. In terms of singular values, $\delta_k = (\sigma_1(H_k)^2 - \sigma_2(H_k)^2)/N$; all graph operators and walk weights here belong to the full graph. □

B.7 Connection with spectral subsampling results

For $\mathcal{L} = U\Lambda U^\top$, with orthonormal columns u_j,

$$Q_k = \sum_{j=1}^{N} \lambda_j^{2k} a_j a_j^\top, \qquad a_j = N^{-1/2} X^\top u_j.$$

Consequently, eigenvalues and feature projections onto the associated eigenspaces jointly determine the relevant Gram matrix. This expression does not depend on eigenvector signs; repeated eigenvalues may be grouped using their spectral projectors. A different spectral subsampling theorem may replace Proposition B.4 if it gives the required Q_k bound, or a bound on $F_{k,s} - P_S F_k$ under the actual normalization. Preserving only an eigenvalue list, or treating $\mathcal{L}_s$ as $P_S \mathcal{L} P_S^\top$, is insufficient for that substitution.

The complete conclusion proved here is preservation of a fixed classifier's accuracy under the specified sampling, feature, gap, orientation, and margin conditions. It does not assert that the classifier has high accuracy in the first place, or that these conditions automatically hold on a benchmark.

C Experimental specifications and complete simulation results

The original experimental design compares Av-GCN, RS-GCN, MaRS-GNN, and SMaRS-GNN, with common data splits within each experiment. Evaluation uses clean graphs, random perturbations, and PGD attacks, with the reported perturbation budget $\epsilon = 20\%$. Simulation results average five independent runs. Training times are reported in minutes.

C.1 Features and sampling

For both simulation experiments, node features are constructed after the graph adjacency matrix is generated, using only observable graph-derived quantities. For vertex v in graph i, the feature vector is

$$X_{iv} = \left(1, \frac{d_{iv}}{N_i - 1} + \eta_{iv}^{(1)}, \mathrm{clust}_{iv} + \eta_{iv}^{(2)}, \xi_{iv}^{(1)}, \xi_{iv}^{(2)}, \xi_{iv}^{(3)}\right),$$

where d_{iv} is the degree of vertex v, clust_{iv} is the local clustering coefficient, $\eta_{iv}^{(1)}, \eta_{iv}^{(2)} \sim N(0, 0.1^2)$, and $\xi_{iv}^{(j)} \sim N(0, 0.4^2)$ for $j = 1, 2, 3$. Thus, each node has a six-dimensional feature vector. The first two nonconstant features carry graph-structural information through degree and local clustering, while the last three features are noise variables.

For SMaRS-GNN, we consider subsampling ratios $\rho = 0.2, 0.4, 0.6$. A sampled subnetwork is formed by selecting $\lfloor \rho N_i \rfloor$ vertices uniformly at random from a training graph and taking the induced subgraph. The label of each sampled subnetwork is inherited from the original graph. We use two replications for subsampling, and then aggregate the results.

C.2 RDPG DATA GENERATION

The first simulation examines a binary graph classification problem generated from two closely related random dot product graph (RDPG) models. For each simulation setting, we generate 500 graphs from each class, giving 1000 graphs in total. The class label for graph i is denoted by $Y_i \in \{0, 1\}$. The number of vertices $N_i = |V_i|$ is sampled uniformly from the graph-size interval specified in Table 3. Each vertex belongs to one of two latent components. Conditional on class label $Y_i = y$, vertex v is assigned a latent component C_{iv}, which takes value 1 or 2 (indicating the row of the latent matrix) with probability 0.5 each. The class-specific latent matrices are

$$B_0 = \begin{pmatrix} 0.52+\delta & 0.18-\delta & 0.12 \\ 0.18-\delta & 0.52+\delta & 0.12 \end{pmatrix}, \qquad B_1 = \begin{pmatrix} 0.52-\delta & 0.18+\delta & 0.12 \\ 0.18+\delta & 0.52-\delta & 0.12 \end{pmatrix},$$

where $\delta \geq 0$ controls the separation between the two classes. Larger values of δ yield more distinguishable graph distributions. Given the latent component assignment, the latent position of vertex v in graph i is generated as

$$Z_{iv} = B_{Y_i, C_{iv}} + \varepsilon_{iv}, \qquad \varepsilon_{iv} \sim N(0, \sigma_Z^2 I_3),$$

with $\sigma_Z = 0.10$. That is, for a vertex v in graph i, we consider the C_{iv}th row of the matrix B_{Y_i} and add noise ϵ_{iv} to obtain the latent vector Z_{iv}. For latent positions with valid dot products in $[0, 1]$, edges are generated independently according to the RDPG model:

$$\mathbb{P}\{A_i(u, v) = 1 \mid Z_i\} = Z_{iu}^\top Z_{iv}, \qquad u < v,$$

with $A_i(v, v) = 0$ and $A_i(u, v) = A_i(v, u)$. The latent positions are used only to generate the graph adjacency matrices and are not supplied to the classifiers.

C.3 NW-ER DATA GENERATION

The second simulation evaluates performance when class information is driven by higher-order network structure. We generate graphs from the Newman–Watts Erdős–Rényi (NW-ER) model, discussed explicitly in Lovekar et al. (2025). In this model, the parameter β controls the strength of the Newman–Watts small-world component relative to the Erdős–Rényi baseline. When $\beta = 0$, the model reduces to an Erdős–Rényi graph, while larger values of β introduce stronger small-world structure through additional local and shortcut connectivity. In the simulation, graphs from class 0 are generated with $\beta = 0$, and graphs from class 1 are generated with the positive β values shown in Table 4, namely $\beta = 0.1$ or $\beta = 0.25$. The classification task is therefore to distinguish graphs containing higher-order small-world structure from Erdős–Rényi-like graphs with no such signal. We use the graph size and signal strength settings reported in Table 4. The feature construction and SMaRS-GNN sampling ratios are the same as in the RDPG simulation.

C.4 BENCHMARK DATASETS AND REPORTING SCOPE

Reddit-B consists of discussion threads: nodes are users and an edge indicates that one user responded to another. D&D and PROTEINS are protein-structure graph datasets. All benchmark entries remain in Table 2. The supplied results contain mean accuracies and times but no uncertainty estimates, so the comparisons are descriptive rather than tests of statistical significance.

D ADDITIONAL DISCUSSION

Sampling design. Uniform node sampling is simple and supports the fixed-size concentration argument. Other choices may retain different aspects of graph structure: degree-based sampling emphasizes connected vertices, random walks preserve local neighborhoods, and cluster sampling can retain community structure (Hamilton et al., 2017; Chen et al., 2018; Chiang et al., 2019; Zeng et al., 2020). The current theorem applies to uniform sampling without replacement; changing the sampling design requires revisiting the sampling and normalization bounds.

Actor order and pooling scale. Increasing K includes more neighborhood orders but adds computation and may introduce redundant representations. Validation performance and critic weights

		Av-GCN	RS-GCN	MaRS	SMaRS $\rho = 0.2$	SMaRS $\rho = 0.4$	SMaRS $\rho = 0.6$
RDPG $N_i \in (1200, 1500)$ $\delta = 0.01$	Clean	56.3	53.7	61.1	55.0	57.0	58.2
	Random	50.1	52.2	56.4	46.9	50.5	53.0
	PGD	33.3	46.6	43.0	40.9	42.2	44.7
	Time	15.9	17.2	30.5	3.2	6.0	9.9
RDPG $N_i \in (1200, 1500)$ $\delta = 0.03$	Clean	87.4	86.2	93.2	72.8	78.0	85.6
	Random	65.3	79.3	77.6	73.5	77.2	80.9
	PGD	55.8	61.6	61.3	56.2	59.9	64.8
	Time	15.9	17.2	30.5	3.2	6.0	9.9
RDPG $N_i \in (1200, 1500)$ $\delta = 0.05$	Clean	100.0	100.0	100.0	95.2	100.0	100.0
	Random	85.0	96.7	100.0	91.4	100.0	100.0
	PGD	53.1	72.9	77.3	62.5	68.4	70.4
	Time	15.9	17.2	30.5	3.2	6.0	9.9
RDPG $N_i \in (1700, 2000)$ $\delta = 0.03$	Clean	82.7	81.8	90.5	69.7	78.5	88.0
	Random	76.8	80.5	81.2	64.9	73.2	76.9
	PGD	35.5	51.8	50.9	41.3	43.1	47.8
	Time	37.8	40.1	93.3	8.4	17.6	26.6
RDPG $N_i \in (1700, 2000)$ $\delta = 0.05$	Clean	100.0	100.0	100.0	100.0	100.0	100.0
	Random	95.2	100.0	100.0	88.5	100.0	100.0
	PGD	89.6	100.0	100.0	68.9	88.7	100.0
	Time	37.8	40.1	93.3	8.4	17.6	26.6
RDPG $N_i \in (2500, 3000)$ $\delta = 0.05$	Clean	100.0	100.0	100.0	100.0	100.0	100.0
	Random	87.8	100.0	100.0	71.8	100.0	100.0
	PGD	84.3	100.0	100.0	77.8	100.0	100.0
	Time	87.9	96.7	258.7	19.0	40.5	62.4

Table 3: Complete RDPG simulation results: accuracy in percent and training time in minutes. Av-GCN and RS-GCN use average and singular pooling, respectively; MaRS is the full-graph actor–critic model.

		Av-GCN	RS-GCN	MaRS	SMaRS $\rho = 0.2$	SMaRS $\rho = 0.4$	SMaRS $\rho = 0.6$
NW-ER $N_i \in (1200, 1500)$ $\beta = 0.1$	Clean	68.7	59.6	75.3	55.2	58.4	63.2
	Random	51.3	64.3	66.5	54.7	59.8	61.9
	PGD	40.1	46.2	55.6	44.0	47.7	49.8
	Time	16.1	19.9	31.5	3.4	6.9	9.4
NW-ER $N_i \in (1200, 1500)$ $\beta = 0.25$	Clean	100.0	100.0	100.0	100.0	100.0	100.0
	Random	79.1	94.8	100.0	75.4	82.6	92.0
	PGD	58.8	71.1	68.4	55.1	61.3	67.9
	Time	16.1	19.9	31.5	3.4	6.9	9.4
NW-ER $N_i \in (2500, 3000)$ $\beta = 0.25$	Clean	100.0	100.0	100.0	100.0	100.0	100.0
	Random	67.8	90.4	98.2	85.3	87.8	93.0
	PGD	54.3	66.6	64.9	58.5	62.2	65.3
	Time	90.4	102.7	279.3	20.4	42.3	64.0

Table 4: Complete NW-ER simulation results: accuracy in percent and training time in minutes. The methods and sampling ratios are the same as in Table 3.

can inform the choice of K (Zhou et al., 2025). The pooling scale τ multiplies the readout perturbation bound, but reducing it also changes the classifier input; a smaller absolute representation discrepancy alone does not guarantee better classification. The hidden-feature spectral gaps and classification margins jointly determine the useful stability regime.

Generalization scope. A bound for an average-pooling function class does not automatically apply to the singular-vector class. Likewise, a Rademacher-complexity rate controls an estimation or generalization-gap term; it does not imply that total classification error tends to zero without an empirical-error or approximation condition. Proposition B.10 and Corollary B.11 retain an explicit estimation term and identify the uniformity required to relate subgraph training to full-graph risk. Establishing sharp complexity bounds under verifiable gap and orientation conditions is a separate question.