IMBALANCED GRAPH CLASSIFICATION VIA GRAPH SIMILARITY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph classification is a fundamental task prevalent in fields such as molecular property prediction, bioinformatics, and social network analysis. However, real-world graph datasets suffer from severe graph size imbalance, which under limited label supervision induces size-driven similarity bias and semantic misalignment, resulting in significant performance gaps across size groups. In response, we propose a unified graph similarity-based learning framework **SimGraph** integrating balanced pairwise graph similarity (BPGS) learning and size-adaptive instance normalization (SAIN): BPGS formulates similarity estimation as pairwise binary classification with an asymmetric scaling scheme and a learnable decision boundary, while SAIN suppresses size-related variations in graph representations. We characterize the design-motivated properties of SimGraph, including its false-negative-free pairwise construction and bounded gradients, which provide principled support for stable optimization under size-imbalanced graph distributions. Extensive experiments on binary and multi-class benchmarks spanning molecular, biological, social, and heterogeneous graphs demonstrate that SimGraph achieves state-of-the-art performance on imbalanced graph classification with limited labels, with an average accuracy improvement of **4.2%** in size-imbalanced settings and up to **13%** on individual datasets.

1 INTRODUCTION

Graph Neural Networks (GNNs) (Kipf & Welling, 2017; Xu et al., 2019; Hamilton et al., 2017) are widely used for graph classification in molecular prediction (Hu et al., 2020), social networks (Cohen, 2016), and bioinformatics (Morris et al., 2020). However, real-world datasets often exhibit severe graph size imbalance, where graph sizes are highly skewed and correlated with labels (Liu et al., 2025; Qin et al., 2025b). This biases GNNs toward structurally rich graphs and degrades generalization (Zhao et al., 2022; Liu et al., 2022; Xu et al., 2024; Yehudai et al., 2021).

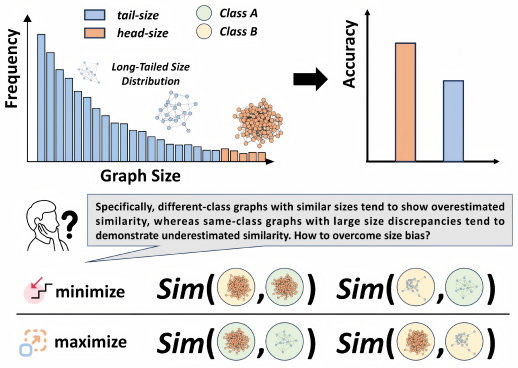


Figure 1: **Illustration of Size-Induced Similarity Bias.** Standard GNN representations entangle semantic information with graph size, overestimating similarity between different-class graphs of similar sizes (false positives) and underestimating similarity between same-class graphs of different sizes (false negatives). Our framework disentangles size from semantics, ensuring size-invariant intra-class compactness and inter-class separation.

In this work, we identify a critical yet underexplored issue in size imbalance: **graph size induces a systematic bias in similarity estimation.** More specifically, representations from GNNs often entangle semantic information with graph size, causing (i) same-class graphs with different sizes to demonstrate underestimated similarity, and (ii) different-class graphs with similar sizes to show overestimated similarity. We term this phenomenon **size-induced similarity bias**, as illustrated in Figure 1, which directly degrades representation quality in imbalanced graph classification. For example, consider two protein graphs from the D&D enzyme dataset: a small enzyme (∼50 nodes) and a large enzyme (∼500 nodes)

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.

both belonging to the same class. Despite sharing the same catalytic function, their GNN embeddings differ substantially due to size-dependent aggregation statistics, causing their cosine similarity to be underestimated (**false negative**). Conversely, a large enzyme and a large non-enzyme with similar sizes may exhibit high cosine similarity simply because their embedding norms and statistics are comparable (**false positive**). This systematic bias is exacerbated under severe size imbalance, where the training set is dominated by graphs of particular sizes.

Existing methods only partially address this issue. Early approaches focus on size-aware adaptation, intervening at the model or training-procedure level rather than the learning-objective level, such as curriculum learning (Wang et al., 2021). Pattern PT (Yehudai et al., 2021) further attributes size generalization gaps to distribution shifts of structural patterns and aligns them via pretraining. More recent works attempt to address this challenge through structural mining and knowledge transfer mechanisms. SOLT-GNN (Liu et al., 2022) identifies co-occurrence subgraph patterns from large graphs and transfers them to augment small graphs. TopoImb (Zhao et al., 2022) dynamically identifies latent topology groups and assigns importance weights to under-represented structural instances. ImbGNN (Xu et al., 2024) captures inter-graph correlations through kernel similarity to smooth feature discrepancies. SIMBA (Qin et al., 2025a) constructs a graph-to-graphs structure and performs energy-based propagation to balance the distribution. Despite their progress, these approaches share two key limitations: (1) they rely on heavy structural mining or complex training paradigms with high computational cost and limited scalability; (2) more fundamentally, they do not explicitly correct similarity estimation bias, as they assume that structural similarity reliably reflects semantic similarity, an assumption that often fails under severe size imbalance.

We therefore reformulate imbalanced graph classification as size-aware graph similarity learning, aiming to preserve semantic similarity across graph sizes. This poses two challenges.

Challenge 1: Size-induced Similarity Bias. Standard similarity learning methods, especially contrastive learning (Sun et al., 2020; You et al., 2020; Qiu et al., 2020; Zhu et al., 2021) based on InfoNCE or NT-Xent (Sohn, 2016; Wu et al., 2018; van den Oord et al., 2018) loss function, treat all non-augmented graphs as negatives. Under class and size imbalance, this induces false negatives, pushing semantically similar graphs—especially those from minority classes—apart (Chuang et al., 2020; Niu et al., 2024). More fundamentally, augmentation-based invariance in GraphCL (You et al., 2020) does not address the core requirement of imbalanced graph classification: aligning same-class graphs of different sizes while separating different-class graphs with similar sizes.

Challenge 2: Semantic Alignment under Size Imbalance. Recent methods implicitly assume that shared substructures dominate semantics (Liu et al., 2022; Zhao et al., 2022). However, random augmentations lack semantic grounding, while heuristic or learnable mining strategies incur high computational cost and require strong assumptions about the existence of rich common substructures (Zhao et al., 2022; Liu et al., 2022). Under severe size imbalance, such assumptions often break down, leading to strong structural dependency but weak semantic alignment across graph sizes.

To address these challenges, we propose a unified graph similarity-based framework **SimGraph**, integrating **Balanced Pairwise Graph Similarity learning (BPGS)** and **Size-Adaptive Instance Normalization (SAIN)**. BPGS tackles similarity estimation bias (Challenge 1) via a binary cross-entropy objective with explicit supervision and size-aware reweighting, naturally avoiding false negatives while emphasizing cross-size consistency. Complementarily, SAIN handles distributional shifts (Challenge 2) by treating graph size as a style variable, employing learnable affine transformations to architecturally disentangle size statistics from semantic features. The underlying primitives (pairwise BCE, instance normalization) are standard; our contribution is their imbalance-specific adaptation—asymmetric pair-level reweighting in BPGS and size-as-style decoupling in SAIN.

Our contributions are summarized as follows: (1) We identify and formalize the problem of size-induced similarity bias in imbalanced graph classification, where semantic similarity is affected by graph size. (2) We propose a size-aware pairwise similarity learning framework with two complementary mechanisms: (i) explicit size-aware reweighting and (ii) disentangling size statistics from semantic features. SimGraph is model-agnostic, interpretable, and orthogonal to existing GNN architectural improvements, while maintaining computational efficiency. (3) We characterize design-motivated properties of the balanced pairwise graph similarity learning loss, showing that it eliminates false negatives by construction and stabilizes training via asymmetrically bounded gradients. Additionally, we derive geometric characterizations of intra-class compactness and inter-class separation. (4) We achieve state-of-the-art performance on imbalanced graph classification

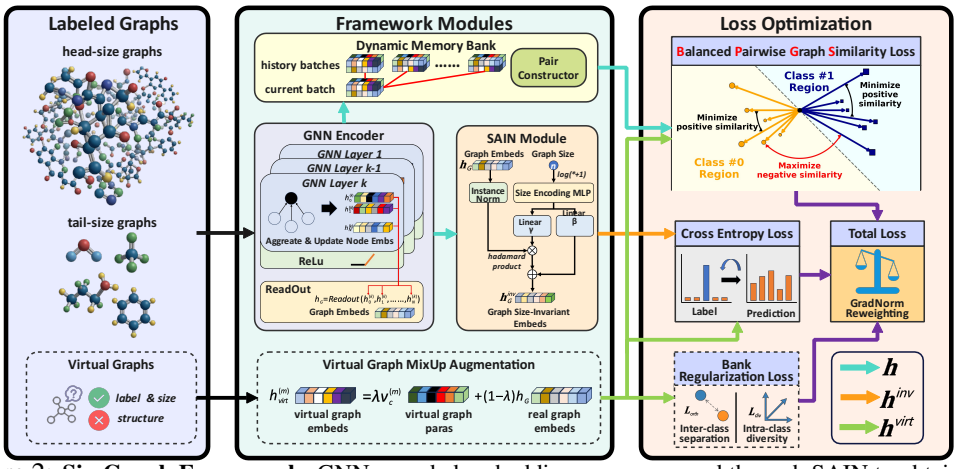


Figure 2: **SimGraph Framework**. GNN-encoded embeddings are processed through SAIN to obtain size-invariant representations. These are fed into both a classification head ($\mathcal{L}_{\text{CE}}$) and the BPGS module, which constructs pairs via a momentum-updated memory bank to eliminate false negatives and handle class imbalance through balance weight α. Optional virtual graph augmentation enhances minority class representation.

benchmarks with limited labels (IGL-Bench (Qin et al., 2025b)), with average accuracy improvements of **4.2%** in size-imbalanced settings over the best baselines, and further validate generality on multi-class, heterogeneous, and large-scale (OGB) graph benchmarks.

2 Preliminaries

Graph Classification Tasks. Given a labeled data set $\mathcal{G}_{\text{labeled}} = \{(G_i, y_i)\}_{i=1}^{N_l}$ where $y_i \in \{1, \ldots, C\}$ indicates the class label, and each graph $G_i = (\mathcal{V}_i, \mathcal{E}_i, \mathbf{X}_i)$ consists of a node set $\mathcal{V}_i$, an edge set $\mathcal{E}_i \subseteq \mathcal{V}_i \times \mathcal{V}_i$, and a feature matrix of nodes $\mathbf{X}_i \in \mathbb{R}^{|\mathcal{V}_i| \times d}$, the goal is to learn a function $f : \mathcal{G} \to \{1, \ldots, C\}$ that predicts labels for unseen graphs. We consider a *low-label supervised* setting where only a small subset of the graph collection is annotated, reflecting the high cost of graph labeling in practice. Following the IGL-Bench protocol (Qin et al., 2025b), a fixed 10%/10%/80% split is used for training, validation, and testing, and *no unlabeled graphs participate in training*: the unlabeled portion serves exclusively as validation and evaluation data. This setting is thus distinct from semi-supervised graph learning methods that optimize training objectives on unlabeled graphs, e.g., via pseudo-labeling or consistency regularization.

3 The SimGraph Framework

We propose **SimGraph**, a unified framework for imbalanced graph classification based on size-aware pairwise similarity learning (Figure 2). SimGraph consists of three complementary components. First, we propose Balanced Pairwise Graph Similarity Learning (BPGS) in Section 3.1. We design a BPGS loss function that eliminates false negatives and explicitly balances positive-negative pairs, applicable to both low-label supervised and self-supervised settings. Second, we develop Size-Adaptive Instance Normalization (SAIN) in Section 3.2 to architecturally decouple size-dependent statistics from semantic features. Third, we introduce training strategies in Section 3.3, including virtual graph augmentation and adaptive loss weighting to enhance generalization in imbalanced graph learning.

Design Philosophy and Advantages. SimGraph reformulates imbalanced graph classification as a problem of learning size-invariant geometry (Wu et al., 2024; Buffelli et al., 2022) in the embedding space, rather than modifying GNN architectures. This design yields several advantages: (1) **Model-agnostic**: BPGS operates on embeddings from any GNN encoder (*e.g.*, GCN, GIN, GraphSAGE), acting as a plug-and-play regularization component. (2) **Interpretable**: similarity function $\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j)$ directly reflects semantic relationships, with parameters (*e.g.*, learnable bias b, balance weight α, scaling factor r) having clear geometric interpretations as decision boundaries, pair-level importance, and margin controls. (3) **Orthogonal**: the framework complements existing GNN advances (*e.g.*, message passing and attention (Veličković et al., 2018; Ying et al., 2021)). (4) **Efficient**: operating on learned embeddings avoids the additional subgraph mining overhead (Liu et al., 2022; Zhao et al., 2022; Xu et al., 2024).

3.1 BALANCED PAIRWISE GRAPH SIMILARITY LEARNING (BPGS)

To tackle the size-induced similarity bias challenge and the false negative problem in traditional graph contrastive learning, we introduce BPGS in this section, which consists of a generalized graph similarity function design, a BPGS loss design, and memory bank-based pair construction.

3.1.1 GRAPH SIMILARITY FUNCTION

Graph size can bias embedding similarity: same-class graphs of different sizes may appear dissimilar, while different-class graphs of similar sizes may appear close. Inspired by similarity learning in face recognition (Sun et al., 2014), we compensate for this bias through a size-aware similarity function.

Given a graph G_i, its embedding is obtained by a GNN encoder f_Φ with parameters Φ and readout function, $\mathbf{h}_i = \text{ReadOut}(f_\Phi(G_i))$. For a pair of graphs (G_i, G_j) and their embeddings $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^d$, we design a graph similarity function as $\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) = \|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\| \cdot (\cos(\theta_{ij}) - b_\theta)$, where θ_{ij} is the angle between embeddings and b_θ is an angular bias hyperparameter selected via validation. Unlike standard cosine similarity that directly normalizes embeddings to a unified norm, the angular bias b_θ introduces a data-dependent correction term $\|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\| \cdot b_\theta$ that adapts to the dataset's size distribution, providing flexibility to mitigate size-induced similarity bias.

3.1.2 BPGS LOSS DESIGN

For a labeled graph pair (G_i, G_j) with binary relationship $y_{ij} = \mathbb{I}[y_i = y_j]$ (where y_i and y_j are class labels), we formulate a balanced binary cross-entropy loss:

$$\mathcal{L}^{\text{sup}+}_{\text{BPGS}}(i,j) = \log\left(1 + \exp\left(-\frac{1}{r}(\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) + \mathbf{b})\right)\right), \mathcal{L}^{\text{sup}-}_{\text{BPGS}}(i,j) = \log\left(1 + \exp\left(r(\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) + \mathbf{b})\right)\right), \tag{1}$$

$$\mathcal{L}^{\text{sup}}_{\text{BPGS}} = \mathbb{E}_{\{i,j\}\sim\mathcal{D}}\left\{\alpha \cdot y_{ij} \cdot \mathcal{L}^{\text{sup}+}_{\text{BPGS}}(i,j) + (1-\alpha)\cdot(1-y_{ij})\cdot\mathcal{L}^{\text{sup}-}_{\text{BPGS}}(i,j)\right\}, \tag{2}$$

where $\mathcal{L}^{\text{sup}+}_{\text{BPGS}}(i,j)$ and $\mathcal{L}^{\text{sup}-}_{\text{BPGS}}(i,j)$ denote the loss function term of positive or negative graph pair (G_i, G_j). $\mathcal{L}^{\text{sup}}_{\text{BPGS}}$ is the total BPGS loss on low-label supervised tasks. $\mathcal{D}$ is training dataset.

This formulation involves one learnable parameter and two hyperparameters:
(1) Learnable Bias $\mathbf{b}$: a trainable scalar optimized jointly with the encoder via gradient descent. It defines the decision boundary $\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) + \mathbf{b} = 0$, providing an adaptive threshold separating positive and negative pairs across all graph sizes. It is initialized to -10 and updated during training.
(2) Balance Weight $\alpha \in [0, 1]$: balances positive and negative pair contributions to address class imbalance. Since valid positive pairs (same-class labeled graphs) are rarer than negative pairs in imbalanced settings, setting $\alpha > 0.5$ assigns higher importance to the positive pairs. Distinct from class-wise (Cao et al., 2019; Tan et al., 2020) and prototype-based (Mao et al., 2023; Yi et al., 2023) methods, our prototype-free strategy uses α to uniformly balance positive versus negative pairs. This handles imbalance at the pair level, maintaining balanced gradients from both intra- and inter-class relationships.
(3) Asymmetric Scaling Factor $r > 0$: addresses the hard pair mining problem. $1/r$ in positive term of Equation 2 smooths gradients for *positive pairs*. r in negative term of Equation 2 sharpens gradients for *negative pairs*. This design alleviates overfitting in imbalanced graph learning by adaptively rectifying the gradients.

Comparison with Existing Loss Functions. BPGS differs from standard losses in several key aspects. (1) Unlike **InfoNCE**-style self-supervised contrastive objectives (van den Oord et al., 2018; You et al., 2020), which treat all non-augmented samples as negatives (causing false negatives, Proposition 3.1), BPGS uses explicit pairwise supervision y_{ij} to eliminate false negatives entirely. (2) Unlike standard **binary cross-entropy (BCE)**, BPGS introduces the asymmetric scaling factor r that provides asymmetric gradient bounds (Property 1) and the angular bias b_θ that enables magnitude-dependent decision boundaries. (3) Unlike **Focal Loss** (Lin et al., 2017), which modulates difficulty at the sample level, BPGS operates at the *pair level*, directly enforcing cross-size semantic alignment. (4) Compared with label-aware **SupCon** (Khosla et al., 2020), which likewise constructs positives from labels and admits bounded gradients (with respect to the similarity score, bounded by $1/\tau$), the distinctions of BPGS are structural rather than lying in false-negative elimination or

gradient boundedness: BPGS replaces the softmax-normalized objective, whose denominator induces competition among samples, with per-pair sigmoid BCE; it employs an asymmetric scaling factor r instead of a single temperature τ to separately control positive- and negative-pair gradient magnitudes; it learns an explicit decision boundary b; it constructs pairs beyond the current batch via a momentum memory bank with stratified sampling; and it operates on size-decoupled features produced by SAIN (see Table 5 in Appendix B.2 for a summary, and Appendix G.3 for an empirical comparison).

3.1.3 Memory Bank-based Pair Construction

In our low-label supervised setting, BPGS operates exclusively on *labeled* graphs: the memory bank $\mathcal{M}$ stores embeddings and labels of labeled training samples only, with all pairwise labels y_{ij} derived from ground-truth annotations. No pseudo-labels or unlabeled graphs are involved, which distinguishes SimGraph from semi-supervised methods that train on unlabeled data.

Constructing sufficient pairs is critical for minority classes. Simply using within-batch pairs limits coverage to $\mathcal{O}(B^2)$, with minority classes often absent from many batches. Therefore, we introduce a **dynamic memory bank** $\mathcal{M}$ that stores embeddings beyond the current mini-batch. In our low-label setting with small labeled sets (a few hundred graphs), $\mathcal{M}$ caches all training-batch embeddings within the current epoch. For larger datasets, $\mathcal{M}$ operates as a FIFO queue of configurable size Q.

Embeddings in $\mathcal{M}$ are produced by a **momentum encoder** f_{Φ_m} updated via exponential moving average $\Phi_m \leftarrow \eta\Phi_m + (1-\eta)\Phi$ with $\eta \in [0.9, 0.999]$, and treated as constants (gradients only back-propagate through the online encoder f_Φ). For each batch graph G_i, we form $B \times |\mathcal{M}|$ pairs with *stratified sampling* that prioritizes same-class graphs from $\mathcal{M}$ to enhance minority-class positive pair coverage.

Complexity. SimGraph's overhead beyond the base GNN is: $O(B|\mathcal{M}|)$ inner products for BPGS, $O(1)$ per-graph MLP for SAIN, and $O(|\Phi|)$ momentum update per batch. Unlike methods requiring $O(N^2)$ kernel computation (Qin et al., 2025a; Xu et al., 2024) or subgraph mining (Zhao et al., 2022; Liu et al., 2022), SimGraph needs **zero preprocessing** and scales linearly. See Appendix L for detailed comparisons.

3.1.4 Properties of the BPGS Loss

To provide design intuitions for the BPGS loss, we characterize several useful properties regarding its gradient behavior, false negative elimination, and embedding geometry. These properties motivate our specific design choices (e.g., asymmetric scaling, explicit pairwise supervision) rather than serving as generalization guarantees. Detailed derivations are provided in Appendix A.

Property 1 (**Gradient Boundedness**). The gradient of $\mathcal{L}_{\text{BPGS}}$ with respect to similarity score $s = \mathcal{S}(\mathbf{h}_i, \mathbf{h}_j)$ satisfies:

$$\left|\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}+}}{\partial s}\right| \leq \frac{1}{r}, \quad \left|\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}-}}{\partial s}\right| \leq r. \tag{3}$$

The gradients are bounded by $\max(r, 1/r)$, ensuring stable optimization. While standard CE loss $\mathcal{L}_{\text{CE}} = -\log\sigma(s)$ is also bounded (by 1), its gradient $\sigma(-s)$ remains non-negligible even for *correctly classified* pairs with moderate confidence, providing no mechanism to down-weight easy pairs. BPGS, by contrast, achieves *asymmetric* control: the scaling factor $r > 1$ compresses positive-pair gradients by $1/r$ while amplifying negative-pair gradients by r, allowing the model to focus on hard negatives while stabilizing positive learning. This is particularly crucial in low-label settings where limited labeled data can lead to noisy gradient signals early in training.

Proposition 3.1 (**False-Negative Penalty of Augmentation-Based Self-Supervised Contrastive Learning**). *InfoNCE implicitly treats all samples except augmentations as negatives. Given a dataset with C classes and class distribution $\{p_c\}_{c=1}^{C}$, the expected probability of false negative (FN) penalization satisfies:*

$$P_{\text{InfoNCE}}(\textit{FN}) = \sum_{c=1}^{C} p_c^2 - \frac{1}{N} > 0, \quad P_{\text{BPGS}}(\textit{FN}) = 0. \tag{4}$$

In imbalanced settings, $P_{\text{InfoNCE}}(\text{FN})$ is dominated by $\sum_{c=1}^{C} p_c^2$: under uniform sampling (widely used in graph classification (You et al., 2020; Sun et al., 2020)), majority-class graphs are dispro-

portionately sampled, inflating false negative rates. Since BPGS uses explicit pairwise supervision $y_{ij} = \mathbb{I}[y_i = y_j]$, it eliminates false negative penalization by construction. We note that this is a direct consequence of using label-based pair definitions rather than a novel theoretical insight, and it applies to augmentation-based self-supervised objectives; label-aware losses such as SupCon also construct positives from labels, so $P_{\text{BPGS}}(\text{FN}) = 0$ does not by itself distinguish BPGS from SupCon (see the comparison above and Appendix B); however, the quantitative comparison with InfoNCE's false negative rate $\sum_{c=1}^{C} p_c^2 - 1/N$ clarifies why supervised pairwise learning is particularly advantageous under class imbalance, where this rate is dominated by majority-class collision probability. Recent research (Zhu et al., 2020) has further validated that eliminating false negatives improves discriminative capability on imbalanced data.

Property 2 (**Intra-class Compactness and Inter-class Separation**). Assume the BPGS loss optimization reaches an approximate stationary point where gradient magnitudes are bounded by a small threshold $\delta > 0$. Then for any two same-class embeddings $\mathbf{h}_i, \mathbf{h}_j$ ($y_i = y_j$) and any two different-class embeddings $\mathbf{h}_k, \mathbf{h}_\ell$ ($y_k \neq y_\ell$), the learned similarity function satisfies:

$$\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) \geq -b + \epsilon_+, \quad \mathcal{S}(\mathbf{h}_k, \mathbf{h}_\ell) \leq -b - \epsilon_-, \tag{5}$$

where $\epsilon_+ = -r \ln(r\delta) > 0$ and $\epsilon_- = -\frac{1}{r} \ln(\frac{\delta}{r}) > 0$ are positive margins.

Property 2 shows that, at approximate stationarity, the BPGS loss induces a margin of $\epsilon_+ + \epsilon_-$ between intra-class and inter-class similarity scores. While this follows algebraically from the loss form, it provides a concrete geometric interpretation: the asymmetric scaling factor r directly controls the margin magnitudes ϵ_+ and ϵ_-, offering practitioners a principled knob for adjusting the compactness separation trade-off.

3.2 Size-Adaptive Instance Normalization (SAIN)

While BPGS provides explicit pairwise supervision to reduce similarity estimation bias, we must also architecturally enforce size invariance to fully disentangle size-dependent statistics from semantic features. Standard GNNs inevitably couple feature statistics with graph size during pooling, causing larger graphs to exhibit distinct statistical distributions regardless of their semantic class. To address this problem, we propose Size-Adaptive Instance Normalization (SAIN), which treats graph size as a style variable and aligns feature distributions via adaptive affine transformations.

Motivation of Normalization. Inspired by style transfer adain, where feature statistics (mean/variance) capture style, we observe that graph size acts as a nuisance "style" factor: just as feature statistics capture artistic style in style transfer, they also encode graph size information. Consequently, we propose to normalize these statistics to strip away size bias, followed by an adaptive re-injection. This performs a form of *style normalization*, ensuring downstream tasks operate on size-invariant features to complement BPGS.

Architecture. Given a graph embedding $\mathbf{h}_G \in \mathbb{R}^d$ from the GNN encoder and the graph size $n = |\mathcal{V}|$, SAIN transforms $\mathbf{h}_G$ into a size-invariant $\mathbf{h}_G^{\text{inv}}$ through three components:

(1) Instance Normalization. We normalize the embedding to zero mean and unit variance across feature dimensions: $\tilde{\mathbf{h}}_G = \frac{\mathbf{h}_G - \mu(\mathbf{h}_G)}{\sigma(\mathbf{h}_G) + \epsilon}$, where $\mu(\mathbf{h}_G) = \frac{1}{d}\sum_{i=1}^{d} \mathbf{h}_{G,i}$ and $\sigma(\mathbf{h}_G) = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(h_{G,i} - \mu(\mathbf{h}_G))^2}$ are per-sample statistics. d is the size of the embedding dimension. $\epsilon = 10^{-5}$ ensures numerical stability. This step removes the dependencies of first- and second-order moments on graph size (Ulyanov et al., 2017), thereby improving the model generalization capability across graphs of varying sizes.

(2) Size Encoding. We encode graph size into a continuous representation using a multi-layer perceptron (MLP): $\mathbf{s}_G = \text{MLP}(\log(n+1)) \in \mathbb{R}^d$, where MLP consists of 2-3 layers with ReLU activations. The logarithmic transformation compresses the large dynamic range of graph sizes. We adopt log-transformed node count $\log|\mathcal{V}_i|$ as the size input to the MLP, as it provides a stable and smooth representation across several orders of magnitude. We explored alternatives including raw node count (numerically unstable for large size ranges), binned one-hot encoding (comparable but coarser), and BatchNorm-based conditioning (less effective as it mixes statistics across batches). The log-transform + MLP combination consistently yielded the most stable results across datasets.

(3) Adaptive Affine Transformation. We predict size-conditioned scale and shift parameters:

$$\boldsymbol{\gamma}_G = \text{Linear}_\gamma(\mathbf{s}_G) \in \mathbb{R}^d, \quad \boldsymbol{\beta}_G = \text{Linear}_\beta(\mathbf{s}_G) \in \mathbb{R}^d, \quad \mathbf{h}_G^{\text{inv}} = \boldsymbol{\gamma}_G \odot \tilde{\mathbf{h}}_G + \boldsymbol{\beta}_G, \tag{6}$$

where $\odot$ denotes element-wise multiplication. Crucially, Linear_γ and Linear_β are initialized such that $\boldsymbol{\gamma}_G \approx \mathbf{1}$ and $\boldsymbol{\beta}_G \approx \mathbf{0}$ at initialization, ensuring SAIN acts as an identity mapping early in training.

Integration with BPGS. In supervised learning, SAIN is applied before both the classification layer and similarity computation:

$$\mathbf{h}_G^{\text{inv}} = \text{SAIN}(\mathbf{h}_G, n), \quad p(y|G) = \text{softmax}(\text{Linear}(\mathbf{h}_G^{\text{inv}})), \tag{7}$$

$$\mathcal{S}(\mathbf{h}_i^{\text{inv}}, \mathbf{h}_j^{\text{inv}}) = \|\mathbf{h}_i^{\text{inv}}\| \cdot \|\mathbf{h}_j^{\text{inv}}\| \cdot (\cos(\theta_{ij}) - b_\theta). \tag{8}$$

Complementary roles of SAIN and BPGS. SAIN and BPGS address size bias at two distinct levels, our discussion is at (Appendix I).

3.3 Training Strategies and Optimization

BPGS and SAIN target size-induced similarity bias. In practice, size imbalance typically co-exists with class imbalance, so we complement our core modules with two orthogonal, off-the-shelf training strategies: virtual graph augmentation (VGA) for minority-class data scarcity, and adaptive loss weighting for multi-objective stability. VGA is interchangeable with other class-balancing techniques and does not interact with the BPGS/SAIN design.

3.3.1 Virtual Graph Augmentation

To mitigate data scarcity under class imbalance, we introduce virtual graph augmentation to enhance representation diversity and balance class distributions.

Learnable Class Prototypes. For each class $c \in \{1, \ldots, C\}$, we maintain M learnable embedding parameters $\{\mathbf{v}_c^{(m)}\}_{m=1}^M \in \mathbb{R}^d$, initialized by sampling from the embedding distribution of real training graphs in class c after a warmup period (typically 10 epochs).

Mixup-based Graph Augmentation. To increase diversity beyond discrete prototypes, we generate virtual graphs through mixup mixup between virtual prototypes and real embeddings: $\mathbf{h}_{\text{virt}} = \lambda \mathbf{v}_c^{(m)} + (1-\lambda)\mathbf{h}_{G_i}, \quad y_{\text{virt}} = c$, where $\lambda \sim \text{Beta}(\alpha_{\text{mix}}, \alpha_{\text{mix}})$ (typically $\alpha_{\text{mix}} = 0.2$), G_i is a randomly sampled real graph from class c, $\text{Beta}(\cdot, \cdot)$ is the beta distribution, and the virtual graph inherits the hard label c. These augmented embeddings serve as additional training samples in both the cross-entropy loss $\mathcal{L}_{\text{CE}}$ and the similarity loss $\mathcal{L}_{\text{BPGS}}^{\text{sup}}$.

Size-aware Virtual Graphs. Crucially, virtual graphs are assigned sizes to populate underrepresented intervals in each class's empirical distribution, explicitly filling distributional gaps (e.g., missing medium ranges) to balance the size spectrum. When processed through SAIN (Equation 6), they benefit from the same size-invariant transformation as real graphs, ensuring consistency. This enables virtual graphs to implicitly augment the size distribution of minority classes.

Bank Regularization. Motivated by the losses used in label noise learning graphacl, we regularize the virtual embedding bank with two losses (divergence loss $\mathcal{L}_{\text{div}}$ and orthogonality loss $\mathcal{L}_{\text{orth}}$) to ensure high-quality class representations and prevent mode collapse, as below:

$$\mathcal{L}_{\text{div}} = -\frac{1}{C(C-1)} \sum_{c \neq c'} D_{\text{KL}}(\mathcal{N}(\mu_c, \Sigma_c) \| \mathcal{N}(\mu_{c'}, \Sigma_{c'})), \quad \mathcal{L}_{\text{orth}} = \frac{1}{C} \sum_{c=1}^{C} \left\| \mathbf{V}_c^\top \mathbf{V}_c - \mathbf{I} \right\|_F^2, \tag{9}$$

where μ_c, Σ_c are the mean and covariance of virtual embeddings $\{\mathbf{v}_c^{(m)}\}_{m=1}^M$ in class c, $\mathbf{V}_c \in \mathbb{R}^{M \times d}$ stacks the L_2-normalized virtual embeddings, and $\| \cdot \|_F$ is the Frobenius norm. D_{KL} is the Kullback-Leibler Divergence. $\mathbf{I}$ is the identity matrix. $\mathcal{L}_{\text{div}}$ encourages inter-class separation, while $\mathcal{L}_{\text{orth}}$ promotes intra-class diversity by encouraging orthogonal prototypes within each class. The regularization losses $\mathcal{L}_{div}$ and $\mathcal{L}_{orth}$ are essential for preventing VGA from introducing noise. As shown in Table 29, VGA consistently improves performance in data-scarce scenarios, confirming that the generated virtual embeddings are semantically meaningful rather than noisy.

3.3.2 Unified Training Objective

Low-Label Supervised Learning. Our complete objective combines classification, BPGS loss, and bank regularization terms of virtual graph augmentation: $\mathcal{L}_{\text{total}}^{\text{sup}} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{BPGS}}^{\text{sup}} + \lambda_2 \mathcal{L}_{\text{div}} + \lambda_3 \mathcal{L}_{\text{orth}}$, where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss computed on both real and virtual graphs, $\mathcal{L}_{\text{BPGS}}^{\text{sup}}$ is the BPGS

Table 1: Accuracy (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GIN / GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GIN (bb.) (Xu et al., 2019)	52.17±4.36	65.01±0.69	66.38±4.27	67.41±2.32
SOLT-GNN (Liu et al., 2022)	47.54±4.33	63.78±1.06	64.20±5.46	49.57±6.78
TopoImb (Zhao et al., 2022)	49.71±1.98	65.16±1.76	66.00±2.41	69.47±3.70
ImbGNN (Xu et al., 2024)	52.18±0.82	65.92±1.84	66.17±2.35	66.71±8.45
SIMBA (Qin et al., 2025a)	55.43±0.51	67.41±1.11	69.04±0.25	63.63±1.19
SimGraph	**56.52±2.42**	**74.99±2.29**	**70.25±0.67**	**73.09±2.30**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GIN (bb.) (Xu et al., 2019)	51.38±6.78	61.46±2.43	65.08±5.78	68.32±1.77
SOLT-GNN (Liu et al., 2022)	53.04±3.91	63.33±1.86	69.38±1.23	73.51±1.14
TopoImb (Zhao et al., 2022)	51.59±4.30	65.99±1.25	68.10±0.87	71.54±0.75
ImbGNN (Xu et al., 2024)	47.55±4.57	68.71±4.77	68.88±2.81	67.64±2.53
SIMBA (Qin et al., 2025a)	57.85±0.85	68.82±0.74	67.08±0.39	74.11±0.21
SimGraph	**59.42±0.78**	**76.43±1.17**	**69.75±1.63**	**80.21±0.33**
High level ρ_{size}	4.93	8.91	4.45	10.97
GIN (bb.) (Xu et al., 2019)	48.41±7.07	60.68±6.89	62.60±3.82	67.41±2.23
SOLT-GNN (Liu et al., 2022)	51.74±5.25	64.97±3.24	65.03±4.12	60.24±2.11
TopoImb (Zhao et al., 2022)	51.96±1.16	64.16±2.96	66.75±0.91	69.14±4.83
ImbGNN (Xu et al., 2024)	52.02±0.31	65.74±1.35	67.71±2.70	69.56±0.36
SIMBA (Qin et al., 2025a)	57.61±0.51	70.10±1.43	68.38±1.62	72.37±1.73
SimGraph	**60.51±1.57**	**77.32±0.87**	**69.00±3.19**	**77.94±0.58**

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GCN (bb.) (Kipf & Welling, 2017)	51.59±7.07	64.82±0.75	69.15±1.44	53.44±1.78
SOLT-GNN (Liu et al., 2022)	51.45±3.13	65.69±2.36	68.23±2.48	37.58±3.04
TopoImb (Zhao et al., 2022)	46.74±1.57	56.18±2.10	68.20±0.70	56.94±3.23
ImbGNN (Xu et al., 2024)	53.36±0.96	65.60±0.54	68.66±2.53	66.61±0.77
SIMBA (Qin et al., 2025a)	52.18±2.13	64.47±0.74	70.50±0.11	64.92±2.24
SimGraph	**53.50±1.78**	**73.85±0.98**	**71.04±0.72**	**70.48±3.76**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GCN (bb.) (Kipf & Welling, 2017)	50.00±5.87	63.99±2.16	67.95±2.82	67.96±0.89
SOLT-GNN (Liu et al., 2022)	56.23±0.84	62.78±1.33	70.25±1.15	61.95±5.55
TopoImb (Zhao et al., 2022)	54.13±4.62	64.73±7.09	68.75±0.76	69.12±0.52
ImbGNN (Xu et al., 2024)	50.42±1.28	67.35±5.02	68.83±2.76	67.35±2.55
SIMBA (Qin et al., 2025a)	57.97±1.02	67.87±1.01	67.46±1.08	68.58±0.78
SimGraph	**59.18±1.20**	**77.14±0.22**	**70.58±0.31**	**82.08±1.44**
High level ρ_{size}	4.93	8.91	4.45	10.97
GCN (bb.) (Kipf & Welling, 2017)	49.93±5.90	60.98±6.71	64.88±2.02	66.38±0.46
SOLT-GNN (Liu et al., 2022)	54.78±6.03	63.21±3.40	69.80±2.07	67.05±2.54
TopoImb (Zhao et al., 2022)	51.81±1.26	69.66±1.92	66.60±0.91	69.09±1.00
ImbGNN (Xu et al., 2024)	51.90±4.33	65.58±1.19	68.04±3.11	67.83±2.02
SIMBA (Qin et al., 2025a)	58.09±0.74	66.42±1.84	68.58±1.25	67.32±2.48
SimGraph	**59.90±0.68**	**76.11±2.30**	**70.67±0.51**	**79.63±0.97**

loss (Equation 2), and $\mathcal{L}_{\text{div}}, \mathcal{L}_{\text{orth}}$ (Equation 9) are the virtual bank regularization terms. λ_1, λ_2, and λ_3 are learnable parameters. Following CSSL cssl, similarity learning serves as a regularizer that leverages pairwise relationships to complement the standard supervised objective.

Adaptive Loss Weighting with GradNorm. Manually tuning the loss weights $\{\lambda_i\}$ is challenging and dataset-specific. We employ GradNorm gradnorm, an adaptive method that automatically balances loss terms by equalizing their gradient magnitudes.

4 Experiments

4.1 Experimental Setup

Datasets. To validate the effectiveness of SimGraph, we conduct comprehensive evaluations on the widely-used TUDataset series (Morris et al., 2020). The four main datasets span the size-imbalance spectrum: PTC-MR (avg. 14 nodes, molecules), D&D (avg. 284, proteins), IMDB-B (avg. 20, social), and RDT-B (avg. 430, size range 6–3,782, social), covering both attributed and degree-featured inputs. We additionally evaluate on COLLAB (3-class), heterogeneous graphs (DBLP, Freebase), and size-generalization baselines in Appendix G, and on the large-scale ogbg-molhiv and ogbg-molpcba (Hu et al., 2020) (Tables 3 and 4); dataset details are in Appendix D.1.

Baselines. For the **imbalanced low-label supervised graph classification** task, we compare against the baselines established in IGL-Bench (Qin et al., 2025b) and the latest methods C^{3}GNN (Ju et al., 2025), KDEX (Mao et al., 2025), and SIMBA (Qin et al., 2025a). Details are in Appendix D.2.

Protocol and Implementation. We align with the IGL-Bench protocol (Qin et al., 2025b): *fixed static* 10%/10%/80% splits (invariant across seeds and runs), 10 independent runs, and manipulated imbalance ratios (definitions in Appendix C); model selection uses the balanced validation set. Baseline numbers are either reported by IGL-Bench (GIN/GCN backbones, G^{2}GNN, TopoImb, DataDec, ImGKB, SOLT-GNN) or re-run by us under the identical protocol and tuning budget (RAHNet, GDeR, C^{3}GNN, KDEX, ImbGNN, SIMBA), following IGL-Bench tuning ranges and the original authors' recommended values. Metrics are Accuracy, Balanced Accuracy, Macro-F1, and AUC-ROC (full results in Appendix F). Implementation details are in Appendix D.

4.2 Results and Analysis

Size Imbalance (Table 1). SimGraph consistently outperforms all baselines across size imbalance levels: under high imbalance (ρ_{size}=4.93–10.97), it achieves 77.32% on D&D and 77.94% on RDT-B with GIN, surpassing SIMBA by 7.22% and 5.57%. Figure 4 (Appendix F) shows improved accuracy on both head- and tail-size graphs. **Class Imbalance** (Table 2). Gains widen under severe class imbalance: at ρ_{class}=9:1, SimGraph reaches 77.04% on D&D (+11.27% over DataDec), and remains

Table 3: Results on ogbg-molhiv with 10% labeled imbalanced split. OOM: out of memory.

Method	Acc	Macro-F1	Preproc. (s)	Epoch (s)
G^{2}GNN (Wang et al., 2022)	OOM	OOM	OOM	OOM
TopoImb (Zhao et al., 2022)	OOM	OOM	OOM	OOM
SOLT-GNN (Liu et al., 2022)	OOM	OOM	OOM	OOM
RAHNet (Mao et al., 2023)	68.58±2.54	66.24±3.62	0.53	46.34
DataDec (Zhang et al., 2023)	OOM	OOM	OOM	OOM
ImGKB (Tang & Liang, 2023)	67.50±2.70	65.87±3.81	12.83	3.77
GDeR (Zhang et al., 2024)	68.92±2.14	66.37±3.93	0.51	5.43
ImbGNN (Xu et al., 2024)	OOM	OOM	OOM	OOM
C^{3}GNN (Ju et al., 2025)	69.29±1.52	68.39±2.33	0.58	114.29
KDEX (Mao et al., 2025)	69.86±1.62	68.83±2.79	0.53	91.77
SIMBA (Qin et al., 2025a)	70.13±1.94	69.63±3.24	0.69	18.72
SimGraph(Q = 1,024)	**71.90±1.47**	**70.28±2.56**	0.03	5.56
SimGraph(Caching all)	**73.75±1.35**	**71.23±2.48**	0.03	8.43

Table 2: Accuracy (% ± stddev) on datasets with changing class imbalance levels over 10 runs, encoded by GIN / GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GIN (bb.) (Xu et al., 2019)	50.43±2.69	64.04±3.79	66.05±2.57	76.66±4.80
G^{2}GNN (Wang et al., 2022)	53.70±3.87	66.07±2.27	61.91±3.77	72.34±2.76
TopoImb (Zhao et al., 2022)	50.91±2.18	66.57±3.81	66.28±1.85	73.99±1.18
RAHNet (Mao et al., 2023)	52.54±3.59	67.87±0.24	70.38±1.34	78.17±1.17
DataDec (Zhang et al., 2023)	54.05±4.85	64.46±1.88	64.09±5.75	79.29±8.18
ImGKB (Tang & Liang, 2023)	53.48±3.50	65.45±2.88	50.16±0.34	50.24±0.29
GDeR (Zhang et al., 2024)	56.64±3.16	59.65±2.18	63.42±2.03	74.66±2.45
C^{3}GNN (Ju et al., 2025)	54.11±5.31	67.58±1.52	70.71±1.76	74.06±2.26
KDEX (Mao et al., 2025)	54.11±1.90	66.84±0.44	70.29±1.44	72.94±0.18
SimGraph	**57.24±2.71**	**75.52±0.80**	**70.83±1.31**	**80.88±1.90**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GIN (bb.) (Xu et al., 2019)	47.83±2.95	51.05±5.07	62.31±3.99	61.10±4.86
G^{2}GNN (Wang et al., 2022)	51.88±6.23	56.29±7.30	63.87±4.64	69.58±3.59
TopoImb (Zhao et al., 2022)	44.86±3.52	49.97±7.24	59.95±5.19	59.67±7.30
RAHNet (Mao et al., 2023)	50.72±5.92	68.57±2.52	64.88±0.31	77.96±0.76
DataDec (Zhang et al., 2023)	55.72±2.88	63.51±1.62	67.92±3.37	78.39±5.01
ImGKB (Tang & Liang, 2023)	50.11±5.95	65.85±3.70	47.74±0.29	48.57±2.14
GDeR (Zhang et al., 2024)	48.55±8.73	60.67±4.11	65.87±3.01	75.85±2.64
C^{3}GNN (Ju et al., 2025)	51.21±5.93	64.73±2.85	64.71±2.41	72.14±1.00
KDEX (Mao et al., 2025)	52.90±6.77	63.46±2.58	68.50±3.22	73.58±1.79
SimGraph	**56.64±1.71**	**75.88±0.31**	**72.25±0.63**	**80.27±1.08**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GIN (bb.) (Xu et al., 2019)	39.42±1.87	41.54±6.57	53.57±3.21	55.56±7.85
G^{2}GNN (Wang et al., 2022)	46.52±9.94	55.38±15.60	59.44±6.49	63.22±4.67
TopoImb (Zhao et al., 2022)	39.42±1.24	39.12±1.62	47.75±3.73	51.58±4.69
RAHNet (Mao et al., 2023)	44.32±8.88	54.57±7.52	59.80±2.77	74.34±0.36
DataDec (Zhang et al., 2023)	58.69±3.10	65.77±2.71	66.30±6.70	77.72±5.12
ImGKB (Tang & Liang, 2023)	44.24±5.65	59.99±7.57	47.08±3.72	51.25±5.10
GDeR (Zhang et al., 2024)	52.06±9.97	64.38±1.09	62.58±2.06	66.04±0.97
C^{3}GNN (Ju et al., 2025)	38.77±0.79	60.99±6.94	62.58±2.32	74.65±1.89
KDEX (Mao et al., 2025)	52.05±6.16	64.20±3.32	64.62±4.61	75.35±1.92
SimGraph	**61.84±3.08**	**77.04±0.74**	**68.21±2.35**	**81.67±1.82**

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GCN (bb.) (Kipf & Welling, 2017)	48.62±7.12	63.97±4.09	59.76±1.78	66.65±1.06
G^{2}GNN (Wang et al., 2022)	44.35±4.32	61.47±7.70	58.35±3.47	67.02±2.55
TopoImb (Zhao et al., 2022)	50.61±6.39	56.13±4.61	49.52±1.01	58.62±6.52
RAHNet (Mao et al., 2023)	58.21±0.62	72.91±3.34	69.75±0.27	79.27±0.65
DataDec (Zhang et al., 2023)	54.23±4.16	63.10±2.62	58.20±4.05	69.08±4.49
ImGKB (Tang & Liang, 2023)	52.83±5.45	64.93±4.44	50.15±0.39	50.29±0.19
GDeR (Zhang et al., 2024)	54.83±3.68	63.74±1.10	63.25±6.38	65.48±5.59
C^{3}GNN (Ju et al., 2025)	59.66±0.90	72.28±0.82	69.29±0.69	79.50±0.36
KDEX (Mao et al., 2025)	58.82±1.19	70.33±0.56	70.58±0.69	78.48±1.17
SimGraph	**61.71±0.45**	**74.74±2.63**	**71.00±0.98**	**79.67±1.28**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GCN (bb.) (Kipf & Welling, 2017)	43.84±7.03	58.32±1.51	49.06±2.17	61.85±3.89
G^{2}GNN (Wang et al., 2022)	47.86±9.03	65.65±5.41	53.11±3.97	66.02±2.08
TopoImb (Zhao et al., 2022)	45.90±6.41	44.63±4.82	49.75±4.81	54.35±3.13
RAHNet (Mao et al., 2023)	46.74±0.59	71.67±1.97	63.08±3.97	80.00±0.44
DataDec (Zhang et al., 2023)	55.86±2.49	64.66±2.04	57.06±5.97	66.45±5.38
ImGKB (Tang & Liang, 2023)	49.49±5.12	67.44±3.50	47.65±0.23	48.58±2.15
GDeR (Zhang et al., 2024)	49.40±8.22	62.86±0.62	61.29±3.40	71.23±0.41
C^{3}GNN (Ju et al., 2025)	45.17±0.74	72.20±1.79	64.25±0.47	79.41±0.34
KDEX (Mao et al., 2025)	55.31±6.58	72.03±1.01	68.58±1.08	75.63±0.94
SimGraph	**59.54±0.61**	**76.86±0.66**	**70.30±0.47**	**81.42±0.55**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GCN (bb.) (Kipf & Welling, 2017)	40.58±7.61	42.48±2.61	45.09±0.18	54.39±5.16
G^{2}GNN (Wang et al., 2022)	41.78±7.61	44.59±4.72	56.40±1.71	65.08±3.08
TopoImb (Zhao et al., 2022)	40.17±2.21	36.50±1.44	46.18±3.06	52.21±3.56
RAHNet (Mao et al., 2023)	38.04±0.00	48.75±8.48	56.37±2.21	74.31±0.76
DataDec (Zhang et al., 2023)	55.41±3.37	65.07±2.40	58.58±5.07	65.56±8.61
ImGKB (Tang & Liang, 2023)	45.98±8.71	58.21±10.60	46.06±2.74	51.23±5.09
GDeR (Zhang et al., 2024)	48.31±9.09	62.44±5.36	60.66±2.65	62.04±2.40
C^{3}GNN (Ju et al., 2025)	40.46±1.63	58.48±8.63	56.71±0.93	71.92±2.60
KDEX (Mao et al., 2025)	54.23±7.88	69.49±5.42	63.33±5.44	77.15±1.98
SimGraph	**60.14±3.87**	**76.26±1.63**	**64.25±3.80**	**79.13±1.22**

superior with the GCN encoder, demonstrating backbone-agnostic generalization.

Large-Scale Evaluation (Tables 3 and 4). On ogbg-molhiv (41,127 graphs, 10% labeled imbalanced split), SimGraph surpasses the strongest runnable baseline SIMBA by 3.62% accuracy with lower per-epoch runtime, and remains ahead of all baselines even with a bounded FIFO queue (Q=1,024); 6 of 12 baselines run out of memory (details in Appendix K). On ogbg-molpcba (437,929 graphs, 128 tasks), SimGraph achieves 25.78 test AP, +1.60 over the best baseline ImGKB, confirming that the gains transfer to large-scale multi-task molecular property prediction.

Additional Benchmark Comparisons. Appendix G reports: (i) size-generalization methods (Bevilacqua et al., SizeShiftReg), 3.0–3.8pp below SimGraph; (ii) multi-class COLLAB, with the best bAcc and Macro-F1 at all size-imbalance levels and the best Accuracy at two of three; (iii) heterogeneous graphs (DBLP, Freebase), +1.0 to +3.1 points over the strongest baseline without architectural changes; (iv) label-aware contrastive losses (SupCon, ProGCL), 4–8pp below CE+BPGS; and (v) diagnostic baselines confirming the gains are semantic rather than size-cue exploitation. **Key Patterns.** (1) SimGraph's margin *widens* as imbalance severity increases; (2) SAIN and VGA are *complementary*, benefiting size- and class-imbalanced settings; (3) BPGS contributes the largest gains, with linear scalability (Appendix L).

Table 4: Results on ogbg-molpcba (official split, AP %).

Method	Test AP
GIN (Xu et al., 2019)	22.66±0.28
RAHNet (Mao et al., 2023)	23.04±0.26
GDeR (Zhang et al., 2024)	23.47±0.34
SIMBA (Qin et al., 2025a)	23.62±0.36
ImGKB (Tang & Liang, 2023)	24.18±0.30
SimGraph	**25.78±0.32**

4.3 Ablation Studies

We ablate each component (**w/o BPGS**, **w/o SAIN**, **w/o VGA**) (Figure 3; full quantitative results, formal variant definitions, and significance tests in Appendix H): removing BPGS causes the largest drop (∼7% on RDT-B size imbalance); removing SAIN severely degrades size-imbalanced performance; removing VGA hurts class-imbalanced scenarios where data scarcity is the bottleneck.

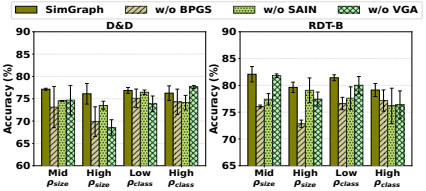


Figure 3: Results on D&D and RDT-B across 4 settings (same as Tables 1 and 2).

5 Conclusion

We proposed SimGraph for imbalanced graph classification, addressing size-induced similarity bias via BPGS and SAIN. It learns size-aware semantic geometry, remains model-agnostic and efficient, and achieves state-of-the-art results across imbalanced benchmarks; limitations and broader impacts are in Appendix M, and future work targets general size generalization beyond imbalance.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and code debugging assistance. We have reviewed all AI-assisted work and verified the correctness of the results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects and is based on publicly available datasets.

REPRODUCIBILITY STATEMENT

We have made extensive efforts to ensure the reproducibility of our work. The complete source code, including the implementation of our proposed method, training and evaluation scripts, and configuration files for all experiments, is provided in the supplementary material as an anonymized archive; we will release it publicly upon acceptance. Detailed descriptions of the model architecture, hyperparameters, computational resources, and training procedures are given in Section 4.1 and Appendix Section D. All datasets used in our experiments are publicly available. For our theoretical results, all assumptions are stated explicitly in Section 3.1, and complete proofs are provided in Appendix Section A.

REFERENCES

Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B. Aditya Prakash. Sub2vec: Feature learning for subgraphs. In *PAKDD*, pp. 170–182, 2018.

Yunsheng Bai, Hao Ding, Yang Qiao, Agustin Marinovic, Ken Gu, Ting Chen, Yizhou Sun, and Wei Wang. Unsupervised inductive graph-level representation learning via graph-graph proximity. In *IJCAI*, pp. 1988–1994, 2019.

Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations of graph neural networks. In *ICLR*, 2021.

Karsten M. Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *ICDM*, pp. 74–81, 2005.

Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56, 2005.

Davide Buffelli, Pietro Liò, and Fabio Vandin. Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks. *NeurIPS*, pp. 31871–31885, 2022.

Chen Cai and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508*, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pp. 793–802, 2018.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *NeurIPS*, 2020.

Sara Cohen. Data management for social networking. In *PODS*, pp. 165–177, 2016.

Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, pp. 786–797, 1991.

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

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pp. 855–864, 2016.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Jianhao Shen, Ziyue Qiao, and Ming Zhang. Cluster-guided contrastive class-imbalanced graph classification. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI*, pp. 11924–11932. AAAI Press, 2025. doi: 10.1609/AAAI.V39I11.33298. URL https://doi.org/10.1609/aaai.v39i11.33298.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, volume 33, pp. 18661–18673, 2020.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, pp. D1102–D1109, 2018.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *SIGKDD*, KDD '05, pp. 177–187. Association for Computing Machinery, 2005. ISBN 159593135X.

Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, pp. 13052–13065, 2022.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2999–3007. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.324. URL https://doi.org/10.1109/ICCV.2017.324.

Zemin Liu, Qiheng Mao, Chenghao Liu, Yuan Fang, and Jianling Sun. On size-oriented long-tailed graph classification of graph neural networks. In *WWW*, pp. 1506–1516, 2022.

Zemin Liu, Yuan Li, Nan Chen, Qian Wang, Bryan Hooi, and Bingsheng He. A survey of imbalanced learning on graphs: Problems, techniques, and future directions. *IEEE Trans. Knowl. Data Eng.*, pp. 3132–3152, 2025.

Zhengyang Mao, Wei Ju, Yifang Qin, Xiao Luo, and Ming Zhang. Rahnet: Retrieval augmented hybrid network for long-tailed graph classification. In *MM*, pp. 3817–3826, 2023.

Zhengyang Mao, Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Qingqing Long, Nan Yin, Xinwang Liu, and Ming Zhang. Learning knowledge-diverse experts for long-tailed graph classification. *ACM Trans. Knowl. Discov. Data*, 19(2):32:1–32:24, 2025. doi: 10.1145/3705323. URL https://doi.org/10.1145/3705323.

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.

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.

Chaoxi Niu, Guansong Pang, and Ling Chen. Affinity uncertainty-based hard negative mining in graph contrastive learning. *IEEE Trans. Neural Networks Learn. Syst.*, pp. 11681–11691, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: an imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Jiawen Qin, Pengfeng Huang, Qingyun Sun, Cheng Ji, Xingcheng Fu, and Jianxin Li. Graph size-imbalanced learning with energy-guided structural smoothing. In *WSDM*, pp. 457–465, 2025a.

Jiawen Qin, Haonan Yuan, Qingyun Sun, Lyujin Xu, Jiaqi Yuan, Pengfeng Huang, Zhaonan Wang, Xingcheng Fu, Hao Peng, Jianxin Li, and Philip S. Yu. Igl-bench: Establishing the comprehensive benchmark for imbalanced graph learning. In *ICLR*, 2025b.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: graph contrastive coding for graph neural network pre-training. In *SIGKDD*, pp. 1150–1160, 2020.

Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, pp. 488–495, 2009.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, pp. 2539–2561, 2011.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, pp. 1849–1857, 2016.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2020.

Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *Advances in neural information processing systems*, 2014.

Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*, 2021.

Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, pp. 11659–11668, 2020.

Hui Tang and Xun Liang. Where to find fascinating inter-graph supervision: Imbalanced graph classification with kernel information bottleneck. In *MM*, pp. 3240–3249, 2023.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *CoRR*, 2017.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.

Luzhi Wang, Yizhen Zheng, Di Jin, Fuyi Li, Yongliang Qiao, and Shirui Pan. Contrastive graph similarity networks. *ACM Transactions on the Web*, 18(2):1–20, 2024.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Curgraph: Curriculum learning for graph classification. In *WWW*, pp. 1238–1248, 2021.

Yu Wang, Yuying Zhao, Neil Shah, and Tyler Derr. Imbalanced graph classification via graph-of-graph neural networks. In *CIKM*, pp. 2067–2076, 2022.

Junran Wu, Xueyuan Chen, Bowen Shi, Shangzhe Li, and Ke Xu. SEGA: structural entropy guided anchor view for graph contrastive learning. In *ICML*, pp. 37293–37312, 2023.

Shirley Wu, Kaidi Cao, Bruno Ribeiro, James Zou, and Jure Leskovec. Graphmetro: Mitigating complex graph distribution shifts via mixture of aligned experts. *NeurIPS*, pp. 9358–9387, 2024.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *WWW*, pp. 1070–1079, 2022a.

Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. ProGCL: Rethinking hard negative mining in graph contrastive learning. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24332–24346, 2022b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Wei Xu, Pengkun Wang, Zhe Zhao, Binwu Wang, Xu Wang, and Yang Wang. When imbalance meets imbalance: Structure-driven learning for imbalanced graph classification. In *WWW*, pp. 905–913, 2024.

Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *SIGKDD*, pp. 1365–1374, 2015.

Gilad Yehudai, Ethan Fetaya, Eli A. Meirom, Gal Chechik, and Haggai Maron. From local structures to size generalization in graph neural networks. In *ICML*, pp. 11975–11986, 2021.

Si-Yu Yi, Zhengyang Mao, Wei Ju, Yong-Dao Zhou, Luchen Liu, Xiao Luo, and Ming Zhang. Towards long-tailed recognition for graph classification via collaborative experts. *IEEE Transactions on Big Data*, pp. 1683–1696, 2023.

Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. Autogcl: Automated graph contrastive learning via learnable view generators. In *AAAI*, pp. 8892–8900, 2022.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *NeurIPS*, pp. 28877–28888, 2021.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, pp. 12121–12132, 2021.

Chunhui Zhang, Chao Huang, Yijun Tian, Qianlong Wen, Zhongyu Ouyang, Youhuan Li, Yanfang Ye, and Chuxu Zhang. When sparsity meets contrastive models: Less graph data can bring better class-balanced representations. In *ICML*, pp. 41133–41150, 2023.

Guibin Zhang, Haonan Dong, Yuchen Zhang, Zhixun Li, Dingshuo Chen, Kai Wang, Tianlong Chen, Yuxuan Liang, Dawei Cheng, and Kun Wang. Gder: Safeguarding efficiency, balancing, and robustness via prototypical graph pruning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/59f278de1619bdb6b53fd04e8e0976e0-Abstract-Conference.html.

Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. Topoimb: Toward topology-level imbalance in learning from graphs. In *LoG*, pp. 37, 2022.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *CoRR*, 2020.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, pp. 2069–2080, 2021. doi: 10.1145/3442381.3449802.

A THEORETICAL ANALYSIS OF BPGS

In this section, we provide detailed proofs for the theoretical properties of the Balanced Pairwise Graph Similarity (BPGS) loss presented in the main text. We analyze its gradient stability, its immunity to false negatives, and the geometric structure it induces in the embedding space.

A.1 THEORETICAL ANALYSIS OF PROPERTY 1

Recall the definitions of the positive and negative loss components for a pair of embeddings $(\mathbf{h}_i, \mathbf{h}_j)$ with similarity $s = \mathcal{S}(\mathbf{h}_i, \mathbf{h}_j)$:

$$\mathcal{L}_{\text{BPGS}}^{\text{sup}+} = \log\left(1 + \exp\left(-\frac{1}{r}(s+b)\right)\right), \tag{10}$$

$$\mathcal{L}_{\text{BPGS}}^{\text{sup}-} = \log\left(1 + \exp\left(r(s+b)\right)\right). \tag{11}$$

Property 3 (**Gradient Boundedness**). The gradient of $\mathcal{L}_{\text{BPGS}}$ with respect to the similarity score s is strictly bounded:

$$\left|\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}+}}{\partial s}\right| < \frac{1}{r}, \quad \left|\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}-}}{\partial s}\right| < r. \tag{12}$$

The asymmetric scaling factor $r > 1$ provides two distinct advantages over standard losses:

1. **Asymmetric gradient control:** Positive-pair gradients are compressed by $1/r$, preventing over-aggressive attraction of already-similar embeddings, while negative-pair gradients are amplified by r, ensuring strong repulsion of different-class pairs. This asymmetry is absent in standard CE, where both directions share the same gradient scale.
2. **Easy-pair suppression:** For a correctly classified positive pair with high similarity $s \gg -b$, the BPGS gradient $\frac{1}{r}\sigma(-\frac{s+b}{r})$ decays exponentially due to both the $1/r$ prefactor and the sigmoid saturation at rate $1/r$ (slower than CE's rate of 1). This means BPGS naturally focuses on informative hard pairs. Standard CE loss $-\log\sigma(s)$ has gradient $\sigma(-s)$, which although bounded by 1, decays at a fixed rate regardless of pair difficulty and provides no mechanism for asymmetric treatment.

Proof. Let $x = s + b$. For the positive pair loss $\mathcal{L}_{\text{BPGS}}^{\text{sup}+}$, let $f(s) = \log(1 + \exp(-\frac{1}{r}(s+b)))$. Applying the chain rule:

$$\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}+}}{\partial s} = \frac{1}{1+\exp(-\frac{x}{r})} \cdot \frac{\partial}{\partial s}\left(1 + \exp\left(-\frac{x}{r}\right)\right) \tag{13}$$

$$= \frac{1}{1+\exp(-\frac{x}{r})} \cdot \exp\left(-\frac{x}{r}\right) \cdot \left(-\frac{1}{r}\right) \tag{14}$$

$$= -\frac{1}{r} \cdot \frac{\exp(-\frac{x}{r})}{1+\exp(-\frac{x}{r})} \tag{15}$$

$$= -\frac{1}{r} \cdot \sigma\left(-\frac{s+b}{r}\right), \tag{16}$$

where $\sigma(\cdot)$ is the sigmoid function. Since $0 < \sigma(z) < 1$ for any $z \in \mathbb{R}$, we have:

$$\left|\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}+}}{\partial s}\right| < \frac{1}{r}. \tag{17}$$

Similarly, for the negative pair loss $\mathcal{L}_{\text{BPGS}}^{\text{sup}-}$, let $g(s) = \log(1 + \exp(r(s+b)))$:

$$\frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}-}}{\partial s} = \frac{1}{1+\exp(rx)} \cdot \exp(rx) \cdot r \tag{18}$$

$$= r \cdot \sigma(r(s+b)). \tag{19}$$

Again, due to the range of the sigmoid function, the magnitude is bounded:

$$\left| \frac{\partial \mathcal{L}_{\text{BPGS}}^{\text{sup}-}}{\partial s} \right| < r. \tag{20}$$

Thus, the gradients are strictly bounded by $\max(1/r, r)$, with the key property that this bound is *controllable* via r, unlike CE where the bound is fixed at 1. □

A.2 Theoretical Analysis of Proposition 3.1

Proposition A.1 (False-Negative Penalty of Augmentation-Based Self-Supervised Contrastive Learning)**.** *Given a dataset with C classes and class distribution $\{p_c\}_{c=1}^{C}$. The expected probability of false negative penalization in standard InfoNCE (which treats all non-paired samples as negatives) is:*

$$P_{\textit{InfoNCE}}(\textit{FN}) = \sum_{c=1}^{C} p_c^2 - \frac{1}{N}. \tag{21}$$

In contrast, BPGS utilizes explicit pairwise supervision y_{ij}, resulting in $P_{\textit{BPGS}}(\textit{FN}) = 0$.

Proof. Consider an anchor sample i with class label $y_i = c$. In the standard InfoNCE framework (or other self-supervised contrastive losses applied naively to labeled data), any sample k in the batch (where $k \neq i$) is treated as a negative sample. A *False Negative* event occurs if the "negative" sample k actually shares the same class as i, i.e., $y_k = y_i = c$.

Assuming samples are drawn i.i.d. from the data distribution, the probability that a randomly selected negative sample k belongs to class c is p_c. The probability of a false negative for a fixed anchor of class c is therefore $P(y_k = c | y_i = c) \approx p_c$ (ignoring the negligible adjustment of sampling without replacement from a large dataset N, which is represented by the $-\frac{1}{N}$ term).

The expected probability of false negatives across all classes is the sum of these probabilities weighted by the class prior:

$$P_{\text{InfoNCE}}(\text{FN}) = \sum_{c=1}^{C} P(y_i = c) \cdot P(y_k = c) = \sum_{c=1}^{C} p_c \cdot p_c = \sum_{c=1}^{C} p_c^2. \tag{22}$$

In imbalanced settings, this probability is dominated by the majority class. For example, if one class occupies 90% of the data ($p_{\text{maj}} = 0.9$), then $P(\text{FN}) \geq 0.81$, meaning 81% of the negative pairs in InfoNCE are actually false negatives, severely damaging the representation structure.

BPGS avoids this structural flaw by using the label indicator $y_{ij} = \mathbb{I}(y_i = y_j)$. The term $\mathcal{L}^{\text{sup}-}$ is only active when $y_{ij} = 0$, ensuring that same-class pairs are never penalized as negatives. Thus, $P_{\text{BPGS}}(\text{FN}) = 0$. □

Remark A.2. The result $P_{\text{BPGS}}(\text{FN}) = 0$ follows directly from BPGS's use of explicit labels, and is stated as a proposition rather than a theorem to reflect this. However, the *quantitative comparison* with InfoNCE is non-trivial: it reveals that the false negative rate under InfoNCE scales as $\sum_c p_c^2$, which approaches 1 under severe class imbalance ($p_{\max} \to 1$). This makes the design choice of label-supervised pair construction particularly impactful in the imbalanced regime that is the focus of this work.

A.3 Theoretical Analysis of Property 2

Property 4 (Intra-class Compactness and Inter-class Separation). Assume the BPGS loss is minimized such that the gradient magnitude for any pair is smaller than a small threshold $\delta > 0$ (approximate stationary point). Then for any same-class pair (i, j) and different-class pair (k, ℓ), the similarity scores satisfy:

$$\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) \geq -b + \epsilon_+, \quad \mathcal{S}(\mathbf{h}_k, \mathbf{h}_\ell) \leq -b - \epsilon_-, \tag{23}$$

where $\epsilon_+ = -r \ln(r\delta)$ and $\epsilon_- = -\frac{1}{r} \ln(\frac{\delta}{r})$ are positive margins (assuming δ is sufficiently small).

Proof. The optimization process minimizes the gradients of the loss function. We analyze the conditions under which gradients become negligible (i.e., the loss saturates), indicating that the model has learned the desired structure. Let δ be a small positive constant representing the gradient convergence threshold.

1. Intra-class Compactness (Positive Pairs): For a positive pair (i, j) with similarity $s = \mathcal{S}(\mathbf{h}_i, \mathbf{h}_j)$, we require the gradient magnitude to be small:

$$\left|\frac{\partial \mathcal{L}^+}{\partial s}\right| = \frac{1}{r}\sigma\left(-\frac{s+b}{r}\right) \leq \delta. \tag{24}$$

Rearranging the inequality:

$$\sigma\left(-\frac{s+b}{r}\right) \leq r\delta \tag{25}$$

$$\frac{1}{1+\exp(\frac{s+b}{r})} \leq r\delta. \tag{26}$$

Assuming $r\delta$ is small (since $\delta \to 0$), the denominator must be large, implying $\exp(\frac{s+b}{r})$ is large. We can approximate using the tail property of the sigmoid function:

$$\exp\left(-\frac{s+b}{r}\right) \approx \sigma\left(-\frac{s+b}{r}\right) \leq r\delta \quad (\text{using } \sigma(z) \approx e^z \text{ for } z \ll 0) \tag{27}$$

$$-\frac{s+b}{r} \leq \ln(r\delta). \tag{28}$$

Multiplying by $-r$ (and flipping the inequality):

$$s + b \geq -r\ln(r\delta). \tag{29}$$

Let $\epsilon_+ = -r\ln(r\delta)$. Since δ is very small, $\ln(r\delta)$ is a large negative number, making ϵ_+ a positive margin. Thus:

$$\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) \geq -b + \epsilon_+. \tag{30}$$

This demonstrates that same-class similarities are pushed above $-b$ by a margin ϵ_+.

2. Inter-class Separation (Negative Pairs): For a negative pair (k, ℓ) with similarity $s = \mathcal{S}(\mathbf{h}_k, \mathbf{h}_\ell)$, we similarly require:

$$\left|\frac{\partial \mathcal{L}^-}{\partial s}\right| = r\sigma(r(s+b)) \leq \delta. \tag{31}$$

Rearranging:

$$\sigma(r(s+b)) \leq \frac{\delta}{r} \tag{32}$$

$$\frac{1}{1+\exp(-r(s+b))} \leq \frac{\delta}{r}. \tag{33}$$

Similarly approximating for the tail of the sigmoid:

$$\exp(r(s+b)) \leq \frac{\delta}{r} \tag{34}$$

$$r(s+b) \leq \ln(\delta/r). \tag{35}$$

Dividing by r:

$$s + b \leq \frac{1}{r}\ln(\delta/r). \tag{36}$$

Let $\epsilon_- = -\frac{1}{r}\ln(\delta/r) = \frac{1}{r}\ln(r/\delta)$. Since δ/r is small, ϵ_- is positive. Thus:

$$\mathcal{S}(\mathbf{h}_k, \mathbf{h}_\ell) \leq -b - \epsilon_-. \tag{37}$$

This guarantees that different-class embeddings are pushed below $-b$ by a margin ϵ_-.

The learned embedding space is structured such that all positive pairs have similarity strictly greater than $-b + \epsilon_+$ and all negative pairs strictly less than $-b - \epsilon_-$. The total separation gap is $\epsilon_+ + \epsilon_- = r\ln\frac{1}{r\delta} + \frac{1}{r}\ln\frac{r}{\delta}$, which grows as $\delta \to 0$, confirming that tighter optimization yields larger margins. The parameter b acts as the geometric center of the decision boundary, effectively separating the classes in the embedding space. □

B Related Works

Recent works have attempted to address the graph size imbalance challenge through structural mining and knowledge transfer mechanisms. For instance, SOLT-GNN Liu et al. (2022) identifies co-occurrence subgraph patterns from large graphs and transfers them to augment small graphs via a pattern-based similarity metric, improving their structural expressiveness. TopoImb Zhao et al. (2022) dynamically identifies latent topology groups and assigns importance weights to under-represented structural instances during training to rebalance the distribution. To further bridge structural gaps, ImbGNN Xu et al. (2024) captures inter-graph correlations through kernel similarity to calibrate feature deviations. Similarly, SIMBA Qin et al. (2025a) constructs a Graph-to-Graphs (G2G) structure and performs energy-based propagation, utilizing global message passing to smooth feature discrepancies across graphs and balance the overall distribution. These methods, however, suffer from common bottlenecks regarding scalability and similarity estimation, as described in Section 1.

While the aforementioned methods emphasize structural and size disparities, a broader line of research focuses primarily on mitigating class imbalance at the graph level. To augment minority classes, G^2GNN Wang et al. (2022) propagates supervisory signals via a Graph-of-Graphs (GoG) paradigm with topological augmentations, whereas ImGKB Tang & Liang (2023) integrates graph kernels with prior knowledge to guide feature enhancement. From a data-centric perspective, DataDec Zhang et al. (2023) employs dynamic distillation to select highly informative subsets, and GDeR Zhang et al. (2024) introduces prototype-guided soft pruning to extract unbiased, representative training samples. Moreover, to align skewed distributions, KDEX Mao et al. (2025) utilizes kernel mapping to narrow the representation gap between majority and minority classes. Finally, C^3GNN Ju et al. (2025) prevents representation collapse by adaptively clustering majority classes and applying Mixup interpolation within a hierarchical contrastive learning framework.

B.1 Comparison with Size-Generalization Methods

A related but distinct line of research studies *size generalization*, where the training set contains (almost) exclusively small graphs and the model must extrapolate to unseen large graphs at test time, i.e., an out-of-distribution covariate-shift problem. Bevilacqua et al. Bevilacqua et al. (2021) build size-invariant graph representations via subgraph density weighting, targeting the stability of a *single graph's* representation across sizes; SizeShiftReg Buffelli et al. (2022) regularizes representations to be invariant to coarsening-based size shifts via central moment discrepancy; Pattern-PT Yehudai et al. (2021) aligns structural pattern distributions via pretraining. By contrast, SimGraph addresses *in-distribution size imbalance*, where both large and small graphs appear in training but the size distribution is severely skewed, and the core failure mode is a bias in *pairwise similarity estimation* rather than instability of single-graph representations. The two problem settings are complementary: even a perfectly size-invariant representation can still yield biased similarity scores if the comparison function conflates norm magnitude with semantic distance, which is precisely what the angular bias b_θ and the learnable boundary b in BPGS address. Empirically, when adapted to the IGL-Bench protocol, both methods remain below SimGraph under high size imbalance (see the experimental comparison of size-generalization methods).

B.2 Comparison with Supervised and False-Negative-Aware Contrastive Learning

Our critique of false negatives applies specifically to *augmentation-based self-supervised* InfoNCE/GraphCL objectives, which treat all non-augmented samples as negatives; it does not apply to label-aware methods. SupCon Khosla et al. (2020) constructs the positive set directly from labels, so same-class pairs are never treated as negatives, and its gradients with respect to the similarity score are likewise bounded (by $1/\tau$). ProGCL Xia et al. (2022b) mitigates false negatives in graph contrastive learning by estimating, via a Beta mixture model, the posterior probability of a negative being a true negative and down-weighting likely false negatives. The substantive distinctions between BPGS and these methods are structural rather than in false-negative elimination or gradient boundedness: (i) BPGS uses per-pair sigmoid binary cross-entropy without a softmax denominator, avoiding competition among samples in the batch; (ii) it employs an asymmetric scaling factor r (instead of a single temperature) that bounds positive- and negative-pair gradients by $1/r$ and r respectively;

(iii) it learns an explicit decision boundary b; (iv) it constructs pairs beyond the current batch via a momentum memory bank with stratified sampling to improve minority-class positive-pair coverage; and (v) it operates on size-decoupled features produced by SAIN. Table 5 summarizes these structural differences; an empirical comparison is provided in Appendix G.3.

Table 5: Structural comparison of BPGS with label-aware contrastive losses.

Property	**SupCon**	**ProGCL**	**BPGS (ours)**
Objective form	softmax InfoNCE	weighted softmax	per-pair sigmoid BCE
Denominator competition	yes	yes	no
Gradient scaling	single τ	single τ	asymmetric r
Decision boundary	none	none	learnable b_θ
Pair construction	in-batch	in-batch	memory bank + stratified
Feature conditioning	raw	raw	SAIN size-decoupled

B.3 Comparison with Graph Similarity Learning

CGSN Wang et al. (2024) learns *inter-graph* similarity via a multi-scale contrastive objective for graph matching and retrieval, using InfoNCE-style contrast and thus retaining the false-negative penalty quantified in Proposition 3.1. The key conceptual difference is that CGSN treats graph similarity as the *primary task* (matching/retrieval), whereas BPGS treats pairwise similarity as a *regularization signal* for classification under size imbalance: CGSN's multi-scale contrastive objectives are designed for structural matching, while BPGS's balanced pairwise BCE, asymmetric scaling, learnable boundary, and size-adaptive angular bias are designed for *class-semantic alignment across size-varying graphs*. An empirical comparison of BPGS with label-aware contrastive objectives (SupCon, ProGCL) is provided in Appendix G.3.

C Definition of Imbalance Ratios

Graph Size Imbalance Ratio. A graph classification dataset exhibits *graph size imbalance* when the distribution of graph sizes (measured by node counts $\{|\mathcal{V}_i|\}$) is highly skewed, with a small fraction of large graphs dominating the dataset. In low-label settings, this issue is exacerbated by a distribution shift between labeled and unlabeled data, where the size distribution of $\mathcal{G}_{\text{labeled}}$ significantly differs from that of $\mathcal{G}_{\text{unlabeled}}$. To quantify size imbalance, we define *graph size imbalance ratio* as:

$$\rho_{\text{size}} = \frac{\frac{1}{|\mathcal{H}_{\text{g}}|}\sum_{\mathcal{G}_i \in \mathcal{H}_{\text{g}}} |\mathcal{V}_i|}{\frac{1}{|\mathcal{T}_{\text{g}}|}\sum_{\mathcal{G}_j \in \mathcal{T}_{\text{g}}} |\mathcal{V}_j|}, \tag{38}$$

where $\mathcal{H}_{\text{g}}$ and $\mathcal{T}_{\text{g}}$ denote the subsets of labeled graphs in the top 20% (head) and bottom 80% (tail) of the size distribution, respectively. A larger ρ_{size} indicates a more severe size imbalance.

Class Imbalance Ratio. Given the labeled training set $\mathbb{G}_{\text{labeled}} = \{g_i\}_{i=1}^{N_L}$ where each graph g_i belongs to class $y_i \in \mathcal{C}$, we define the class imbalance ratio as:

$$\rho_{\text{class}} = \frac{\max_{c\in\mathcal{C}} |\{g_i \in \mathbb{G}_L | y_i = c\}|}{\min_{c\in\mathcal{C}} |\{g_i \in \mathbb{G}_L | y_i = c\}|} \tag{39}$$

This formulation captures the disparity between the most and least frequent classes in the training data. In practice, such imbalance leads to biased classifiers that favor majority classes, typically manifesting as low recall rates for minority classes where predictions disproportionately favor the dominant categories.

D Details of the Experimental Settings

D.1 Description of Datasets

PTC-MR Bai et al. (2019) is a molecular graph dataset containing chemical compounds, where the task is to predict the carcinogenicity in male rats.

Table 6: Statistics of benchmark datasets for graph classification.

Dataset	#Graphs	Avg. #Nodes	Avg. #Edges	#Classes	#Node Labels	#Edge Labels	#Node Attrs	#Edge Attrs
PTC-MR	344	14.3	14.7	2	+	+	-	-
PROTEINS	1,113	39.1	72.8	2	+	-	+(1)	-
MUTAG	188	17.9	19.8	2	+	+	-	-
D&D	1,178	284.3	715.7	2	+	-	-	-
NCI1	4,110	29.9	32.3	2	+	-	-	-
IMDB-BINARY	1,000	19.8	96.5	2	-	-	-	-
REDDIT-BINARY	2,000	429.6	497.8	2	-	-	-	-
COLLAB	5,000	74.49	2457.8	3	-	-	-	-
ogbg-molhiv	41,127	25.5	27.5	2	-	-	+(9)	+(3)

PROTEINS Borgwardt et al. (2005) is a bioinformatics graph dataset representing proteins, where nodes are secondary structure elements, and the goal is to classify whether a protein is an enzyme or not.

D&D Shervashidze et al. (2011) is a bioinformatics graph dataset where nodes represent amino acids, designed for the task of predicting whether a protein structure is an enzyme or a non-enzyme.

MUTAG Debnath et al. (1991) is a molecular graph dataset consisting of nitro aromatic compounds, where the objective is to predict their mutagenic effect on a specific bacterium.

NCI1 Kim et al. (2018); Wale et al. (2008); Shervashidze et al. (2011) is a molecular graph dataset representing chemical compounds, where the binary classification task determines if the compound possesses anti-cancer activity against non-small cell lung cancer.

IMDB-BINARY Cai & Wang (2018) is a social network dataset derived from movie collaborations, where nodes are actors/actresses and the task is to classify the movie genre (Action vs. Romance).

REDDIT-BINARY Yanardag & Vishwanathan (2015) is a social network dataset representing online discussion threads, where nodes are users and edges represent responses, with the task of predicting the subreddit community where the thread originated.

COLLAB Leskovec et al. (2005) is a social network dataset derived from scientific collaboration networks, where the goal is to determine the field of physics (High Energy, Condensed Matter, or Astro Physics) to which a researcher belongs.

ogbg-molhiv Hu et al. (2020) is a molecular graph classification dataset adopted from MoleculeNet Wu et al. (2017) and is among the largest datasets in the collection. Each graph in the dataset represents a molecule, with nodes as atoms and edges as chemical bonds, and input node features are 9-dimensional, including atomic number, chirality, formal charge, and whether the atom is in a ring.

Detailed statistics of the datasets from TUDataset series Morris et al. (2020) are summarized in Table 6. Following the data processing protocol of IGL-Bench Qin et al. (2025b), we initialize node features based on dataset characteristics. Specifically, for molecular and bioinformatics graph datasets with node labels, we utilize one-hot encodings of node labels. For social network datasets without node attributes, we employ one-hot encodings of node degrees as input features.

D.2 Baselines

We compare our proposed method against a comprehensive set of baselines tailored to each specific task.

For **self-supervised graph classification**, the comparison encompasses four distinct categories: 1) *Supervised learning backbones*, namely GCN Kipf & Welling (2017) and GIN Xu et al. (2019), serving as upper-bound references; 2) *Traditional kernel-based methods*, including SP Borgwardt & Kriegel (2005), GK Shervashidze et al. (2009), WL Shervashidze et al. (2011), and DGK Yanardag & Vishwanathan (2015); 3) *Unsupervised graph embedding methods*, specifically node2vec Grover & Leskovec (2016), sub2vec Adhikari et al. (2018), and graph2vec Narayanan et al. (2017); and 4) *State-of-the-art graph contrastive learning methods*, including InfoGraph Sun et al. (2020), GraphCL You

et al. (2020), AD-GCL Suresh et al. (2021), JOAOv2 You et al. (2021), RGCL Li et al. (2022), SimGrace Xia et al. (2022a), SEGA Wu et al. (2023), and AutoGCL Yin et al. (2022).

For the **imbalanced low-label supervised graph classification** task, we include methods established in the IGL-Bench Qin et al. (2025b) benchmark: G^2GNN Wang et al. (2022), TopoImb Zhao et al. (2022), DataDec Zhang et al. (2023), ImGKB Tang & Liang (2023), and SOLT-GNN Liu et al. (2022). We further report another five latest baselines RAHNet Mao et al. (2023), ImbGNN Xu et al. (2024), C3GNN Ju et al. (2025), KDEX Mao et al. (2025) and SIMBA Qin et al. (2025a). We implement our framework using both GIN Xu et al. (2019) and GCN Kipf & Welling (2017) as backbones.

D.3 Experimental Protocol Details

For **self-supervised graph classification**, specific implementation follows these steps: The model is first trained in an unsupervised manner to learn graph representations. The quality of these embeddings is then assessed using a linear SVM classifier. We report the mean accuracy obtained from a 10-fold cross-validation procedure, averaged over 5 independent runs. Within each fold, the optimal hyperparameters for the SVM are selected via cross-validation. During the representation learning phase, the dataset follows an 80%/10%/10% random split for training, validation, and testing, respectively.

For **imbalanced low-label supervised graph classification**, we adopt the fixed data splits where the dataset is partitioned into 10% for training, 10% for validation, and 80% for testing. To simulate real-world data scarcity and long-tailed distributions, the training set is manipulated to exhibit varying imbalance ratios as per IGL-Bench settings.

D.4 Evaluation Metrics

To provide a comprehensive performance assessment, particularly under potential class or size imbalance, we employ four standard metrics: Accuracy (Acc), Balanced Accuracy (Bacc), Macro-F1 Score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Let TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. The metrics are defined as follows:

- **Accuracy:** The most intuitive metric, representing the ratio of correctly predicted observations to the total observations.
$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{40}$$
- **Balanced Accuracy:** Calculated as the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate). It avoids inflated performance estimates on imbalanced datasets.
$$\text{Bacc} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{41}$$
- **Macro-F1 Score:** The unweighted mean of the F1-score computed for each class. It treats all classes equally, regardless of their support, making it robust against class imbalance.
$$\text{Macro-F1} = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \tag{42}$$
where P_i and R_i denote the Precision and Recall for class i, and C is the number of classes.
- **AUC-ROC:** The area under the ROC curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. It measures the model's capability to distinguish between classes independently of the decision threshold.

D.5 Hyperparameter Settings

Practical Tuning Complexity. Despite the apparent number of components, SimGraph requires manual tuning of only **two** hyperparameters in practice: the balance weight α and the asymmetric scaling factor r. The loss weights $\lambda_1, \lambda_2, \lambda_3$ are automatically balanced by GradNorm Chen et al.

(2018). The angular bias b_θ is selected from a small discrete set $\{0.2, 0.3, 0.4\}$ via validation (default 0.3). The learnable bias b is an `nn.Parameter` initialized at -10 and optimized end-to-end. The momentum coefficient η is fixed at 0.999, and the virtual graph budget M is fixed at 20 or 100 per class. As shown in the sensitivity studies (Appendix J), both α and r exhibit wide stable plateaus, confirming that SimGraph does not require extensive hyperparameter search.

In low-label supervised graph classification, the hyperparameters are categorized into four groups: (1) General training settings, (2) BPGS Loss parameters, (3) SAIN architecture parameters, and (4) Virtual Graph Augmentation settings. For BPGS Loss, we finetune the balance weight α and asymmetric scaling factor r. For the Size-Adaptive Instance Normalization (SAIN), we set the size embedding dimension according to the graph encoder hidden dimension. The size encoder is a 2-layer MLP with ReLU activation. In self-supervised graph classification, SAIN Module and Virtual Graph methods are not adopted; the hyperparameters are categorized into three groups: (1) General training settings, (2) BPGS Loss parameters, (3) GraphCL-style You et al. (2020) augmentation settings in dropping edges or nodes, attribute masking, and subgraph sampling. We provide a detailed listing of the hyperparameter search spaces and default values used in our experiments in Table 7 and Table 8 for low-label supervised and self-supervised tasks, respectively.

Table 7: Hyperparameter search space and default settings in low-label supervised graph classification.

Category	**Hyperparameter**	**Search Space / Value**
General Settings	Hidden Dimension (d)	{32, 64, 128}
	GNN Layers	{2, 3, 4}
	Learning Rate	{1e-3, 5e-3, 1e-2}
	Weight Decay	5e-4
	Dropout	{0.1,0.5}
	Batch Size (B)	128
	Training Epochs	500
BPGS Loss	balance weight (α)	{1e-4, 0.5}
	asymmetric scaling factor (r)	{1.0, 3.0}
	angular bias (b_θ)	{0.2, 0.3, 0.4}
	learnable decision boundary bias (b)	initial value -10
SAIN Module	Size Embed. Dim (d_s)	{$d/2$,d,$2d$}
	Size Encoder Layers	2
Virtual Graph Augmentation	Virtual Graph Bank Budget (M)	20,100
	Mixup Beta (α_{mix})	0.2

D.6 Hardware and Computation Environment

All experiments for SimGraph were conducted on a server equipped with four NVIDIA A100 GPUs (80GB CUDA memory each) and 387GB RAM. The model was implemented in Python using PyTorch Paszke et al. (2019) 2.4.0 (with CUDA 12.4) and PyTorch-Geometric 2.6.1 Fey & Lenssen (2019).

E Self-Supervised Extension of SimGraph

E.1 Self-supervised Graph Classification

We also consider a self-supervised setting where $\mathcal{G}_{\text{unlabeled}} = \{G_i\}_{i=1}^{N_u}$ denotes a collection of unlabeled graphs. The goal is to learn a graph encoder $f_\theta : \mathcal{G} \to \mathbb{R}^d$ on $\mathcal{G}_{\text{unlabeled}}$ that produces meaningful representations without label supervision, which can later be transferred to downstream tasks through linear evaluation.

E.2 Methods of Self-supervised BPGS

While presented for low-label supervised tasks, BPGS is readily adaptable to fully unsupervised settings by utilizing pseudo-labels or augmentation consistency to define pairwise relationships y_{ij}.

Table 8: Hyperparameter search space and default settings in self-supervised graph classification.

Category	Hyperparameter	Search Space / Value
General Settings	Hidden Dimension (d)	{128, 256, 512}
	GNN Layers	{2, 3}
	Learning Rate	{1e-3, 5e-3, 1e-2}
	Weight Decay	5e-4
	Dropout	0.5
	Batch Size (B)	256
	Training Epochs	100
BPGS Loss	balance weight (α)	{1.0, 0.5}
	asymmetric scaling factor (r)	{1.0}
Augmentation View settings	Subgraph Sampling Ratio	0.9
	Drop Edge Ratio	0.1
	Drop Node Ratio	0.1
	Mask Node Ratio	0.1
	Number of Views	2

While supervised BPGS directly addresses the imbalance with labeled data, abundant unlabeled graphs can be leveraged through self-supervised pretraining. BPGS naturally extends to the self-supervised setting by replacing labeled pairs with augmentation-based pairs, while preserving the same loss formulation.

Augmentation-based pair construction. We generate positive pairs by applying stochastic augmentation (You et al., 2020) to each graph. For a source graph G_i, we sample two augmented views denoted as G_i' and G_i''. Views from the same source graph form positive pairs. Negative pairs are formed between views from different source graphs: for G_i' from source G_i and G_j' from source G_j where $i \neq j$, they constitute a negative pair.

Self-supervised loss. Let $\mathbf{h}_i'$ and $\mathbf{h}_i''$ denote embeddings of augmented views G_i' and G_i'' from the same source graph G_i. The self-supervised loss applies the same BPGS formulation:

$$\mathcal{L}_{\text{BPGS}}^{\text{SSL}} = \mathbb{E}\big\{\alpha \cdot \log\left(1 + \exp\left(-\frac{1}{r}(\mathcal{S}(\mathbf{h}_i', \mathbf{h}_i'') + b)\right)\right) + (1-\alpha) \cdot \log\left(1 + \exp\left(r(\mathcal{S}(\mathbf{h}_i', \mathbf{h}_j') + b)\right)\right)\big\}, \tag{43}$$

where the first term measures loss for positive pairs (G_i', G_i'' from the same source), and the second term for negative pairs (G_i', G_j' from different sources). The hyperparameters r and b serve the same roles as in supervised learning. We recommend setting $\alpha \approx 1.0$ to avoid treating semantically similar graphs from different sources as strong negatives to reduce the risk of false negatives where semantically similar graphs from different sources are incorrectly pushed apart.

Self-supervised pretraining. In the pretraining stage, we optimize only the contrastive similarity loss $\mathcal{L}_{\text{BPGS}}^{\text{SSL}}$. After pretraining, we freeze the GNN encoder and evaluate the learned graph representations using a linear SVM classifier.

E.3 Experiments of Self-supervised Graph Classification

E.3.1 Experimental Setup

Datasets. Similar to the low-label supervised setting, we use the TUDatasets dataset series for evaluation of self-supervised tasks.

Baselines. For **self-supervised graph classification**, the comparison encompasses four categories: supervised backbones (e.g., GIN, GCN), traditional kernel-based methods, unsupervised graph embeddings, and state-of-the-art graph contrastive learning methods.

Protocol and Implementation. For **self-supervised learning**, we strictly adhere to the standard evaluation protocol utilized by previous state-of-the-art studies Sun et al. (2020); You et al. (2020), where the model is first trained in an unsupervised manner, followed by linear evaluation using an SVM classifier. We report the mean accuracy from a 10-fold cross-validation procedure.

Table 9: Supervised GIN/GCN and self-supervised classification accuracy (%) on TUDatasets. **Bold** and underlined highlight the best and second-best results in self-supervised methods respectively.

Model	PROTEINS	MUTAG	D&D	NCI1	IMDB-B	RDT-B	COLLAB
GCN Kipf & Welling (2017)	74.92±0.33	85.63±0.24	76.24±0.14	80.20±0.14	70.45±0.37	50.00±0.10	79.01±0.18
GIN Xu et al. (2019)	76.28±0.28	89.47±0.16	78.91±0.13	82.75±0.19	73.70±0.60	92.44±0.25	80.23±0.19
SP Borgwardt & Kriegel (2005)	75.07±0.54	85.25±0.24	$>$ 10 hours	73.53±0.16	55.62±0.02	64.13±0.04	$>$ 10 hours
GK Shervashidze et al. (2009)	71.67±0.55	81.71±0.21	78.53±0.03	66.06±0.12	65.93±0.10	77.32±0.02	71.81±0.31
WL Shervashidze et al. (2011)	72.92±0.56	80.76±0.30	79.78±0.36	80.01±0.50	72.30±0.44	68.82±0.41	69.30±0.42
DGK Yanardag & Vishwanathan (2015)	73.21±0.61	87.51±0.65	74.79±0.32	79.98±0.36	67.09±0.37	78.35±0.43	64.43±0.48
node2vec Grover & Leskovec (2016)	57.58±0.36	72.62±1.02	$>$ 10 hours	54.93±0.16	50.25±0.09	$>$ 10 hours	56.12±0.02
sub2vec Adhikari et al. (2018)	53.06±0.56	61.17±1.59	54.33±0.24	52.82±0.15	55.34±0.15	71.48±0.42	55.26±0.15
graph2vec Narayanan et al. (2017)	73.33±0.21	83.28±0.93	79.32±0.29	73.21±0.18	71.16±0.05	75.78±0.96	71.10±0.54
InfoGraph Sun et al. (2020)	74.27±0.43	88.65±1.09	73.02±1.31	76.33±0.79	72.97±0.49	83.06±0.82	70.41±0.34
GraphCL You et al. (2020)	74.41±0.35	86.62±1.23	78.49±0.36	77.54±0.34	71.29±0.38	89.50±0.73	71.28±1.16
AD-GCL Suresh et al. (2021)	73.33±0.36	88.70±1.66	75.74±0.68	74.15±0.62	70.35±0.59	90.04±0.76	72.01±0.53
JOAOv2 You et al. (2021)	74.02±0.53	87.23±1.12	77.36±0.29	78.36±0.21	70.12±0.25	86.36±0.30	69.05±0.22
RGCL Li et al. (2022)	75.11±0.58	87.62±1.16	78.77±0.52	78.14±0.89	71.67±0.76	90.27±0.54	70.85±0.63
SimGrace Xia et al. (2022a)	75.26±0.24	89.23±1.25	77.13±0.88	79.06±0.15	71.22±0.30	89.41±0.33	71.26±0.51
AutoGCL Yin et al. (2022)	69.80±0.38	85.22±1.43	75.76±0.93	78.39±0.58	72.26±0.34	86.56±1.31	71.40±0.26
SEGA Wu et al. (2023)	76.00±0.68	90.23±0.89	78.81±0.33	79.15±0.52	73.60±0.33	90.12±0.46	74.10±0.62
SimGraph	**82.68±1.01**	**92.00±2.74**	**82.52±0.70**	**80.44±0.13**	**77.20±1.94**	**90.40±0.22**	**77.48±2.52**

E.3.2 Results and Analysis.

Self-supervised Graph Classification Results. Table 9 presents the self-supervised learning results across eight benchmark datasets. SimGraph consistently achieves state-of-the-art performance, demonstrating an average improvement of **2.67%** over the best contrastive learning baselines. Notably, on PROTEINS, SimGraph achieves 82.68% accuracy, surpassing SEGA by **6.7%**. This substantial improvement validates that our BPGS loss effectively addresses the false negative problem in contrastive learning, particularly benefiting datasets with diverse graph structures.

F Additional Results on IGL-Bench

F.1 Performance of Graph Size Imbalance Classification

Table 10: **Balanced Accuracy** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GIN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GIN (bb.) Xu et al. (2019)	51.08±2.07	63.46±0.92	66.38±4.27	67.41±2.32
SOLT-GNN Liu et al. (2022)	45.47±2.10	62.58±0.61	64.20±5.46	49.57±6.78
TopoImb Zhao et al. (2022)	50.07±2.77	67.24±1.53	66.00±2.41	69.47±3.70
ImbGNN Xu et al. (2024)	52.18±0.82	65.92±1.84	66.17±2.35	66.71±8.45
SIMBA Qin et al. (2025a)	54.71±0.31	64.72±0.49	69.04±0.26	63.62±1.19
SimGraph	**56.38±1.20**	**76.96±1.07**	**70.25±0.67**	**73.08±2.30**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GIN (bb.) Xu et al. (2019)	49.83±1.30	62.22±0.98	65.08±5.78	68.32±1.77
SOLT-GNN Liu et al. (2022)	50.06±0.75	59.61±3.40	69.38±1.23	73.51±1.14
TopoImb Zhao et al. (2022)	50.88±2.36	65.25±2.43	68.10±0.87	71.54±0.75
ImbGNN Xu et al. (2024)	47.55±4.57	68.71±4.77	68.88±2.81	67.64±2.53
SIMBA Qin et al. (2025a)	51.53±2.16	63.76±1.07	67.08±0.39	74.11±0.21
SimGraph	**58.66±0.98**	**74.57±1.35**	**69.75±1.63**	**80.21±0.33**
High level ρ_{size}	4.93	8.91	4.45	10.97
GIN (bb.) Xu et al. (2019)	50.20±0.69	61.43±2.36	62.60±3.82	67.41±2.23
SOLT-GNN Liu et al. (2022)	48.02±1.77	60.75±6.03	65.03±4.12	60.24±2.11
TopoImb Zhao et al. (2022)	52.07±0.96	66.48±1.89	66.75±0.91	69.14±4.83
ImbGNN Xu et al. (2024)	52.02±0.31	65.74±1.35	67.71±2.70	69.56±0.36
SIMBA Qin et al. (2025a)	50.71±1.00	66.32±0.61	68.37±1.62	72.37±1.73
SimGraph	**58.36±0.61**	**74.92±0.48**	**69.00±3.19**	**77.94±0.58**

Table 11: **Macro-F1** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GIN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GIN (bb.) Xu et al. (2019)	48.26±3.11	63.39±0.84	65.67±5.40	66.14±1.90
SOLT-GNN Liu et al. (2022)	43.63±1.91	62.34±0.66	63.66±5.94	49.17±6.76
TopoImb Zhao et al. (2022)	48.87±2.90	64.97±1.65	65.71±2.62	69.15±3.83
ImbGNN Xu et al. (2024)	49.07±2.94	65.51±2.58	64.55±3.74	66.64±8.49
SIMBA Qin et al. (2025a)	54.60±0.22	64.87±0.46	68.73±0.25	59.78±1.92
SimGraph	**55.45±1.66**	**74.77±2.05**	**70.08±0.53**	**71.85±2.72**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GIN (bb.) Xu et al. (2019)	38.58±5.97	60.89±1.84	63.28±9.06	67.20±2.32
SOLT-GNN Liu et al. (2022)	43.20±5.36	58.67±4.91	68.73±1.97	73.24±1.40
TopoImb Zhao et al. (2022)	49.17±0.95	64.65±1.76	67.98±0.95	71.52±0.75
ImbGNN Xu et al. (2024)	47.20±4.44	67.96±4.32	68.56±2.64	65.69±4.50
SIMBA Qin et al. (2025a)	41.55±7.26	63.77±1.34	67.07±0.37	72.98±0.23
SimGraph	**58.40±0.85**	**74.88±1.28**	**69.73±1.63**	**80.17±0.36**
High level ρ_{size}	4.93	8.91	4.45	10.97
GIN (bb.) Xu et al. (2019)	34.56±6.32	57.98±5.51	59.75±6.69	66.20±2.77
SOLT-GNN Liu et al. (2022)	40.70±3.27	58.50±10.48	64.53±4.68	54.80±3.23
TopoImb Zhao et al. (2022)	51.65±1.07	63.97±2.78	66.67±0.91	68.41±5.34
ImbGNN Xu et al. (2024)	50.56±1.10	65.22±2.13	67.22±2.33	69.48±0.40
SIMBA Qin et al. (2025a)	39.00±3.66	66.69±0.67	67.74±2.33	71.91±1.63
SimGraph	**57.86±0.93**	**75.46±0.56**	**67.87±4.14**	**77.88±0.60**

Table 12: **AUC-ROC** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GIN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GIN (bb.) Xu et al. (2019)	51.36±2.71	66.09±0.71	71.34±4.89	78.51±9.44
SOLT-GNN Liu et al. (2022)	45.70±3.17	68.55±0.93	71.29±6.08	48.18±7.14
TopoImb Zhao et al. (2022)	49.33±1.90	73.92±3.23	72.37±2.68	65.86±4.55
ImbGNN Xu et al. (2024)	53.90±1.83	69.26±3.72	67.91±1.52	62.67±6.55
SIMBA Qin et al. (2025a)	53.68±0.53	67.18±1.69	73.35±1.19	**85.35±0.71**
SimGraph	**56.89±1.93**	**81.73±1.51**	**75.99±2.61**	73.67±13.99
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GIN (bb.) Xu et al. (2019)	48.84±2.20	67.45±0.86	68.30±6.16	78.44±2.37
SOLT-GNN Liu et al. (2022)	50.40±2.91	68.57±2.72	**76.53±1.52**	76.14±4.50
TopoImb Zhao et al. (2022)	52.09±1.56	72.48±2.37	73.33±2.00	74.96±1.51
ImbGNN Xu et al. (2024)	49.92±2.68	73.23±7.11	70.40±1.10	62.91±8.88
SIMBA Qin et al. (2025a)	49.72±0.60	69.02±1.09	70.40±1.05	80.71±3.92
SimGraph	**60.10±1.02**	**80.24±1.03**	73.17±2.68	**81.76±3.55**
High level ρ_{size}	4.93	8.91	4.45	10.97
GIN (bb.) Xu et al. (2019)	49.85±1.83	70.66±0.82	69.92±3.25	75.86±5.42
SOLT-GNN Liu et al. (2022)	47.82±3.75	73.57±5.28	70.85±5.29	43.84±5.55
TopoImb Zhao et al. (2022)	52.26±1.24	73.84±1.92	72.55±1.25	69.04±9.48
ImbGNN Xu et al. (2024)	51.99±2.60	68.58±3.56	69.38±1.69	71.65±2.89
SIMBA Qin et al. (2025a)	49.13±1.22	72.49±1.79	**75.34±1.40**	**78.96±4.14**
SimGraph	**59.50±1.16**	**82.02±0.78**	72.75±4.65	74.02±8.81

Table 13: **Balanced Accuracy** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GCN (bb.) Kipf & Welling (2017)	50.47±0.75	63.13±0.43	69.15±1.44	53.44±1.78
SOLT-GNN Liu et al. (2022)	48.24±1.20	63.81±1.86	68.23±2.48	37.58±3.04
TopoImb Zhao et al. (2022)	50.43±0.87	57.64±0.94	68.20±0.70	56.94±3.23
ImbGNN Xu et al. (2024)	53.36±0.96	65.60±0.54	68.66±2.53	66.61±0.77
SIMBA Qin et al. (2025a)	51.83±2.05	62.16±1.61	70.50±0.10	64.92±2.24
SimGraph	**54.92±0.83**	**76.90±0.58**	**71.04±0.72**	**70.48±3.76**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GCN (bb.) Kipf & Welling (2017)	49.74±2.52	58.27±4.65	67.95±2.82	67.96±0.89
SOLT-GNN Liu et al. (2022)	51.13±0.79	59.12±2.47	70.25±1.15	61.95±5.55
TopoImb Zhao et al. (2022)	54.34±3.22	64.63±3.42	68.75±0.76	69.12±0.52
ImbGNN Xu et al. (2024)	50.42±1.28	67.35±5.02	68.83±2.76	67.35±2.55
SIMBA Qin et al. (2025a)	51.10±1.55	63.64±2.20	67.46±1.08	68.58±0.78
SimGraph	**55.98±0.53**	**74.81±0.57**	**70.58±0.31**	**82.08±1.44**
High level ρ_{size}	4.93	8.91	4.45	10.97
GCN (bb.) Kipf & Welling (2017)	49.96±0.54	57.58±4.32	64.88±2.02	66.38±0.46
SOLT-GNN Liu et al. (2022)	51.11±1.15	60.94±1.68	69.80±2.07	67.05±2.54
TopoImb Zhao et al. (2022)	52.12±0.78	66.63±4.54	66.60±0.91	69.09±1.00
ImbGNN Xu et al. (2024)	51.90±4.33	65.58±1.19	68.04±3.11	67.83±2.02
SIMBA Qin et al. (2025a)	52.10±1.33	58.90±3.34	68.58±1.25	67.31±2.48
SimGraph	**58.01±0.74**	**73.60±2.16**	**70.67±0.51**	**79.63±0.97**

Table 14: **Macro-F1** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GCN (bb.) Kipf & Welling (2017)	36.72±5.58	63.09±0.46	69.02±1.52	49.68±8.19
SOLT-GNN Liu et al. (2022)	45.00±3.14	63.82±2.01	67.88±2.35	35.00±2.77
TopoImb Zhao et al. (2022)	44.72±2.61	55.77±1.91	67.78±1.00	56.09±3.68
ImbGNN Xu et al. (2024)	48.62±2.32	65.41±0.42	68.10±2.48	66.08±0.93
SIMBA Qin et al. (2025a)	51.69±2.07	62.03±1.52	70.33±0.15	64.75±2.16
SimGraph	**52.69±1.49**	**73.82±0.96**	**71.00±0.77**	**69.00±3.97**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GCN (bb.) Kipf & Welling (2017)	40.94±5.67	54.22±8.96	67.80±2.84	67.18±1.39
SOLT-GNN Liu et al. (2022)	45.63±4.84	58.14±3.23	70.05±1.19	57.57±8.59
TopoImb Zhao et al. (2022)	51.99±3.88	62.10±6.59	68.64±0.77	69.10±0.49
ImbGNN Xu et al. (2024)	49.65±0.72	67.47±5.38	68.25±2.45	67.29±2.60
SIMBA Qin et al. (2025a)	39.57±4.47	63.54±2.73	67.22±1.21	68.53±0.73
SimGraph	**54.74±1.19**	**75.34±0.46**	**70.55±0.31**	**82.05±1.48**
High level ρ_{size}	4.93	8.91	4.45	10.97
GCN (bb.) Kipf & Welling (2017)	39.32±7.82	52.46±8.79	63.65±3.15	65.91±0.31
SOLT-GNN Liu et al. (2022)	39.57±6.16	60.32±2.31	69.54±2.16	65.40±3.77
TopoImb Zhao et al. (2022)	51.43±0.98	65.66±5.28	66.54±0.89	68.76±1.17
ImbGNN Xu et al. (2024)	48.77±4.86	64.77±1.47	67.33±3.20	67.68±2.02
SIMBA Qin et al. (2025a)	44.61±3.61	55.94±6.24	68.41±1.04	66.20±2.62
SimGraph	**57.96±0.78**	**74.15±2.30**	**70.56±0.42**	**79.54±1.01**

Table 15: **AUC-ROC** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Low level ρ_{size}	1.60	1.56	1.64	2.79
GCN (bb.) Kipf & Welling (2017)	50.81±2.98	64.25±0.94	74.69±1.04	50.48±13.45
SOLT-GNN Liu et al. (2022)	46.36±0.64	65.61±1.44	74.16±1.36	32.50±5.56
TopoImb Zhao et al. (2022)	52.94±0.50	60.66±1.62	74.55±1.80	63.78±2.65
ImbGNN Xu et al. (2024)	51.81±2.77	69.38±0.75	67.89±1.38	66.49±2.54
SIMBA Qin et al. (2025a)	51.60±1.14	63.82±1.49	**76.98±0.84**	68.58±2.01
SimGraph	**56.71±3.03**	**81.02±2.24**	76.90±0.70	**82.68±3.53**
Mid level ρ_{size}	2.90	2.53	2.11	4.92
GCN (bb.) Kipf & Welling (2017)	50.03±4.80	61.45±4.93	74.23±3.12	74.51±2.91
SOLT-GNN Liu et al. (2022)	52.54±0.92	59.75±4.06	**78.21±2.21**	66.93±6.74
TopoImb Zhao et al. (2022)	**57.29±2.45**	74.47±2.44	74.30±0.66	77.25±0.41
ImbGNN Xu et al. (2024)	51.87±1.77	71.23±7.75	70.52±2.22	65.54±1.06
SIMBA Qin et al. (2025a)	50.62±1.38	64.44±2.23	73.78±0.44	73.06±0.61
SimGraph	56.85±0.37	**78.66±1.04**	75.50±2.42	**87.07±1.63**
High level ρ_{size}	4.93	8.91	4.45	10.97
GCN (bb.) Kipf & Welling (2017)	51.12±1.94	65.50±6.38	68.31±3.07	73.41±0.82
SOLT-GNN Liu et al. (2022)	49.47±4.36	64.79±0.57	**78.04±1.14**	72.92±1.77
TopoImb Zhao et al. (2022)	53.65±2.72	74.37±3.19	71.84±1.07	75.52±1.17
ImbGNN Xu et al. (2024)	55.62±4.30	68.02±2.42	68.71±1.81	67.16±5.29
SIMBA Qin et al. (2025a)	50.56±0.59	64.35±3.82	74.57±1.45	71.90±4.58
SimGraph	**58.08±0.72**	**80.04±2.17**	75.57±1.83	**86.04±1.67**

F.2 Performance of Class Imbalance Classification

Table 16: **Balanced Accuracy** (% ± stddev) on datasets with changing class imbalance levels over 10 runs, encoded by GIN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GIN (bb.) Xu et al. (2019)	51.40±1.64	63.37±3.26	66.05±2.57	76.66±4.80
G^{2}GNN Wang et al. (2022)	53.42±3.72	63.79±2.13	61.91±3.77	72.34±2.76
TopoImb Zhao et al. (2022)	53.99±2.19	67.19±2.46	66.28±1.85	73.99±1.18
RAHNet Mao et al. (2023)	54.08±2.67	66.37±1.17	70.38±1.34	78.16±1.17
DataDec Zhang et al. (2023)	53.70±4.25	62.96±1.21	64.22±5.73	79.80±7.38
ImGKB Tang & Liang (2023)	53.34±3.61	62.33±4.38	49.99±0.47	50.24±0.29
GDeR Zhang et al. (2024)	56.05±3.10	51.25±2.40	63.42±2.03	74.67±2.45
C^{3}GNN Ju et al. (2025)	52.41±2.60	65.91±1.00	70.71±1.76	74.06±2.26
KDEX Mao et al. (2025)	53.91±1.30	66.29±0.12	70.29±1.44	72.94±0.18
SimGraph	**58.08±1.28**	**74.51±0.52**	**70.83±1.31**	**80.88±1.90**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GIN (bb.) Xu et al. (2019)	51.36±2.25	57.33±3.16	63.54±3.46	62.89±4.61
G^{2}GNN Wang et al. (2022)	52.75±2.14	61.09±4.11	64.28±5.02	70.26±3.41
RAHNet Mao et al. (2023)	50.84±0.82	66.30±2.18	65.37±0.22	78.06±0.73
TopoImb Zhao et al. (2022)	50.44±3.75	58.33±4.40	61.14±4.56	61.34±6.80
DataDec Zhang et al. (2023)	53.42±2.33	61.27±1.04	67.95±2.77	78.68±4.93
ImGKB Tang & Liang (2023)	51.61±3.12	63.39±1.76	50.23±0.28	50.07±0.15
GDeR Zhang et al. (2024)	52.08±1.66	59.10±3.79	66.53±2.92	76.16±2.60
C^{3}GNN Ju et al. (2025)	50.87±3.06	61.54±1.86	65.70±2.35	72.90±0.92
KDEX Mao et al. (2025)	51.36±2.34	63.13±2.46	68.63±2.76	74.39±1.65
SimGraph	**56.39±1.68**	**74.75±0.15**	**72.08±0.65**	**80.51±1.11**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GIN (bb.) Xu et al. (2019)	47.89±2.12	52.47±4.58	57.46±2.77	59.54±7.11
G^{2}GNN Wang et al. (2022)	50.76±1.78	53.17±2.90	61.85±5.36	65.97±4.07
TopoImb Zhao et al. (2022)	49.99±0.93	52.05±1.31	52.82±3.06	55.87±4.09
RAHNet Mao et al. (2023)	51.70±2.40	59.16±1.45	62.42±2.47	75.65±0.68
DataDec Zhang et al. (2023)	55.22±3.81	61.60±2.60	66.30±4.93	77.77±5.49
ImGKB Tang & Liang (2023)	51.38±3.53	59.14±4.10	50.07±0.35	50.28±0.23
GDeR Zhang et al. (2024)	52.06±2.44	57.25±4.19	62.37±1.50	68.25±1.26
C^{3}GNN Ju et al. (2025)	49.91±0.21	57.77±3.73	63.94±1.14	75.72±1.87
KDEX Mao et al. (2025)	55.30±1.33	59.52±5.49	65.76±3.44	76.53±1.72
SimGraph	**55.60±0.97**	**74.43±0.50**	**67.21±2.76**	**82.13±1.54**

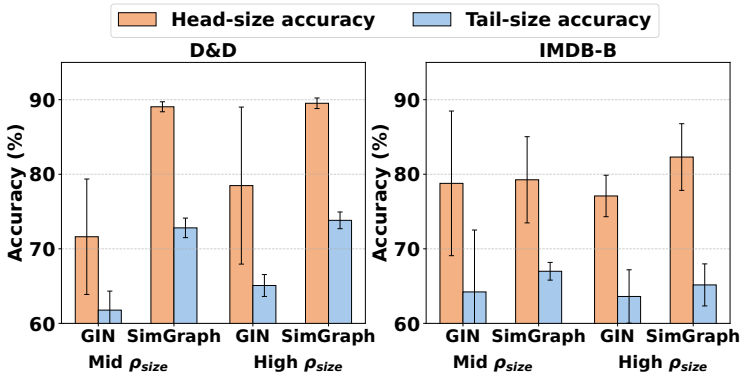


Figure 4: Accuracy on head-size and tail-size graphs in D&D and IMDB-B by vanilla GIN and SimGraph(GIN).

Table 17: **Macro-F1** (% ± stddev) on datasets with changing class imbalance levels over 10 runs, encoded by GIN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GIN (bb.) Xu et al. (2019)	48.85±3.33	59.87±7.24	65.24±3.21	76.20±5.00
G^2GNN Wang et al. (2022)	52.88±3.70	63.31±2.42	60.99±3.50	71.74±3.14
TopoImb Zhao et al. (2022)	48.33±3.70	65.72±3.45	65.85±2.29	73.41±0.01
RAHNet Mao et al. (2023)	52.27±3.69	66.04±1.43	70.18±1.42	78.00±1.29
DataDec Zhang et al. (2023)	52.98±4.65	62.83±1.20	62.42±8.07	79.02±8.63
ImGKB Tang & Liang (2023)	50.36±5.89	61.06±7.96	35.27±4.74	33.97±0.42
GDeR Zhang et al. (2024)	55.12±2.70	45.13±5.90	61.67±2.67	74.53±2.56
C^3GNN Ju et al. (2025)	49.11±4.37	65.87±1.06	70.66±1.81	73.46±2.70
KDEX Mao et al. (2025)	53.40±1.03	65.87±0.25	70.17±1.39	72.67±0.13
SimGraph	**56.62±2.90**	**74.41±0.61**	**70.82±1.30**	**80.87±1.90**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GIN (bb.) Xu et al. (2019)	46.63±2.95	49.89±6.19	59.66±5.94	56.10±6.75
G^2GNN Wang et al. (2022)	47.64±7.63	55.34±8.20	61.37±8.64	69.03±3.78
TopoImb Zhao et al. (2022)	40.26±2.06	47.37±9.53	56.78±8.24	54.07±11.22
RAHNet Mao et al. (2023)	45.67±5.23	66.04±2.26	64.68±0.40	77.95±0.75
DataDec Zhang et al. (2023)	52.31±3.50	61.12±1.22	67.42±3.69	77.96±5.66
ImGKB Tang & Liang (2023)	44.62±8.86	62.91±2.34	32.70±0.62	32.81±1.21
GDeR Zhang et al. (2024)	39.57±8.12	58.77±3.91	65.46±3.15	75.82±2.65
C^3GNN Ju et al. (2025)	49.31±4.38	60.39±1.93	63.59±2.66	71.76±1.13
KDEX Mao et al. (2025)	46.95±5.45	62.14±2.47	68.16±3.44	73.12±2.05
SimGraph	**56.01±1.68**	**74.40±0.19**	**71.97±0.73**	**80.26±1.09**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GIN (bb.) Xu et al. (2019)	35.14±4.01	38.83±8.36	47.51±5.47	48.74±11.66
G^2GNN Wang et al. (2022)	37.35±7.10	42.22±11.10	56.79±9.27	61.15±6.30
TopoImb Zhao et al. (2022)	32.16±3.27	34.24±2.40	36.64±6.35	41.81±9.90
RAHNet Mao et al. (2023)	36.69±12.92	52.91±5.90	58.02±3.43	74.22±0.30
DataDec Zhang et al. (2023)	**54.76±4.61**	61.67±2.67	64.82±8.14	77.39±5.38
ImGKB Tang & Liang (2023)	41.35±7.92	57.16±5.54	32.23±1.61	34.31±2.68
GDeR Zhang et al. (2024)	45.47±10.48	55.28±6.79	62.17±1.72	65.16±0.90
C^3GNN Ju et al. (2025)	30.19±2.97	54.66±5.27	61.60±2.60	74.60±1.91
KDEX Mao et al. (2025)	49.58±6.04	58.70±5.99	63.66±4.69	75.29±1.98
SimGraph	53.80±1.89	**74.48±0.65**	**66.82±3.21**	**81.62±1.80**

G Additional Benchmark Comparisons

This section reports additional benchmark comparisons that complement the main results: comparisons with size-generalization methods, multi-class classification on COLLAB, and heterogeneous graphs. The full class-imbalance results (Table 2) are reported in the main text.

G.1 Comparison with Size-Generalization Methods

Size-generalization methods target a regime complementary to ours: training predominantly on small graphs and extrapolating to unseen larger ones. To position SimGraph against this line of work, we adapt Bevilacqua et al. Bevilacqua et al. (2021) and SizeShiftReg Buffelli et al. (2022) to the IGL-Bench protocol (GIN encoder, high size imbalance). As shown in Table 22, both adapted methods improve over the vanilla GIN backbone yet remain 3.0–3.8pp below SimGraph, since their invariance objectives stabilize individual graph representations but do not correct the pairwise similarity bias that dominates under size imbalance.

G.2 Multi-class and Heterogeneous Graphs

Multi-class classification. To evaluate beyond binary tasks, we run SimGraph on COLLAB (3 classes) under the IGL-Bench size-imbalance protocol (Table 23). SimGraph achieves the best Balanced Accuracy and Macro-F1 at all three imbalance levels and the best Accuracy at the Low and High levels (+3.28 and +3.11 over the strongest baseline); at the Mid level its Accuracy is statistically tied with SOLT-GNN (69.23 vs. 69.69, with overlapping standard deviations) while its bAcc and

Table 18: **AUC-ROC** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GIN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GIN (bb.) Xu et al. (2019)	52.26±2.95	66.54±5.28	73.99±2.41	**87.61±4.35**
G^{2}GNN Wang et al. (2022)	54.55±5.56	70.05±2.41	64.87±4.48	80.22±4.01
TopoImb Zhao et al. (2022)	56.93±4.29	72.86±1.46	74.83±1.01	81.49±3.03
RAHNet Mao et al. (2023)	54.55±0.85	71.12±3.04	**76.62±2.07**	65.40±4.91
DataDec Zhang et al. (2023)	54.57±6.02	68.18±1.32	72.25±5.65	81.26±13.78
ImGKB Tang & Liang (2023)	54.09±5.28	66.54±2.08	50.60±1.23	74.36±5.01
GDeR Zhang et al. (2024)	55.67±2.06	65.54±7.75	70.75±2.52	79.78±2.90
C^{3}GNN Ju et al. (2025)	55.77±2.50	64.00±7.19	74.98±2.53	64.09±6.06
KDEX Mao et al. (2025)	54.51±4.13	67.48±2.05	75.80±1.74	66.70±3.92
SimGraph	**58.12±2.32**	**81.08±0.54**	73.58±0.12	83.29±3.76
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GIN (bb.) Xu et al. (2019)	51.28±4.16	54.13±6.72	74.38±2.48	84.08±4.85
G^{2}GNN Wang et al. (2022)	51.90±3.85	64.49±5.43	69.02±3.60	77.97±4.62
TopoImb Zhao et al. (2022)	50.82±3.94	67.76±4.85	72.61±2.10	83.30±3.79
RAHNet Mao et al. (2023)	54.55±0.85	71.12±3.04	**76.62±2.07**	65.40±4.91
DataDec Zhang et al. (2023)	53.49±1.94	67.05±1.47	75.40±3.23	81.00±8.17
ImGKB Tang & Liang (2023)	52.89±4.84	53.59±1.12	51.06±1.08	76.25±3.29
GDeR Zhang et al. (2024)	55.67±2.06	65.54±4.53	70.75±2.52	79.78±5.57
C^{3}GNN Ju et al. (2025)	55.77±2.50	64.00±7.19	74.98±2.53	64.09±6.06
KDEX Mao et al. (2025)	56.97±4.13	67.48±2.05	75.80±1.74	66.70±3.92
SimGraph	**57.05±2.11**	**80.51±0.21**	74.46±3.74	**86.58±0.80**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GIN (bb.) Xu et al. (2019)	47.04±3.22	47.06±5.01	65.96±9.53	80.58±3.45
G^{2}GNN Wang et al. (2022)	49.32±4.07	60.76±4.90	65.79±7.00	72.55±5.64
TopoImb Zhao et al. (2022)	48.09±3.94	61.49±5.30	69.12±7.15	80.08±5.47
RAHNet Mao et al. (2023)	49.64±3.02	52.93±0.88	63.00±2.57	71.94±2.77
DataDec Zhang et al. (2023)	**57.66±5.55**	69.11±3.67	**74.89±4.11**	82.59±6.05
ImGKB Tang & Liang (2023)	52.80±5.19	63.79±4.04	51.04±1.57	71.65±6.48
GDeR Zhang et al. (2024)	50.92±4.20	65.73±1.55	63.43±3.83	57.73±1.33
C^{3}GNN Ju et al. (2025)	48.14±2.21	58.51±3.89	61.72±2.88	71.23±0.98
KDEX Mao et al. (2025)	52.66±1.41	63.07±6.46	64.48±7.47	63.68±2.64
SimGraph	54.90±2.27	**79.25±1.65**	66.35±2.29	**85.77±2.04**

Macro-F1 remain clearly ahead (+1.35 and +1.41 over the best baseline). These results confirm that the pairwise objective $y_{ij} = \mathbb{I}[y_i = y_j]$ generalizes naturally to multi-class settings.

Heterogeneous graphs. We further evaluate SimGraph against SOLT-GNN, KDEX, and SIMBA on two heterogeneous graph datasets, DBLP and Freebase, under both class- and size-imbalance at Low/Mid/High levels (Tables 24 and 25). SimGraph achieves the best results across all imbalance types, levels, and metrics on both datasets. On DBLP, it improves over the strongest baseline (SIMBA) by 1.0–2.3 points in Accuracy, and is notably robust under size imbalance—AUC-ROC stays $\approx$92.5 from Low to High—indicating that SAIN's size-decoupling transfers to heterogeneous settings. Freebase is substantially harder due to its many relation types and higher density, which make node-level semantics noisier; there SimGraph still leads by 1.8–3.1 points in Accuracy and its AUC-ROC remains well above chance (66–72), showing that BPGS still recovers useful pairwise class geometry. These results demonstrate that SimGraph is applicable to heterogeneous graphs without architectural changes, while relation-aware encoders for dense heterogeneous graphs remain a direction for future work.

G.3 Comparison with Label-Aware Contrastive Losses

To complement the structural comparison in Section 3 of the main text, we empirically compare BPGS against supervised contrastive objectives used as auxiliary similarity losses on top of the same cross-entropy classifier: CE + SupCon Khosla et al. (2020) (official `SupConLoss`) and CE + ProGCL Xia et al. (2022b) (official implementation, `mode=weight`). Both baselines use the same GIN encoder and IGL-Bench splits as SimGraph. As shown in Table 26, both improve over the

Table 19: **Balanced Accuracy** (% ± stddev) on datasets with changing class imbalance levels over 10 runs, encoded by GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GCN (bb.) Kipf & Welling (2017)	50.07±0.23	62.63±2.06	59.76±1.78	66.65±1.06
G^{2}GNN Wang et al. (2022)	50.02±0.06	60.38±5.83	58.35±3.47	67.02±2.55
TopoImb Zhao et al. (2022)	53.71±2.88	60.56±2.30	49.71±0.34	58.68±6.48
RAHNet Mao et al. (2023)	56.10±1.13	71.95±3.61	69.75±0.27	79.27±0.65
DataDec Zhang et al. (2023)	54.36±3.61	61.41±2.42	58.11±3.96	69.08±4.38
ImGKB Tang & Liang (2023)	53.03±3.56	62.84±2.56	50.15±0.39	50.29±0.19
GDeR Zhang et al. (2024)	54.79±3.24	56.69±2.53	63.25±6.38	65.48±5.59
C^{3}GNN Ju et al. (2025)	56.40±3.37	71.03±1.01	69.29±0.69	79.50±0.82
KDEX Mao et al. (2025)	56.09±0.98	68.81±0.56	70.58±0.69	78.48±1.17
SimGraph	**59.55±0.89**	**75.02±2.36**	**71.00±0.98**	**79.67±1.28**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GCN (bb.) Kipf & Welling (2017)	49.69±0.68	60.53±1.43	51.35±1.88	63.51±3.57
G^{2}GNN Wang et al. (2022)	49.73±0.87	65.61±2.52	53.84±3.33	66.54±2.09
TopoImb Zhao et al. (2022)	51.60±3.14	54.87±2.95	48.94±4.07	56.51±2.84
RAHNet Mao et al. (2023)	50.75±1.13	71.79±0.20	63.98±3.77	79.27±0.65
DataDec Zhang et al. (2023)	54.50±1.74	62.01±1.75	56.87±5.73	67.02±5.12
ImGKB Tang & Liang (2023)	51.70±2.33	63.70±3.66	50.14±0.22	50.08±0.16
GDeR Zhang et al. (2024)	52.94±3.24	57.58±2.11	63.25±6.38	65.48±5.59
C^{3}GNN Ju et al. (2025)	**56.40±3.37**	69.97±1.01	65.13±0.44	79.50±0.36
KDEX Mao et al. (2025)	56.09±0.98	68.81±0.56	68.73±1.01	78.48±1.17
SimGraph	54.16±0.14	**73.88±0.85**	**70.48±0.60**	**81.41±0.50**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GCN (bb.) Kipf & Welling (2017)	50.36±1.14	53.22±2.13	50.08±0.16	58.47±4.64
G^{2}GNN Wang et al. (2022)	49.94±1.47	54.99±3.11	58.87±1.18	65.09±3.02
TopoImb Zhao et al. (2022)	50.78±0.68	49.47±1.20	51.02±2.25	56.40±2.67
RAHNet Mao et al. (2023)	49.94±0.08	56.13±3.95	58.67±0.97	74.76±0.36
DataDec Zhang et al. (2023)	53.28±2.65	60.83±2.61	58.11±4.26	66.61±6.89
ImGKB Tang & Liang (2023)	50.19±3.07	60.22±4.56	50.06±0.31	50.26±0.22
GDeR Zhang et al. (2024)	51.36±2.60	58.60±1.85	62.20±2.47	64.59±2.10
C^{3}GNN Ju et al. (2025)	51.82±1.24	61.23±2.15	59.62±0.86	71.71±2.00
KDEX Mao et al. (2025)	56.26±2.42	67.04±5.93	64.47±4.94	77.45±1.78
SimGraph	**58.89±1.36**	**74.61±0.67**	**65.30±3.19**	**79.76±1.52**

vanilla backbone, yet CE + BPGS (without SAIN) already surpasses them by 4–8pp, and the full SimGraph extends the margin further. A key implementation-level observation: SupCon's softmax denominator sums over *all* non-self samples, including same-class samples outside the positive mask; under severe class imbalance ($\sim$34 labeled graphs), these act as *implicit false negatives* that pull same-class embeddings apart, whereas BPGS's per-pair sigmoid formulation avoids this denominator competition entirely.

G.4 Complementarity with Size-Invariant Regularization

SizeShiftReg Buffelli et al. (2022) enforces coarsening-invariant node-embedding distributions via central moment discrepancy (CMD); it does not model or correct the *pairwise similarity function* used for classification—two graphs can have CMD-invariant representations yet still suffer size-induced similarity bias. We therefore test SimGraph with an additional CMD regularizer ($\lambda_{\text{cmd}} = 0.1$). Table 27 shows the two mechanisms are complementary: adding CMD on top of SimGraph is roughly neutral on three datasets and mildly positive on PTC-MR, confirming that representation-level invariance and similarity-level correction address different levels of the bias.

G.5 Diagnostic Baselines: Semantic Gains vs. Size Cues

A central concern is whether SimGraph's gains reflect semantic improvement or exploitation of dataset-specific size cues. Table 28 reports diagnostic baselines under high size imbalance: size-only predictors (an MLP on $\log n$; $\log n$ plus a degree histogram), a structure-agnostic baseline (mean-

Table 20: **Macro-F1** (% ± stddev) on datasets with changing class imbalance levels over 10 runs, encoded by GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GCN (bb.) Kipf & Welling (2017)	32.82±3.29	62.02±2.84	59.32±2.48	67.58±1.06
G^{2}GNN Wang et al. (2022)	31.29±2.68	56.99±10.85	56.41±7.64	65.63±3.26
TopoImb Zhao et al. (2022)	45.01±8.36	55.05±4.87	33.04±0.43	50.17±12.11
RAHNet Mao et al. (2023)	55.74±1.51	71.78±3.52	69.63±0.19	79.24±0.67
DataDec Zhang et al. (2023)	53.24±3.23	61.35±2.55	54.26±7.60	67.37±6.20
ImGKB Tang & Liang (2023)	48.53±8.31	62.34±3.64	33.84±0.60	33.97±0.41
GDeR Zhang et al. (2024)	53.34±3.29	53.88±4.63	61.68±7.30	63.87±6.75
C^{3}GNN Ju et al. (2025)	49.11±4.37	65.87±1.06	69.25±1.45	79.49±0.37
KDEX Mao et al. (2025)	55.50±1.15	68.85±0.57	70.55±0.69	78.47±1.17
SimGraph	**59.02±1.70**	**74.18±2.56**	**70.97±0.96**	**79.63±1.31**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GCN (bb.) Kipf & Welling (2017)	31.49±5.35	58.06±1.46	36.70±6.23	57.56±6.17
G^{2}GNN Wang et al. (2022)	33.06±4.46	64.18±4.21	45.81±10.04	65.51±2.39
TopoImb Zhao et al. (2022)	39.66±9.73	40.50±7.73	37.79±5.88	45.50±5.89
RAHNet Mao et al. (2023)	45.93±1.06	70.64±1.90	62.12±4.62	79.24±0.67
DataDec Zhang et al. (2023)	**54.16±1.97**	61.94±1.91	50.18±11.79	64.62±6.82
ImGKB Tang & Liang (2023)	45.80±7.16	63.41±4.10	32.52±0.48	32.82±1.24
GDeR Zhang et al. (2024)	44.48±9.02	57.14±2.29	60.22±3.63	70.97±0.60
C^{3}GNN Ju et al. (2025)	44.07±4.41	69.99±1.63	64.92±1.78	79.41±0.34
KDEX Mao et al. (2025)	51.22±3.71	69.39±1.63	68.58±1.08	75.55±1.01
SimGraph	52.26±0.95	**74.43±0.78**	**70.27±0.48**	**81.37±0.53**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GCN (bb.) Kipf & Welling (2017)	29.74±6.39	40.59±3.18	31.21±0.36	47.46±8.81
G^{2}GNN Wang et al. (2022)	32.13±7.08	42.79±6.25	54.41±2.63	64.77±3.25
TopoImb Zhao et al. (2022)	32.03±3.31	31.98±1.84	33.79±6.25	44.46±6.18
RAHNet Mao et al. (2023)	27.71±0.22	47.26±8.79	54.20±3.75	74.24±0.72
DataDec Zhang et al. (2023)	52.48±3.00	60.89±2.60	53.87±8.76	62.46±11.54
ImGKB Tang & Liang (2023)	39.08±8.03	55.46±11.43	31.73±0.22	34.27±2.64
GDeR Zhang et al. (2024)	37.24±7.06	58.12±2.45	60.15±2.79	60.50±2.99
C^{3}GNN Ju et al. (2025)	31.99±2.93	56.38±5.87	54.25±1.15	71.55±2.38
KDEX Mao et al. (2025)	51.21±9.33	66.08±5.64	62.42±6.30	77.08±1.93
SimGraph	**56.33±1.46**	**74.10±1.17**	**63.98±4.03**	**79.07±1.27**

pooled node features + MLP, no message passing), and GIN with alternative readouts (attention pooling, Set2Set, size-normalized readout).

Three observations follow. (1) Size-only predictors approach chance on PTC-MR and IMDB-B, indicating that size alone is not a decisive cue there; on D&D and RDT-B size carries real signal, yet SimGraph still exceeds the size-only baselines by 14.6–17.0pp, showing it does not merely exploit size. (2) The structure-agnostic baseline trails SimGraph by 5–14pp, so message passing contributes substantially beyond node-feature statistics. (3) Alternative readouts (attention, Set2Set, size normalization) each recover only 1–3pp over vanilla GIN, and all remain 2.2–8pp below SimGraph. We note that variable-length mean pooling is itself a standard readout; our claim is narrower: under severe size imbalance, mean/sum pooling couples first- and second-order feature statistics with node count (Appendix I, $r(\|\mathbf{h}\|, \log n)$ up to $+0.9$), which distorts *norm-based* similarity. SAIN is complementary to all of these readouts—it normalizes size-linked statistics before the readout is applied, rather than replacing it.

Table 21: **AUC-ROC** (% ± stddev) on datasets with changing size imbalance levels over 10 runs, encoded by GCN. **Bold** and underlined highlight the best and second-best results respectively.

Algorithm	PTC-MR	D&D	IMDB-B	RDT-B
Balance ($\rho_{\text{class}} = 5:5$)				
GCN (bb.) Kipf & Welling (2017)	49.33±7.50	67.24±2.12	63.97±2.24	75.18±0.08
G^2GNN Wang et al. (2022)	45.43±5.01	68.38±3.80	61.49±6.19	74.01±3.96
TopoImb Zhao et al. (2022)	56.98±4.51	70.29±1.96	48.80±10.38	77.48±1.84
RAHNet Mao et al. (2023)	57.07±3.89	77.31±4.12	**78.25±1.29**	80.98±3.09
DataDec Zhang et al. (2023)	55.99±4.78	66.85±3.20	62.53±3.42	66.49±9.00
ImGKB Tang & Liang (2023)	53.61±5.87	67.82±1.50	51.83±1.56	72.98±5.95
GDeR Zhang et al. (2024)	56.73±2.19	70.28±1.74	67.97±7.59	75.71±3.68
C^3GNN Ju et al. (2025)	57.88±4.92	77.36±0.35	77.54±0.93	83.85±0.68
KDEX Mao et al. (2025)	58.67±2.52	75.57±0.91	78.05±1.04	81.71±2.24
SimGraph	**65.20±0.76**	**79.68±3.13**	77.27±2.44	**85.88±0.79**
Low Imbalance ($\rho_{\text{class}} = 7:3$)				
GCN (bb.) Kipf & Welling (2017)	45.84±4.13	65.26±2.20	62.50±1.26	75.19±1.08
G^2GNN Wang et al. (2022)	48.93±5.12	71.03±1.85	57.06±4.74	73.96±2.66
TopoImb Zhao et al. (2022)	55.07±4.49	72.19±3.37	50.00±10.14	77.58±0.54
RAHNet Mao et al. (2023)	50.85±1.00	69.93±11.55	73.24±0.32	78.88±1.27
DataDec Zhang et al. (2023)	**56.27±2.77**	68.13±2.14	64.08±3.48	65.24±11.34
ImGKB Tang & Liang (2023)	52.91±3.96	68.34±2.86	51.08±0.71	76.70±3.52
GDeR Zhang et al. (2024)	53.22±3.19	65.84±0.78	65.34±6.35	77.91±0.14
C^3GNN Ju et al. (2025)	49.13±1.07	65.60±8.26	76.02±0.75	77.21±4.67
KDEX Mao et al. (2025)	53.23±1.85	76.23±0.99	**77.07±0.50**	76.79±4.06
SimGraph	55.23±0.74	**80.53±0.91**	77.05±0.67	**85.28±3.07**
High Imbalance ($\rho_{\text{class}} = 9:1$)				
GCN (bb.) Kipf & Welling (2017)	44.65±4.74	64.67±1.98	56.60±8.91	76.80±2.42
G^2GNN Wang et al. (2022)	46.82±4.30	67.30±2.70	62.46±2.16	72.42±3.79
TopoImb Zhao et al. (2022)	53.31±5.12	57.78±6.66	52.42±10.28	77.07±0.24
RAHNet Mao et al. (2023)	48.29±2.45	55.19±6.33	60.60±5.44	77.15±1.25
DataDec Zhang et al. (2023)	55.65±3.63	67.68±2.86	63.17±4.96	64.92±15.31
ImGKB Tang & Liang (2023)	51.01±4.09	53.74±0.93	51.05±1.00	71.01±5.58
GDeR Zhang et al. (2024)	53.03±2.64	65.45±7.75	62.15±3.83	63.49±4.20
C^3GNN Ju et al. (2025)	51.31±2.75	65.32±5.81	**67.31±2.03**	74.18±2.65
KDEX Mao et al. (2025)	56.43±2.62	64.05±8.42	66.81±8.37	70.27±12.67
SimGraph	**60.21±1.51**	**79.39±0.69**	66.72±3.44	**84.01±1.53**

Table 22: Size-generalization methods adapted to the IGL-Bench protocol (GIN, Accuracy %, high size imbalance).

Method	PTC-MR	D&D	IMDB-B	RDT-B
Bevilacqua et al. Bevilacqua et al. (2021)	56.82±1.04	73.53±0.63	64.24±0.89	73.75±1.36
SizeShiftReg Buffelli et al. (2022)	57.29±1.46	73.28±1.12	65.82±0.83	74.93±1.32
SimGraph (ours)	**60.51±1.27**	**77.32±0.84**	**69.00±0.74**	**77.94±1.27**

Table 23: COLLAB (3-class) under size imbalance (GIN encoder, mean ± std over 10 runs, %). AUC-ROC is not reported as it is not defined for this 3-class protocol. **Bold** and underlined highlight the best and second-best results respectively.

Level	Method	Acc	bAcc	Macro-F1
Low (ρ_{size}=7.94)	GIN (bb.) Xu et al. (2019)	64.34±3.55	64.73±5.42	58.45±1.80
	SOLT-GNN Liu et al. (2022)	61.79±2.09	70.23±1.56	59.76±1.81
	TopoImb Zhao et al. (2022)	62.71±2.42	71.56±1.68	60.91±2.15
	SimGraph	**67.62±1.37**	**72.15±1.58**	**61.45±2.04**
Mid (ρ_{size}=11.66)	GIN (bb.) Xu et al. (2019)	65.84±3.12	74.40±1.72	63.79±2.55
	SOLT-GNN Liu et al. (2022)	**69.69±2.45**	75.86±1.12	66.96±1.86
	TopoImb Zhao et al. (2022)	68.68±1.34	76.54±0.54	66.58±1.09
	SimGraph	69.23±1.31	**77.21±1.73**	**68.37±1.84**
High (ρ_{size}=16.89)	GIN (bb.) Xu et al. (2019)	67.05±2.46	74.99±0.95	64.92±2.18
	SOLT-GNN Liu et al. (2022)	67.12±3.28	73.58±2.12	64.68±2.81
	TopoImb Zhao et al. (2022)	67.52±0.77	76.10±0.32	65.65±0.63
	SimGraph	**70.63±1.25**	**78.25±1.31**	**67.63±1.42**

Table 24: Results on DBLP (heterogeneous graph; GIN, mean ± std over 10 runs, %). **Bold** and underlined highlight the best and second-best results respectively.

Imb.	Level	Method	Acc	bAcc	Macro-F1	AUC-ROC
Class	Low	SOLT-GNN Liu et al. (2022)	78.30±0.74	77.08±0.89	76.75±0.91	87.05±1.50
		KDEX Mao et al. (2025)	78.90±0.98	78.83±0.86	78.15±0.81	89.28±1.91
		SIMBA Qin et al. (2025a)	79.97±0.65	80.11±0.47	79.25±0.59	91.29±1.88
		SimGraph	**81.07±0.71**	**80.68±0.68**	**80.47±0.75**	**92.73±0.58**
	Mid	SOLT-GNN Liu et al. (2022)	78.21±0.69	76.86±0.72	76.61±0.84	86.90±1.43
		KDEX Mao et al. (2025)	78.76±0.91	78.27±0.69	77.90±0.74	89.01±1.82
		SIMBA Qin et al. (2025a)	79.76±0.60	79.29±0.38	78.91±0.54	90.92±1.78
		SimGraph	**80.78±0.36**	**79.75±0.20**	**80.03±0.34**	**92.29±0.72**
	High	SOLT-GNN Liu et al. (2022)	73.00±1.13	70.51±1.25	71.14±1.12	86.45±1.21
		KDEX Mao et al. (2025)	73.91±1.50	72.98±1.21	72.85±0.99	88.24±1.54
		SIMBA Qin et al. (2025a)	75.55±0.99	74.77±0.66	74.20±0.72	89.85±1.51
		SimGraph	**77.23±1.63**	**75.57±1.66**	**75.69±1.75**	**91.01±1.12**
Size	Low	SOLT-GNN Liu et al. (2022)	75.69±1.52	75.60±1.33	75.54±1.32	86.95±1.45
		KDEX Mao et al. (2025)	76.91±2.02	78.21±1.28	77.57±1.17	89.10±1.85
		SIMBA Qin et al. (2025a)	79.13±1.33	80.10±0.70	79.16±0.85	91.04±1.81
		SimGraph	**81.39±0.18**	**80.95±0.17**	**80.93±0.16**	**92.43±0.37**
	Mid	SOLT-GNN Liu et al. (2022)	75.58±1.46	75.53±1.27	75.44±1.27	87.02±1.49
		KDEX Mao et al. (2025)	76.76±1.94	78.03±1.23	77.40±1.13	89.22±1.89
		SIMBA Qin et al. (2025a)	78.89±1.28	79.84±0.67	78.94±0.82	91.21±1.86
		SimGraph	**81.06±0.45**	**80.66±0.40**	**80.64±0.47**	**92.63±0.40**
	High	SOLT-GNN Liu et al. (2022)	75.86±1.62	75.42±1.18	75.35±1.23	86.96±1.46
		KDEX Mao et al. (2025)	77.16±2.14	77.74±1.14	77.23±1.08	89.12±1.86
		SIMBA Qin et al. (2025a)	79.51±1.42	79.42±0.62	78.71±0.79	91.07±1.82
		SimGraph	**80.91±1.09**	**80.18±1.01**	**80.35±1.14**	**92.47±0.51**

Table 25: Results on Freebase (heterogeneous graph; GIN, mean ± std over 10 runs, %). **Bold** and underlined highlight the best and second-best results respectively.

Imb.	Level	Method	Acc	bAcc	Macro-F1	AUC-ROC
Class	Low	SOLT-GNN Liu et al. (2022)	22.15±1.22	34.00±1.66	19.72±0.90	64.17±2.06
		KDEX Mao et al. (2025)	23.12±1.61	37.25±1.60	21.10±0.79	67.22±2.62
		SIMBA Qin et al. (2025a)	24.89±1.07	39.62±0.88	22.18±0.58	69.96±2.57
		SimGraph	**26.70±1.59**	**40.68±0.23**	**23.38±0.90**	**71.93±0.87**
	Mid	SOLT-GNN Liu et al. (2022)	22.67±1.51	34.12±1.76	19.80±0.94	63.52±1.73
		KDEX Mao et al. (2025)	23.89±2.00	37.59±1.70	21.24±0.83	66.08±2.21
		SIMBA Qin et al. (2025a)	26.09±1.33	40.11±0.93	22.38±0.60	68.40±2.16
		SimGraph	**28.34±2.89**	**41.24±0.95**	**23.63±1.20**	**70.06±3.13**
	High	SOLT-GNN Liu et al. (2022)	23.64±2.06	33.68±1.39	20.59±1.35	63.30±1.63
		KDEX Mao et al. (2025)	25.29±2.73	36.42±1.35	22.65±1.19	65.72±2.07
		SIMBA Qin et al. (2025a)	28.28±1.80	38.41±0.74	24.28±0.87	67.89±2.03
		SimGraph	**31.34±3.46**	**39.30±0.90**	**26.08±1.30**	**69.45±1.93**
Size	Low	SOLT-GNN Liu et al. (2022)	22.68±1.51	33.37±1.14	20.03±1.06	62.89±1.42
		KDEX Mao et al. (2025)	23.89±2.01	35.61±1.10	21.66±0.94	65.00±1.81
		SIMBA Qin et al. (2025a)	26.10±1.33	37.23±0.60	22.94±0.68	66.91±1.78
		SimGraph	**28.35±0.72**	**37.96±1.52**	**24.35±1.48**	**68.27±1.82**
	Mid	SOLT-GNN Liu et al. (2022)	22.60±1.47	33.01±0.84	19.71±0.89	61.85±0.91
		KDEX Mao et al. (2025)	23.79±1.95	34.67±0.81	21.07±0.79	63.21±1.16
		SIMBA Qin et al. (2025a)	25.93±1.29	35.87±0.45	22.15±0.57	64.43±1.14
		SimGraph	**28.12±1.88**	**36.41±1.52**	**23.34±0.88**	**65.30±0.60**
	High	SOLT-GNN Liu et al. (2022)	22.45±1.38	32.74±0.62	19.67±0.87	62.42±1.19
		KDEX Mao et al. (2025)	23.56±1.83	33.95±0.60	21.00±0.77	64.19±1.52
		SIMBA Qin et al. (2025a)	25.57±1.21	34.83±0.33	22.05±0.56	65.79±1.49
		SimGraph	**27.63±2.69**	**35.23±0.23**	**23.21±1.82**	**66.93±1.45**

Table 26: BPGS vs. label-aware contrastive losses as auxiliary similarity losses (GIN, size imbalance High, Accuracy %).

Method	PTC-MR	D&D	IMDB-B	RDT-B
CE only (vanilla GIN)	48.41±1.25	60.68±1.45	62.60±0.89	67.41±1.42
CE + SupCon	52.46±1.24	65.94±1.25	63.53±0.84	69.14±1.31
CE + ProGCL	53.02±1.42	66.73±1.63	65.21±0.92	71.26±1.26
CE + BPGS (w/o SAIN)	57.42±1.19	73.35±1.36	67.32±0.76	75.03±1.23
Full SimGraph	**60.51±1.28**	**77.32±1.17**	**69.00±0.95**	**77.94±1.06**

Table 27: Complementarity of SimGraph and CMD regularization (GIN, size imbalance High, Accuracy %).

Method	PTC-MR	D&D	IMDB-B	RDT-B
SizeShiftReg (GIN + CMD)	57.29±1.26	73.28±1.28	65.82±2.73	74.93±1.42
SimGraph (ours)	60.51±1.57	77.32±0.87	69.00±3.19	77.94±0.58
SimGraph + CMD	61.15±0.94	77.54±0.97	68.23±2.78	77.06±0.83

Table 28: Diagnostic baselines vs. SimGraph (GIN, size imbalance High, Accuracy %).

Method	PTC-MR	D&D	IMDB-B	RDT-B
Size-only (MLP on $\log n$)	48.15±1.06	60.43±0.94	62.36±1.23	66.52±0.94
Size-only ($\log n$ + degree hist.)	49.22±1.43	62.73±1.32	63.64±1.67	67.47±1.39
Structure-agnostic (mean-pool + MLP)	53.62±1.57	63.57±1.63	65.62±1.84	72.48±1.59
GIN + attention pooling	54.21±1.41	69.34±1.67	66.42±1.38	73.74±1.63
GIN + Set2Set	55.78±1.73	70.39±1.84	65.95±1.73	75.35±1.49
GIN + size-normalized readout	57.28±1.35	71.04±1.36	66.44±1.52	75.79±1.61
SimGraph (ours)	**60.51±1.57**	**77.32±0.87**	**69.00±3.19**	**77.94±0.58**

H Ablation Studies

In this section, we verify the effectiveness of each key component in our proposed SimGraph framework: Balanced Pairwise Graph Similarity (BPGS), Size-Adaptive Instance Normalization (SAIN), and Virtual Graph Augmentation (VGA). We conducted ablation studies on two representative datasets, D&D and RDT-B, covering both size imbalance and class imbalance scenarios (Mid and High levels). The encoder backbone is set to GCN for all variants.

We compare the full **SimGraph** model with three variants:

- **w/o BPGS**: The model trained without the BPGS loss, using standard Cross-Entropy loss instead. This isolates the contribution of our pairwise balanced supervision.
- **w/o SAIN**: The model with the SAIN module removed, using standard batch normalization or no normalization. This tests the importance of disentangling size statistics from semantic features.
- **w/o VGA**: The model trained without Virtual Graph Augmentation. This evaluates the benefit of generating diverse virtual samples for minority classes.

Formal Ablation Definitions. To clarify the exact training objectives for each variant, let $\mathbf{h}$ and $\mathbf{h}^{inv}$ denote graph embeddings before and after SAIN, respectively, and $\mathbf{h}^{virt}$ denote virtual graph embeddings. The training losses are:

$$\mathcal{L}_{\text{SimGraph}} = \mathcal{L}_{CE}(\mathbf{h}^{inv}, \mathbf{h}^{virt}) + \lambda_1 \mathcal{L}_{BPGS}(\mathbf{h} \times \mathbf{h},\ \mathbf{h} \times \mathbf{h}^{virt}) + \lambda_2 \mathcal{L}_{div}(\mathbf{h}^{virt}) + \lambda_3 \mathcal{L}_{orth}(\mathbf{h}^{virt}), \tag{44}$$

$$\mathcal{L}_{\text{w/o BPGS}} = \mathcal{L}_{CE}(\mathbf{h}^{inv}, \mathbf{h}^{virt}) + \lambda_2 \mathcal{L}_{div}(\mathbf{h}^{virt}) + \lambda_3 \mathcal{L}_{orth}(\mathbf{h}^{virt}), \tag{45}$$

$$\mathcal{L}_{\text{w/o SAIN}} = \mathcal{L}_{CE}(\mathbf{h}, \mathbf{h}^{virt}) + \lambda_1 \mathcal{L}_{BPGS}(\mathbf{h} \times \mathbf{h},\ \mathbf{h} \times \mathbf{h}^{virt}) + \lambda_2 \mathcal{L}_{div}(\mathbf{h}^{virt}) + \lambda_3 \mathcal{L}_{orth}(\mathbf{h}^{virt}), \tag{46}$$

$$\mathcal{L}_{\text{w/o VGA}} = \mathcal{L}_{CE}(\mathbf{h}^{inv}) + \lambda_1 \mathcal{L}_{BPGS}(\mathbf{h} \times \mathbf{h}), \tag{47}$$

where λ_i are balanced by GradNorm. BPGS, SAIN, and VGA target size bias at three different levels of the learning pipeline: SAIN at the *representation level* (decoupling size statistics from features), BPGS at the *embedding geometry level* (shaping pairwise structure), and VGA at the *data distribution level* (filling size gaps in minority classes). As a result, each component's impact is proportional to how severely its target level is stressed by the imbalance type: SAIN matters most under size imbalance, VGA matters most under class imbalance, and BPGS consistently dominates across all settings.

The quantitative results are reported in Table 29.

Effectiveness of BPGS. As observed in Table 29, removing BPGS (*w/o BPGS*) generally leads to a significant performance degradation across almost all settings. For instance, on the RDT-B dataset under High Size Imbalance, the accuracy drops from 79.63% to 72.84%. This sharp decline highlights the critical role of BPGS in handling severe imbalance. By explicitly defining positive pairs and reweighting them, BPGS effectively eliminates the false negatives inherent in standard contrastive learning and prevents the majority class from dominating the gradients.

Effectiveness of SAIN. The ablation results confirm that SAIN is essential for robust classification under size imbalance. In the D&D dataset, removing SAIN (*w/o SAIN*) results in a clear performance drop in Size Imbalance settings (e.g., from 76.11% to 73.50% accuracy in the High setting). This validates our theoretical motivation that graph size acts as a "style" variable that biases similarity estimation. SAIN successfully aligns the feature distributions of graphs with varying sizes, ensuring that the learned representations are semantically discriminative rather than size-dependent.

Effectiveness of VGA. Virtual Graph Augmentation generally contributes to improved generalization, particularly in scenarios with limited structural diversity. On the RDT-B dataset, removing VGA (*w/o VGA*) consistently lowers performance across most metrics. For example, in the Class

Table 29: Ablation study results on D&D and RDT-B datasets under different imbalance settings. We report Accuracy, Macro-F1, and AUC-ROC (mean ± std). **Bold** and underlined highlight the best and second-best results respectively.

Dataset	Split	Model	Accuracy	Macro-F1	AUC-ROC
D&D	Size Imb Mid	SimGraph	**77.14±0.22**	**75.34±0.46**	78.66±1.04
		w/o BPGS	73.14±4.62	72.15±4.09	71.40±2.03
		w/o SAIN	74.52±0.09	73.67±0.26	**81.34±1.08**
		w/o VGA	74.70±3.29	73.09±2.34	77.00±2.20
	Size Imb High	SimGraph	**76.11±2.30**	**74.15±2.30**	**80.04±2.17**
		w/o BPGS	69.89±3.31	68.55±2.38	75.63±1.99
		w/o SAIN	73.50±0.95	72.77±1.08	79.78±2.06
		w/o VGA	68.58±1.73	67.92±1.56	74.07±1.36
	Class Imb Low	SimGraph	**76.86±0.66**	74.43±0.78	80.53±0.91
		w/o BPGS	75.10±2.08	73.42±1.52	80.07±2.52
		w/o SAIN	76.47±0.49	**75.17±0.35**	**81.28±0.20**
		w/o VGA	73.86±1.76	72.69±1.29	79.40±0.63
	Class Imb High	SimGraph	76.26±1.63	74.10±1.17	79.39±0.69
		w/o BPGS	74.32±2.84	71.39±3.37	75.05±3.11
		w/o SAIN	74.14±1.63	72.27±1.09	79.80±0.55
		w/o VGA	**77.64±0.35**	**74.93±0.68**	**79.86±0.65**
RDT-B	Size Imb Mid	SimGraph	**82.08±1.44**	**82.05±1.48**	87.07±1.63
		w/o BPGS	76.06±0.27	76.04±0.26	83.49±0.54
		w/o SAIN	77.42±1.05	77.39±1.06	82.52±2.18
		w/o VGA	81.81±0.33	81.76±0.30	**89.16±0.99**
	Size Imb High	SimGraph	**79.63±0.97**	**79.54±1.01**	**86.04±1.67**
		w/o BPGS	72.84±0.68	72.71±0.64	80.24±1.03
		w/o SAIN	79.08±2.31	78.85±2.43	85.63±1.39
		w/o VGA	77.44±1.31	77.17±1.38	84.43±1.65
	Class Imb Low	SimGraph	**81.42±0.55**	**81.37±0.53**	85.28±3.07
		w/o BPGS	76.61±1.16	76.24±1.25	83.45±0.22
		w/o SAIN	77.61±2.12	77.38±2.34	83.85±1.34
		w/o VGA	80.02±1.63	79.97±1.66	**85.85±1.37**
	Class Imb High	SimGraph	**79.13±1.22**	**79.07±1.27**	**84.01±1.53**
		w/o BPGS	77.17±1.98	77.08±1.95	82.93±0.96
		w/o SAIN	76.17±3.30	75.95±3.42	82.04±4.87
		w/o VGA	76.37±2.60	76.22±2.60	80.36±2.75

Imbalance High setting on RDT-B, the full SimGraph outperforms the variant without VGA by over 2.7% in accuracy. While there are isolated cases where the simplified model performs competitively (e.g., D&D Class Imbalance High), VGA proves to be a stable and effective strategy for enriching the minority class space and preventing overfitting in the majority of imbalanced scenarios.

We note that the single reversal on D&D Class Imbalance High (w/o VGA at 77.64±0.35 versus full SimGraph at 76.26±1.63) is not statistically significant: a paired Wilcoxon signed-rank test over the 10 runs yields $p = 0.22$, and the 1.38pp gap lies within the combined standard deviation ($\sqrt{1.63^2 + 0.35^2} \approx 1.67$); it is the only setting among the eight class-imbalance configurations where the ablated variant edges ahead, whereas VGA yields significant gains elsewhere (e.g., +2.7pp on RDT-B Class Imbalance High).

In conclusion, the three components, BPGS, SAIN, and VGA, are synergistic, with each component dominating the imbalance type that stresses its target level: BPGS at the learning-signal level, SAIN at the representation level, and VGA at the data-distribution level. SimGraph achieves the best overall performance by integrating these modules to simultaneously address learning signal bias, representation bias, and data scarcity.

H.1 Ablation on Learnable Decision Boundary Bias b

To validate the contribution of the learnable bias b, we compare the full SimGraph (with b as an `nn.Parameter` initialized at -10) against a variant where b is fixed at 0 throughout training. Results are reported in Table 30.

Table 30: Ablation study on learnable bias b: learnable (default) vs. fixed at 0. Results on D&D and RDT-B across imbalance types and levels with GIN encoder.

Dataset	Imb. Type	Level	Acc	bAcc	Macro-F1	AUC-ROC
D&D	Class	Balanced (learnable)	75.52±0.80	74.51±0.52	74.41±0.61	81.08±0.54
		Balanced (fixed b=0)	73.30±2.58	72.32±2.07	72.17±2.37	77.85±1.76
		Low Imb. (learnable)	75.88±0.31	74.75±0.15	74.40±0.19	80.51±0.21
		Low Imb. (fixed b=0)	72.10±2.33	72.62±1.31	71.18±1.95	78.47±1.41
		High Imb. (learnable)	77.04±0.74	74.43±0.50	74.48±0.65	79.25±1.65
		High Imb. (fixed b=0)	76.62±0.62	73.72±0.65	73.84±0.34	77.99±1.79
	Size	Low Imb. (learnable)	74.99±2.29	76.96±1.07	74.77±2.05	81.73±1.51
		Low Imb. (fixed b=0)	70.81±2.62	74.73±2.02	70.76±2.64	86.48±0.98
		Mid Imb. (learnable)	76.43±1.17	74.57±1.35	74.88±1.28	80.24±1.03
		Mid Imb. (fixed b=0)	69.25±2.62	67.78±0.70	67.50±1.22	72.03±2.28
		High Imb. (learnable)	77.32±0.87	74.92±0.48	75.46±0.56	82.02±0.78
		High Imb. (fixed b=0)	73.00±5.01	70.48±2.65	70.68±3.65	78.57±5.33
RDT-B	Class	Balanced (learnable)	80.88±1.90	80.88±1.90	80.87±1.90	83.29±3.76
		Balanced (fixed b=0)	76.79±1.22	76.79±1.22	76.74±1.16	78.65±1.10
		Low Imb. (learnable)	80.27±1.08	80.51±1.11	80.26±1.09	86.58±0.80
		Low Imb. (fixed b=0)	78.25±0.62	78.08±0.58	78.12±0.59	82.02±1.48
		High Imb. (learnable)	81.67±1.82	82.13±1.54	81.62±1.80	85.77±2.04
		High Imb. (fixed b=0)	79.27±0.24	79.76±0.25	79.25±0.23	78.91±5.87
	Size	Low Imb. (learnable)	73.09±2.30	73.08±2.30	71.85±2.72	73.67±13.99
		Low Imb. (fixed b=0)	69.52±3.37	69.52±3.37	68.33±4.03	78.02±3.36
		Mid Imb. (learnable)	80.21±0.33	80.21±0.33	80.17±0.36	81.76±3.55
		Mid Imb. (fixed b=0)	76.65±0.93	76.65±0.93	76.53±0.94	78.67±1.94
		High Imb. (learnable)	77.94±0.58	77.94±0.58	77.88±0.60	74.02±8.81
		High Imb. (fixed b=0)	77.31±1.46	77.31±1.46	77.23±1.56	79.96±0.30

The learnable b consistently outperforms the fixed variant across almost all settings, with particularly large improvements on D&D Size Imb Mid (+7.18% Acc) and RDT-B Class Imb Low (+4.09% Acc). This validates that the adaptive decision boundary learned by b is essential for accommodating diverse graph size distributions, as the optimal threshold varies significantly across datasets and imbalance conditions.

H.2 Similarity Function Ablation

We compare our generalized similarity function $\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) = \|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\| \cdot (\cos\theta_{ij} - b_\theta)$ against cosine similarity and standard inner product on D&D and RDT-B under size imbalance (Table 31).

Our generalized similarity consistently outperforms both alternatives, with the advantage being most pronounced under high size imbalance on RDT-B (+7.00% Acc over cosine, +4.62% over inner product). This validates the importance of the angular bias b_θ and the norm-product term in compensating for size-induced similarity distortion.

I In-Depth Evaluation: SAIN's Interaction with Graph Size

SAIN Modulates Rather Than Removes Size Information. A key design principle of SAIN is that it *modulates* rather than suppresses size information. The instance normalization step removes uncontrolled size-dependent statistics, while the adaptive affine transformation ($\boldsymbol{\gamma}_G, \boldsymbol{\beta}_G$) re-injects size information in a task-optimized manner. Empirically, the Pearson correlation between $\|\mathbf{h}^{inv}\|$ and $\log n$ remains positive, confirming that SAIN preserves genuinely predictive size signals while removing the uncontrolled coupling that causes size-induced similarity bias.

A central question raised: *when should graph size be treated as a nuisance factor versus a genuinely predictive signal, and how does SAIN behave in each regime?* This appendix provides a systematic empirical investigation comprising four complementary analyses: (1) Pearson correlation analysis of

Table 31: Ablation on similarity functions under size imbalance (Accuracy / Bacc / Macro-F1 / AUC-ROC).

Dataset	Level	Similarity	Acc	Bacc	Macro-F1	AUC-ROC
D&D	Low	Ours (default)	74.10±0.24	75.86±0.28	73.92±0.35	77.90±0.36
		Cosine	64.54±0.63	69.79±1.27	64.25±1.70	84.94±2.54
		Inner product	69.32±0.50	73.87±1.02	69.24±1.57	77.32±2.40
	Mid	Ours (default)	77.60±0.52	76.33±0.55	76.42±1.02	80.57±1.87
		Cosine	75.90±0.25	72.62±1.27	73.36±1.46	80.33±2.22
		Inner product	76.11±0.84	73.47±0.92	74.06±1.52	79.15±2.35
	High	Ours (default)	78.13±0.57	74.64±0.77	75.59±1.28	82.34±2.07
		Cosine	73.99±0.39	73.31±0.59	73.02±0.79	79.66±1.50
		Inner product	78.03±0.93	75.23±1.14	75.98±1.61	83.08±2.50
RDT-B	Low	Ours (default)	76.06±0.93	76.06±0.93	75.12±2.57	80.93±2.88
		Cosine	72.44±2.36	72.44±2.36	72.06±2.66	66.71±2.89
		Inner product	73.81±0.36	73.81±0.36	73.43±0.68	76.39±1.58
	Mid	Ours (default)	74.12±0.44	74.12±0.44	74.01±0.96	78.46±1.62
		Cosine	73.44±2.44	73.44±2.44	73.40±2.48	71.81±2.98
		Inner product	74.06±0.59	74.06±0.59	73.16±1.08	77.02±1.99
	High	Ours (default)	78.00±1.51	78.00±1.51	77.95±1.83	79.62±2.74
		Cosine	71.00±1.76	71.00±1.76	70.94±1.72	72.44±2.78
		Inner product	73.38±1.96	73.37±1.93	72.86±2.19	63.42±2.84

embedding norms and graph sizes before and after SAIN (§I.1); (2) visualization of SAIN's learned affine parameters $\gamma(n), \beta(n)$ (§I.2); (3) a three-panel embedding geometry decomposition that goes beyond t-SNE (§I.3); and (4) a controlled boundary experiment characterizing SAIN's applicable regime (§I.4). Together, these analyses demonstrate that SAIN is an *adaptive* mechanism—it acts as a debiaser when size is confounding and as a regularizer when size is semantically meaningful—and we provide practical guidelines for practitioners.

I.1 Embedding Norm–Size Correlation Analysis

If SAIN's instance normalization step simply erased all size information, the norm $\|h^{inv}\|$ of the post-SAIN embedding would be uncorrelated with graph size. Conversely, if the learnable affine transform $\gamma(n), \beta(n)$ successfully re-injects task-relevant size information, a controlled residual correlation would remain. We test this by computing the Pearson correlation coefficient $r(\|\cdot\|, \log(n+1))$ on test-set embeddings under three conditions: (i) without SAIN (raw $\|h\|$), (ii) with SAIN but measuring the backbone output before SAIN ($\|h\|$), and (iii) with SAIN measuring the post-SAIN output ($\|h^{inv}\|$).

Table 32: Pearson correlation between embedding norm and $\log(n+1)$ on test sets across datasets and size imbalance levels (GIN encoder).

Dataset	Size Imb.	w/o SAIN: $r(\|h\|)$	w/ SAIN: $r(\|h\|)$	w/ SAIN: $r(\|h^{inv}\|)$
D&D	High	-0.161	$+0.049$	$+0.271$
	Mid	$+0.549$	$+0.545$	-0.204
	Low	$+0.774$	$+0.758$	$+0.357$
RDT-B	High	$+0.899$	$+0.738$	$+0.751$
	Mid	$+0.533$	$+0.517$	$+0.389$
	Low	$+0.777$	$+0.700$	$+0.186$

Three findings emerge from Table 32:

(1) Consistent size–norm decoupling. In 5 out of 6 settings, the absolute correlation $|r(\|h^{inv}\|, \log n)|$ is lower than $|r(\|h\|, \log n)|$ without SAIN, confirming that SAIN reduces uncontrolled size–norm coupling. On RDT-B Low, this reduction is dramatic: from $+0.777$ to $+0.186$ (76% reduction).

(2) Adaptive intervention strength. On RDT-B, SAIN removes 76% of the size–norm correlation under Low imbalance ($0.777 \rightarrow 0.186$) but only 16% under High imbalance ($0.899 \rightarrow 0.751$). This is consistent with the learned affine parameters (see §I.2): under High imbalance, where size is strongly correlated with class labels, SAIN deliberately *preserves* size information rather than removing it.

(3) Controlled re-injection, not erasure. On D&D High, the backbone itself becomes more size-agnostic when trained with SAIN ($|r|$ drops from 0.161 to 0.049), while the SAIN output exhibits a moderate positive correlation ($r = +0.271$). This replaces the original negative, uncontrolled coupling with a positive, task-optimized signal—demonstrating SAIN's "normalize-then-modulate" design.

Implication for the similarity function. These results directly address the concern that the norm product $\|h_i^{inv}\| \cdot \|h_j^{inv}\|$ in the similarity function (Eq. 8) might re-introduce size bias. Since $\|h^{inv}\|$ exhibits substantially reduced and controlled size dependence compared to raw $\|h\|$, the residual norm product reflects *task-optimized* size information rather than uncontrolled statistical artifacts.

I.2 Visualization of Learned SAIN Parameters

To understand *how* SAIN modulates size information, we visualize the learned affine parameters $\|\gamma_G\|$ and $\|\beta_G\|$ as functions of $\log(n+1)$ for test-set graphs, along with their Pearson correlations.

Table 33: Pearson correlation of SAIN's learned affine parameters with $\log(n+1)$.

Dataset	Size Imb.	$r(\|\gamma_G\|, \log n)$	$r(\|\beta_G\|, \log n)$
D&D	High	$+0.9997$	$+1.0000$
RDT-B	High	$+0.9986$	$+0.9999$
	Mid	-0.9859	$+1.0000$
	Low	-0.2507	$+0.3857$

The RDT-B progression from Low to High reveals a striking *self-adaptive* behavior: as size imbalance increases, SAIN's affine parameters become progressively more size-dependent ($|r|$ increases from ~ 0.3 to > 0.99). Under Low imbalance, γ_G and β_G capture complex, non-size-monotonic feature modulation patterns; under High imbalance, they converge to a systematic, monotonic size correction. This adaptivity is driven entirely by task-supervised gradients, with no manual mode-switching.

I.3 Embedding Geometry Decomposition

To provide more rigorous analysis than t-SNE (which conflates norm and direction), we design a three-panel visualization scheme that decomposes the embedding geometry into interpretable, quantitative axes:

- **Panel A (Norm–Angle polar plot):** Each graph is plotted in polar coordinates with $r = \|h\|$ and θ = angle projected onto the LDA discriminant direction. This reveals whether class separation arises from *direction* (angular—desirable) or *norm* (radial—size-confounded).
- **Panel B (Stereographic sphere projection):** Embeddings are L2-normalized, reduced to S^2 via PCA, and projected via stereographic projection. Since all norm information is removed, this shows *purely directional* class structure.
- **Panel C (Norm vs. size scatter):** $\|h\|$ vs. $\log(n+1)$ with per-class OLS regression lines and Pearson r values, quantifying the norm–size coupling per class.

We apply this analysis on D&D and RDT-B under all three size imbalance levels, comparing matched models with and without SAIN. Table 34 summarizes the quantitative results; selected visualizations are shown in Figures 6–8.

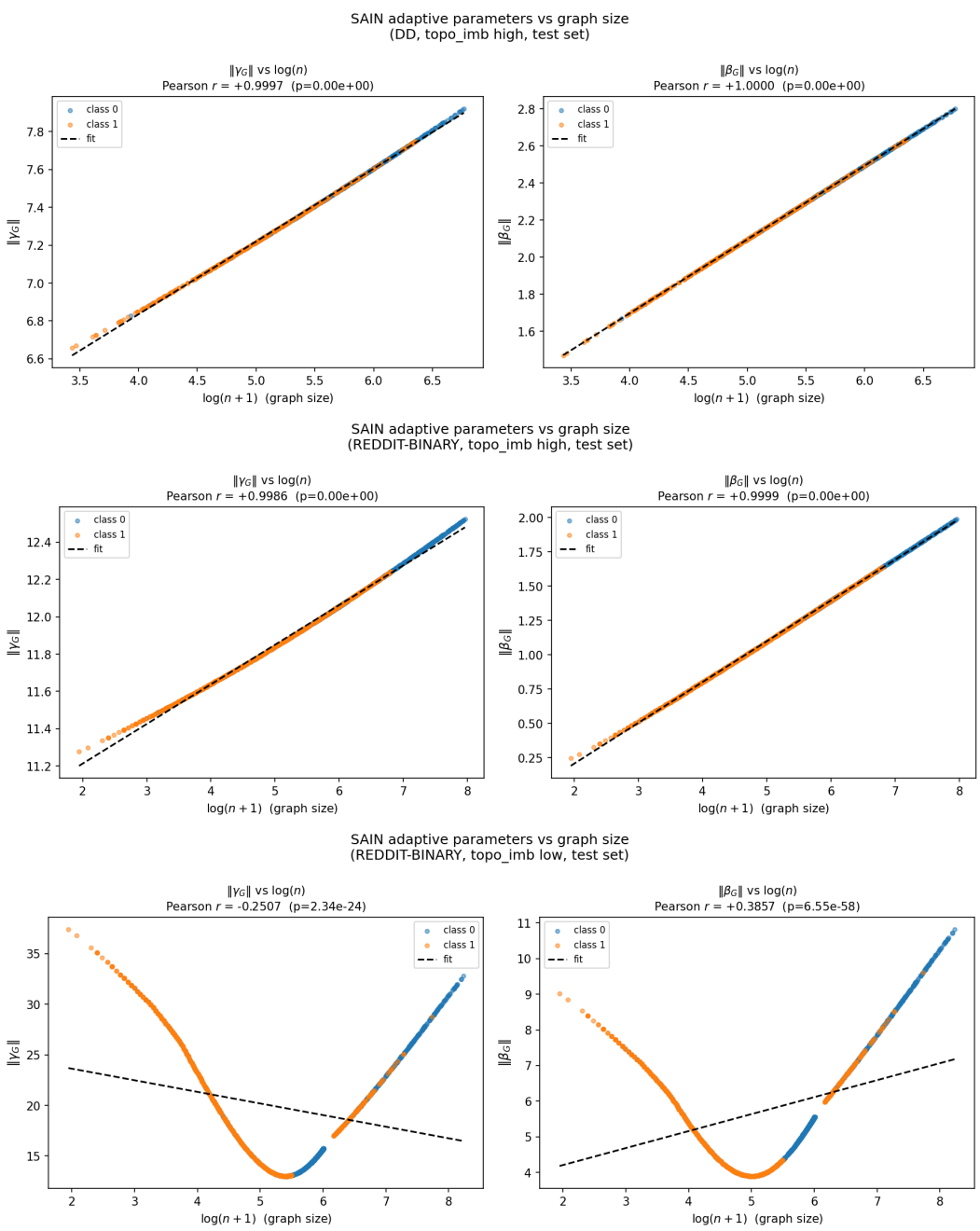


Figure 5: SAIN's learned $\|\gamma_G\|$ and $\|\beta_G\|$ vs. $\log(n{+}1)$. **Left:** D&D High—near-perfect linear dependence ($r > 0.999$), confirming systematic size-conditioned modulation. **Center:** RDT-B High—similarly strong linear dependence. **Right:** RDT-B Low—weak, non-monotonic relationship ($r_\gamma = -0.25$, $r_\beta = +0.39$), indicating SAIN *reduces* its size-dependent intervention when imbalance is mild.

Cross-dataset comparison. The contrast between D&D and RDT-B reveals two qualitatively different adaptation modes of SAIN, both emerging from the same architectural mechanism:

- **D&D (size is semantically meaningful):** SAIN's $\gamma(n), \beta(n)$ learn to *preserve and regularize* the size signal. At High imbalance, per-class correlations are amplified and symmetrized

Table 34: Consolidated summary of the three-panel embedding analysis.

Dataset	Imb.	Accuracy w/o	Accuracy w/	$r(\|h\|, \log n)$ w/o	$r(\|h\|, \log n)$ w/	Per-class r w/o	Per-class r w/	SAIN's role
D&D	Low	58.3	**77.1**	$+0.445$	$+0.719$	asym. (+0.58/+0.13)	sym. (+0.65/+0.71)	Regularizer
	Mid	75.3	**78.8**	$+0.705$	$+0.488$	both strong	both attenuated	Attenuator
	High	77.9	**78.0**	$+0.719$	$+0.889$	asym. (+0.83/+0.52)	sym. (+0.86/+0.84)	Regularizer
RDT-B	Low	44.3	**74.4**	$+0.316$	-0.220	both positive	divergent (-0.74)	Debiaser
	Mid	74.4	**75.7**	$+0.289$	$+0.080$	asym.	cls 1 sign reversal	Debiaser
	High	71.2	**77.7**	$+0.444$	-0.191	both positive	both flipped neg.	Debiaser

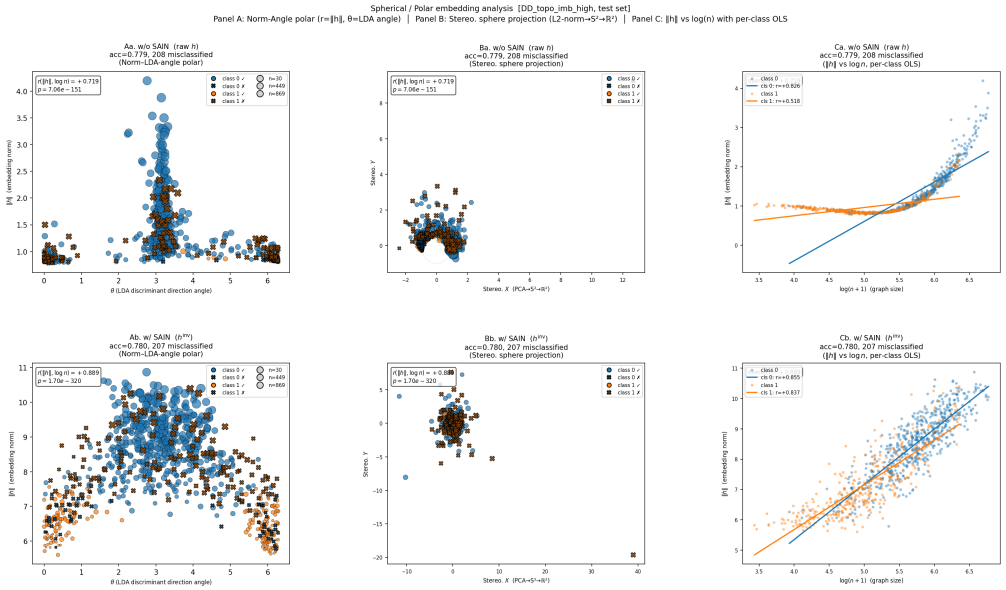


Figure 6: Three-panel embedding analysis on D&D under High size imbalance. **Top row:** without SAIN. **Bottom row:** with SAIN. Panel A shows that SAIN shifts the norm range upward while maintaining angular class separation. Panel B confirms that directional class structure is preserved. Panel C reveals that SAIN *symmetrizes* per-class correlations (from $+0.826/+0.518$ to $+0.855/+0.837$), acting as a regularizer that stabilizes the genuinely predictive size signal.

(from asymmetric $+0.826/+0.518$ to symmetric $+0.855/+0.837$), producing a more robust exploitation of size as a predictor.

- **RDT-B (size is primarily a confound):** SAIN's $\gamma(n), \beta(n)$ learn to *suppress or invert* the size–norm coupling. At High imbalance, both per-class correlations flip from positive to negative, and Panel B consistently shows improved *directional* class separation—confirming that SAIN restructures the embedding geometry rather than merely compressing norms.

This dual behavior—regularizer when size is predictive, debiaser when size is confounding—is driven entirely by task-supervised gradients, with no manual configuration.

I.4 Boundary Experiment: When Is SAIN Beneficial?

To directly characterize SAIN's applicable regime, we design a controlled experiment using three programmatic data splits that independently manipulate two measurable properties of the training distribution:

- $\hat{\Delta}_{OD} = \mathrm{KL}_{\mathrm{sym}}(P_{\mathrm{size}}^{\mathrm{train}} \| P_{\mathrm{size}}^{\mathrm{test}})$: the symmetric KL divergence between training and test size distributions, quantifying the degree of size-based distribution shift. This quantity requires **no labels** and can be computed in any low-label supervised or transductive setting.

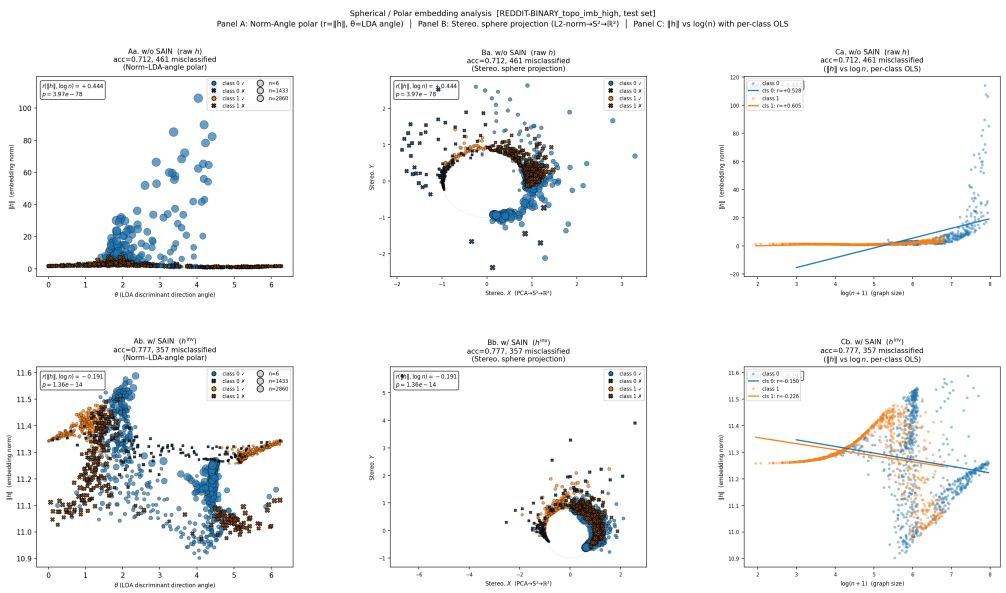


Figure 7: Three-panel embedding analysis on RDT-B under High size imbalance. **Top row:** without SAIN—Panel A shows large radial spread (0–100) with heavy angular class overlap; Panel B shows weak directional separation; Panel C shows both classes with strong positive norm–size coupling ($+0.528/+0.605$). **Bottom row:** with SAIN—Panel A shows dramatic radial compression ($\|h^{inv}\| \approx 10.9$–11.6) with clear angular class separation; Panel B shows markedly improved directional clusters; Panel C shows both correlations flipped to negative ($-0.150/-0.226$). SAIN acts as a *debiaser*, redirecting discriminative information from the norm axis to the angular axis.

- $\hat{\Delta}_{CS}^{\text{train}} = \text{KL}_{\text{sym}}(P_{\text{size}|y=0}^{\text{train}} \| P_{\text{size}|y=1}^{\text{train}})$: the symmetric KL divergence between per-class size distributions within the training set, quantifying how strongly size discriminates between classes in the current split.

Table 35: Three programmatic split strategies and their intended properties.

Split	Construction	$\hat{\Delta}_{OD}$	$\hat{\Delta}_{CS}^{\text{train}}$	Scenario
`size_ood`	Each class: smallest graphs → train	Large	≈ 0	Size is OOD nuisance
`size_mix`	Stratified by size within each class	Small	Small	Neutral baseline
`size_sep`	Class 0 = smallest, Class 1 = largest → train	Variable	Large	Size is training-set discriminator

For each of 4 datasets × 3 splits, we train matched models with and without SAIN using identical hyperparameters, and report balanced accuracy (BACC) and Pearson correlation $r(\|h^{inv}\|, \log n)$. Results are in Table 36.

Key findings. (1) SAIN is consistently beneficial under `size_mix` (balanced, non-discriminative size). Across all four datasets, ΔBACC is non-negative under `size_mix`, with gains of $+0.07$ to $+4.09$ pp. In this regime, size is a pure nuisance factor, and SAIN's decorrelation provides a clean benefit.

(2) SAIN prevents shortcut learning under `size_sep`. Under `size_sep`, the no-SAIN model overfits the spurious training-set size–class correlation and fails on the i.i.d. test set. SAIN's decorrelation prevents this shortcut, yielding large gains on RDT-B ($+13.66$ pp) and IMDB-B ($+8.57$ pp). This demonstrates an important robustness benefit: SAIN protects against size-based spurious correlations even when size appears discriminative in the training set.

(3) SAIN can be harmful when size is simultaneously OOD-shifted *and* genuinely discriminative. D&D `size_ood` ($\hat{\Delta}_{CS}^{train} = 10.90$, $r = 0.996$) shows -10.62 pp. The near-perfect norm–size correlation confirms that size is an extremely strong signal on D&D; removing it under distribution shift hurts. RDT-B `size_ood` shows a milder -3.10 pp for similar reasons.

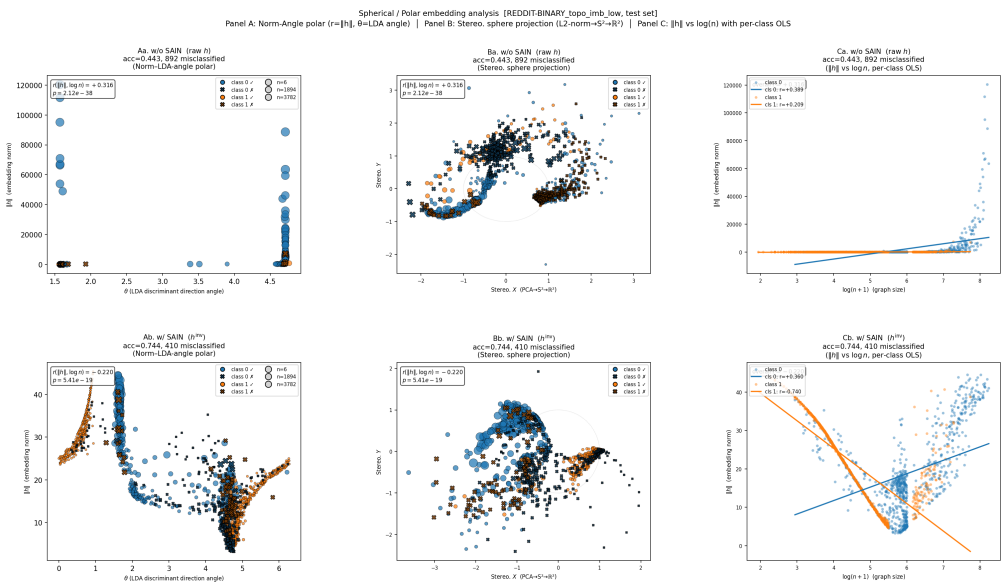


Figure 8: Three-panel embedding analysis on RDT-B under Low size imbalance. Without SAIN (**top**), embedding norms span up to 120,000 and the model collapses to near-chance accuracy (44.3%). With SAIN (**bottom**), the norm range is compressed by $\sim$3 orders of magnitude, angular class separation emerges in Panel B, and accuracy recovers to 74.4% (+30.1 pp). This demonstrates that size-induced embedding distortion is driven by the dataset's inherent size heterogeneity ($n \in [6, 3782]$), not merely by the imbalance ratio.

Table 36: Boundary experiment: SAIN's effect under controlled size distribution manipulations. ΔBACC = SAIN BACC $-$ no-SAIN BACC. Positive values indicate SAIN is beneficial.

Dataset	Split	$\hat{\Delta}_{OD}$	$\hat{\Delta}_{CS}^{train}$	SAIN BACC	no-SAIN BACC	ΔBACC	$r(\|\|h\|\|, n)$ no-SAIN	$r(\|\|h^{inv}\|\|, n)$ SAIN
IMDB-B	`size_ood`	13.93	0.42	**72.15**	71.02	+1.13	0.31	0.09
	`size_mix`	0.22	0.28	73.66	73.59	+0.07	0.49	0.31
	`size_sep`	5.07	17.07	**57.18**	48.61	+8.57	0.82	0.64
D&D	`size_ood`	3.65	10.90	61.55	**72.17**	−10.62	0.996	0.465
	`size_mix`	0.04	5.66	**74.27**	70.18	+4.09	0.86	0.43
	`size_sep`	0.34	13.37	**51.42**	48.43	+2.99	0.90	−0.05
MUTAG	`size_ood`	8.14	19.89	75.51	**76.35**	−0.84	0.74	0.11
	`size_mix`	0.67	11.72	**81.83**	80.21	+1.62	0.38	0.03
	`size_sep`	13.46	18.49	**82.15**	77.25	+4.90	−0.04	0.34
RDT-B	`size_ood`	4.73	9.71	72.76	**75.86**	−3.10	0.80	0.12
	`size_mix`	0.17	1.44	**78.05**	74.76	+3.29	0.69	0.20
	`size_sep`	3.82	3.21	**82.13**	68.47	+13.66	0.67	0.06

(4) MUTAG exhibits structural-feature dominance. Despite high intrinsic $\hat{\Delta}_{CS}$ (≈ 9.7 globally), MUTAG embeddings show very low norm–size correlation even without SAIN ($r = 0.04$ under `random`), indicating that structural features (e.g., ring systems, functional groups) dominate over size. SAIN is neutral-to-positive across all splits.

Table 37: Summary: SAIN's effect as a function of size distribution properties.

Condition	$\hat{\Delta}_{OD}$	$\hat{\Delta}_{CS}^{train}$	SAIN effect
Balanced, not discriminative	Small	Small	**Beneficial** (mild)
Size is OOD nuisance only	Large	Small	**Beneficial**
Size is training-set shortcut	Variable	Large	**Beneficial** (prevents shortcut)
Size OOD *and* genuinely semantic	Large	Large	**Potentially harmful**

Summary of SAIN's applicable regime.

Connection to main experiments. The size/class imbalance splits produced by IGL-Bench used in the main experiments (Tables Tables 1 and 2) create systematic size imbalance where the labeled set is dominated by graphs of particular sizes, corresponding to moderate-to-large $\hat{\Delta}_{OD}$ and small-to-moderate $\hat{\Delta}_{CS}^{train}$. According to our boundary analysis (Table 37), this is precisely the regime in which SAIN provides consistent benefit—explaining the consistent accuracy improvements observed in the main results.

Practical guideline. Compute $\hat{\Delta}_{OD} = \mathrm{KL}_{\mathrm{sym}}(P_{\mathrm{size}}^{\mathrm{train}} \| P_{\mathrm{size}}^{\mathrm{test}})$ (requires **no labels**). If $\hat{\Delta}_{OD} > 2.0$, SAIN is likely beneficial unless domain knowledge indicates that size is a primary semantic feature (e.g., polymer chain length). When uncertain, retain SAIN as a regularization layer: its learnable $\gamma(n), \beta(n)$ will adaptively calibrate the degree of size modulation via task-supervised gradients.

J Hyperparameter Sensitivity Studies

In this section, we examine the sensitivity of our proposed SimGraph framework to two critical hyperparameters in the Balanced Pairwise Graph Similarity (BPGS) module: the balance weight α and the asymmetric scaling factor r. Understanding the robustness of these hyperparameters is essential for demonstrating the practical applicability of our method across diverse imbalanced scenarios.

J.1 Balance Weight α

The balance weight $\alpha \in [0, 1]$ controls the relative importance of positive pairs versus negative pairs in the BPGS loss (Section 3.1). Since valid positive pairs (same-class labeled graphs) are inherently rarer than negative pairs in imbalanced settings, setting $\alpha > 0.5$ assigns higher importance to scarce positive pairs, directly addressing class imbalance at the pair level. We evaluate the sensitivity of α across 14 values: $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ on the D&D dataset under size imbalance scenarios.

Results. Figure 9 shows the performance of head graphs (large graphs) and tail graphs (small graphs) as α varies. We observe several key findings:

- **Effectiveness of $\alpha > 0$:** Compared to $\alpha = 0$ (no positive pair weighting), increasing α significantly improves tail graph performance, validating our design motivation that emphasizing positive pairs is crucial for handling imbalanced scenarios.
- **Robust plateau region:** Most importantly, the model exhibits remarkable stability across a wide range of $\alpha \in [0.5, 0.9]$ (highlighted as the plateau region). Within this range, tail graph accuracy remains consistently high ($\sim$68-70%), demonstrating that our method does not require meticulous hyperparameter tuning.
- **Pareto improvement:** As α increases, tail graph performance improves substantially without compromising head graph accuracy. The head graph performance remains stable (around 85-88%) across most α values, indicating that our balance weight mechanism achieves genuine performance gains rather than merely shifting the prediction bias.

These results confirm that the balance weight α is both *effective* (demonstrable improvement over baseline) and *robust* (stable performance across a broad parameter range), making SimGraph practical for real-world applications without extensive hyperparameter search.

J.2 Asymmetric Scaling Factor r

The asymmetric scaling factor $r > 0$ controls the gradient sharpness in hard pair mining, with reversed directions for positive and negative pairs (Section 3.1). For positive pairs, $1/r$ smooths gradients to prevent overfitting to scarce same-class pairs; for negative pairs, r sharpens gradients to enforce robust decision boundaries. This asymmetric design is critical because symmetric scaling would be compensated by magnitude adjustment and lose its discriminative power. We evaluate r across 5 values: $\{1/3, 1/2, 1, 2, 3\}$ on the D&D dataset under class imbalance scenarios.

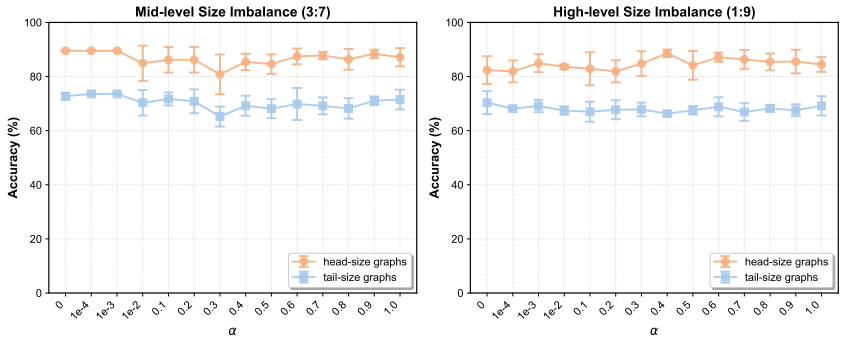


Figure 9: **Sensitivity analysis of balance weight α on D&D dataset.** The plot shows accuracy on head (large) and tail (small) graphs across 14 α values under Mid-level (left) and High-level (right) size imbalance. The shaded region highlights the robust plateau where performance remains optimal and stable. Error bars represent standard deviations over 10 independent runs.

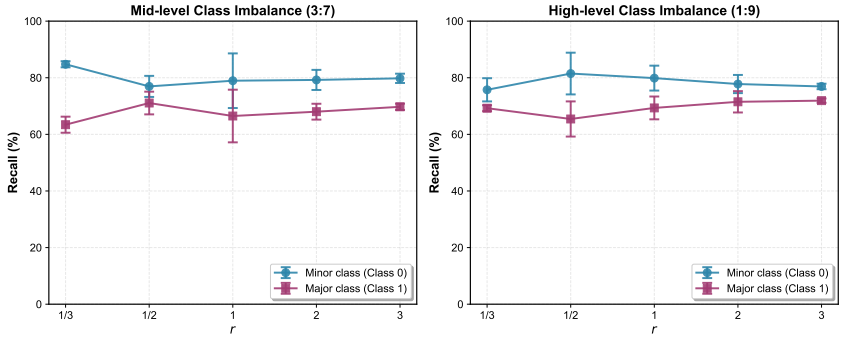


Figure 10: **Sensitivity analysis of asymmetric scaling factor r on D&D dataset.** The plot shows recall for minor class (Class 0) and major class (Class 1) across 5 r values under Mid-level (left) and High-level (right) class imbalance. The asymmetric design ($r \neq 1$) demonstrates clear benefits over symmetric scaling ($r = 1$). Error bars represent standard deviations over 10 independent runs.

Results. Figure 10 presents the recall performance for minor class (Class 0, less frequent) and major class (Class 1, more frequent) as r varies. Key observations include:

- **Necessity of asymmetric scaling ($r \neq 1$):** The symmetric baseline ($r = 1$) does not achieve optimal performance for either class. In the Mid-level setting, $r = 1/3$ significantly boosts minor class recall (84.76%) while maintaining reasonable major class recall (63.41%), demonstrating that asymmetric scaling is essential for addressing class imbalance.
- **Balanced trade-offs:** As r increases from $1/3$ to 3, minor class recall gradually decreases while major class recall improves, offering a spectrum of trade-offs. Notably, at $r = 2$ and $r = 3$, both classes maintain relatively balanced performance (around 70-79% for minor class, 68-71% for major class in Mid-level).
- **Stable performance region:** In the High-level imbalance setting (right panel), the model shows remarkable stability across $r \in [1/2, 3]$, with minor class recall remaining around 76-81% and major class recall around 65-71%. This demonstrates that r provides effective control over class balance without requiring precise tuning.

These findings validate our asymmetric scaling design: it provides meaningful control over the hard pair mining process, enables flexible trade-offs between classes, and maintains stable performance across reasonable parameter ranges.

J.3 Summary and Practical Guidelines

Our sensitivity studies demonstrate that both α and r exhibit desirable properties for practical deployment:

- **For balance weight α:** We recommend setting $\alpha \in [0.5, 0.9]$ for most imbalanced scenarios. Within this range, the model achieves robust performance with minimal sensitivity to exact values. A safe default choice is $\alpha = 0.5$ (equal weighting of positive and negative pairs).
- **For asymmetric scaling factor r:** We recommend $r \in [1/3, 2]$ depending on the desired balance between minor and major class performance. For scenarios prioritizing minor class recall, use $r \leq 1$; for more balanced performance, use $r \in [1, 2]$. The key insight is that $r = 1$ (symmetric) should be avoided as it fails to leverage the benefits of asymmetric gradient control.

Importantly, both hyperparameters demonstrate wide stable regions ("plateau" behavior) rather than sharp peaks, confirming that SimGraph does not require extensive hyperparameter tuning and is robust to parameter variations—a critical property for practical real-world applications.

J.4 Angular Bias b_θ

The angular bias b_θ is a hyperparameter selected from $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ via validation. Table 38 summarizes the best-performing b_θ and the baseline ($b_\theta = 0.0$, equivalent to standard inner product) across all settings on D&D and RDT-B with GIN encoder.

Table 38: Sensitivity of angular bias b_θ: best b_θ vs. b_θ=0 (standard inner product). Accuracy (%) reported.

Dataset	Setting	b_θ=0 (Acc)	Best b_θ (Acc)	Best b_θ
D&D	Class Balanced	70.83±2.60	73.23±0.79	0.4
	Low Class Imb.	75.10±1.43	76.01±1.10	0.3
	High Class Imb.	75.42±0.90	75.42±0.90	0.0
	Low Size Imb.	73.39±1.17	74.98±2.01	0.1
	Mid Size Imb.	69.22±2.13	73.81±1.89	0.4
	High Size Imb.	77.00±1.31	77.21±0.26	0.2
RDT-B	Class Balanced	73.42±0.90	77.04±1.11	0.5
	Low Class Imb.	76.21±2.15	78.00±0.16	0.3
	High Class Imb.	79.27±0.94	80.10±1.60	0.4
	Low Size Imb.	60.54±6.30	73.10±0.97	0.3
	Mid Size Imb.	73.85±1.55	76.69±0.98	0.3
	High Size Imb.	71.77±1.05	77.52±0.85	0.3

Setting $b_\theta > 0$ consistently improves over the standard inner product (b_θ=0), with particularly large gains on RDT-B Size Low (+12.56% Acc). The optimal value typically falls in $\{0.2, 0.3, 0.4\}$, with $b_\theta = 0.3$ being a robust default. Full per-setting results across all six b_θ values are provided in the supplementary materials.

J.5 Virtual Graph Budget M

We evaluate the sensitivity of the virtual graph bank budget $M \in \{20, 50, 100\}$ per class on D&D and RDT-B. Table 39 reports accuracy across imbalance settings with GIN encoder.

Performance is generally robust across $M \in \{20, 50, 100\}$, with M=20 being a practical default. Larger budgets do not consistently improve results and may introduce redundancy.

J.6 Momentum Coefficient η

We evaluate $\eta \in \{0.9, 0.95, 0.99, 0.999, 0.9999\}$ on D&D and RDT-B. Table 40 reports accuracy with GIN encoder.

Table 39: Sensitivity of virtual graph budget M on D&D and RDT-B (Accuracy %).

Dataset	Setting	M=20	M=50	M=100
D&D	Class Balanced	72.06±3.61	70.97±3.74	70.26±4.16
	Low Class Imb.	75.73±0.22	75.27±2.51	74.88±2.22
	High Class Imb.	75.56±0.57	75.06±0.92	74.92±1.87
	Low Size Imb.	73.22±0.50	71.94±1.42	71.59±2.50
	Mid Size Imb.	72.01±3.39	73.82±1.00	71.66±2.57
	High Size Imb.	74.74±4.37	76.75±1.17	75.09±3.57
RDT-B	Class Balanced	75.46±0.15	75.52±0.82	75.75±0.69
	Low Class Imb.	77.67±0.42	77.75±0.71	78.10±0.43
	High Class Imb.	81.19±1.21	78.96±2.84	81.19±0.74
	Low Size Imb.	72.17±3.58	67.79±2.07	74.12±2.46
	Mid Size Imb.	76.37±0.17	76.37±0.97	75.00±2.55
	High Size Imb.	77.88±0.78	76.44±1.45	75.56±3.37

Table 40: Sensitivity of momentum coefficient η (Accuracy %).

Dataset	Setting	η=0.9	η=0.95	η=0.99	η=0.999	η=0.9999
D&D	Class Balanced	75.55±1.82	74.64±1.82	72.87±2.99	74.04±2.05	70.93±3.84
	Low Class Imb.	75.45±1.14	76.01±0.49	74.11±2.35	75.45±1.41	74.88±3.06
	High Class Imb.	76.72±0.85	76.65±0.56	70.12±2.60	74.92±1.87	73.37±1.99
	Low Size Imb.	71.94±2.29	72.82±2.90	73.35±1.74	67.94±5.31	65.43±6.21
	Mid Size Imb.	71.02±0.83	69.53±1.59	69.21±2.11	72.40±3.98	70.88±3.42
	High Size Imb.	74.77±2.45	75.51±3.18	75.83±2.67	74.98±3.65	75.05±3.48
RDT-B	Class Balanced	75.79±0.52	76.08±0.36	75.96±0.84	76.15±0.33	76.21±1.36
	Low Class Imb.	77.19±0.82	79.38±0.26	77.89±0.59	77.33±0.19	78.13±0.45
	High Class Imb.	78.87±1.62	80.29±0.98	79.54±2.90	80.13±1.42	80.77±0.59
	Low Size Imb.	61.73±12.26	73.13±1.81	67.00±9.95	73.29±1.17	68.56±0.74
	Mid Size Imb.	75.58±0.74	74.46±1.45	75.02±2.50	76.38±1.04	74.96±2.84
	High Size Imb.	71.29±4.99	74.62±2.84	70.83±7.81	67.98±5.89	78.12±0.50

The model is generally robust across $\eta \in [0.9, 0.999]$, with η=0.999 being the default. On D&D, smaller η values (faster encoder updates) tend to work well for class imbalance, while larger η values provide more stable embeddings for the memory bank.

J.7 Memory Bank FIFO Queue Size Q

For large-scale deployment, the memory bank operates as a FIFO queue of configurable size Q. We evaluate $Q \in \{256, 384, 512, 640, 768, 896, 1024\}$ with batch size 128 on D&D, RDT-B, and ogbg-molhiv (80% labeled random split). Table 41 reports results.

Table 41: Effect of FIFO queue size Q on Accuracy (%) and resource usage (batch size = 128).

Q	D&D Acc	RDT-B Acc	molhiv Acc	Epoch Time (s)	CUDA Mem (MB)
256	76.19±2.41	85.17±2.25	78.12±2.14	0.27 / 0.49 / 5.17	1,534 / 7,246 / 3,084
512	77.59±3.24	83.17±1.31	84.52±1.95	0.23 / 0.49 / 5.14	1,540 / 7,258 / 3,308
768	73.95±1.81	85.33±1.18	87.41±1.52	0.26 / 0.48 / 5.13	1,548 / 7,270 / 3,568
1,024	73.67±2.41	84.50±0.41	81.31±2.45	0.23 / 0.48 / 5.11	1,548 / 7,268 / 3,858

Performance is robust across queue sizes, with training time remaining nearly constant. CUDA memory increases linearly but modestly. For our low-label setting with small labeled sets, caching all training batches (no FIFO needed) is the default.

K Large-Scale Evaluation Details: ogbg-molhiv and ogbg-molpcba

To evaluate scalability, we test on ogbg-molhiv Hu et al. (2020) (41,127 graphs) with 10% labeled imbalanced split and a queue size of 4,200 (caching all training batches). The full results are reported in Table 3 in the main text.

SimGraph outperforms SIMBA by 3.62% in accuracy, while requiring less runtime. Furthermore, limiting the size of the FIFO Queue reduces the epoch time, while the classification accuracy still surpasses all baselines. This confirms SimGraph's scalability to real-world large-scale graph datasets.

ogbg-molpcba. We further evaluate on ogbg-molpcba Hu et al. (2020) (437,929 graphs, 128 tasks) with the official split and evaluator, reporting test average precision (AP). As shown in Table 4 in the main text, SimGraph achieves 25.78% AP, surpassing the best baseline ImGKB by 1.60 points, confirming that the gains transfer to multi-task molecular property prediction.

L Computational Cost Analysis

We compare SimGraph's computational cost against all baselines across five datasets of increasing scale. All methods are trained for 500 epochs on a single NVIDIA A100 GPU.

Table 42: Training runtime (500 epochs, seconds) across datasets. OOM = Out of Memory.

Method	PTC-MR	D&D	IMDB-B	RDT-B	ogbg-molhiv
GIN Xu et al. (2019)	16.18	88.53	34	1540.40	1788.16
G^2GNN Wang et al. (2022)	38.35	354.31	103.19	1672.08	OOM
TopoImb Zhao et al. (2022)	36.03	159.88	65.32	601.91	OOM
SOLT-GNN Liu et al. (2022)	39.44	147.13	93.93	1997.77	OOM
RAHNet Mao et al. (2023)	47.29	149.81	97.93	267.83	23171.60
DataDec Zhang et al. (2023)	50.31	165.80	652.86	2020.56	OOM
ImGKB Tang & Liang (2023)	22.46	90.01	34.47	212.39	2512.86
GDeR Zhang et al. (2024)	23.36	76.14	38.80	143.75	2716.70
ImbGNN Xu et al. (2024)	35.62	329.27	98.16	1593.76	OOM
C^3GNN Ju et al. (2025)	37.87	126.25	83.48	225.82	57145.50
KDEX Mao et al. (2025)	95.58	271.26	196.52	474.67	45886.40
SIMBA Qin et al. (2025a)	47.17	150.54	97.45	244.71	9361.80
SimGraph	36.28	86.37	51.08	161.04	4216.50

Table 43: Preprocessing time (seconds) across datasets. OOM = Out of Memory.

Method	PTC-MR	D&D	IMDB-B	RDT-B	ogbg-molhiv
GIN Xu et al. (2019)	0.01	0.01	0.01	0.01	0.03
G^2GNN Wang et al. (2022)	1.76	872.93	6.95	5270.75	OOM
TopoImb Zhao et al. (2022)	6.65	216.91	13.98	456.72	OOM
SOLT-GNN Liu et al. (2022)	0.61	14.11	1.41	72.50	OOM
RAHNet Mao et al. (2023)	0.13	0.10	0.12	0.11	0.53
DataDec Zhang et al. (2023)	3.41	0.60	0.49	0.89	OOM
ImGKB Tang & Liang (2023)	0.47	4.56	0.89	20.43	21.13
GDeR Zhang et al. (2024)	0.08	0.26	0.12	0.33	0.51
ImbGNN Xu et al. (2024)	1.64	842.78	6.37	5014.84	OOM
C^3GNN Ju et al. (2025)	0.06	0.09	0.07	0.10	0.58
KDEX Mao et al. (2025)	0.11	0.11	0.12	0.10	0.53
SIMBA Qin et al. (2025a)	0.06	0.07	0.07	0.08	0.69
SimGraph	0.01	0.01	0.01	0.01	0.03

SimGraph requires **zero preprocessing** (no kernel computation or subgraph mining), achieves comparable or faster training than most baselines, and maintains the lowest main memory footprint. On the large-scale ogbg-molhiv dataset (41K graphs), SimGraph successfully trains while 6 out of 12 baselines encounter OOM errors.

Table 44: Peak main memory usage (MB) across datasets. OOM = Out of Memory.

Method	PTC-MR	D&D	IMDB-B	RDT-B	ogbg-molhiv
GIN Xu et al. (2019)	1,171	1,403	1,187	3,507	1,726
G^2GNN Wang et al. (2022)	1,419	3,986	1,460	15,435	OOM
TopoImb Zhao et al. (2022)	1,476	6,565	1,504	7,228	OOM
SOLT-GNN Liu et al. (2022)	1,446	2,979	1,513	15,965	OOM
RAHNet Mao et al. (2023)	1,580	1,573	1,581	2,632	1,878
DataDec Zhang et al. (2023)	18,740	19,252	18,820	23,133	OOM
ImGKB Tang & Liang (2023)	1,409	1,658	1,410	4,861	1,940
GDeR Zhang et al. (2024)	1,603	1,601	1,599	1,605	1,859
ImbGNN Xu et al. (2024)	1,395	3,701	1,285	14,926	OOM
C^3GNN Ju et al. (2025)	1,574	1,598	1,614	2,632	1,885
KDEX Mao et al. (2025)	1,423	1,417	1,421	1,426	1,754
SIMBA Qin et al. (2025a)	1,512	1,519	1,522	1,540	1,943
SimGraph	1,319	1,322	1,335	1,855	1,676

Table 45: Peak CUDA memory usage (MB) across datasets. OOM = Out of Memory.

Method	PTC-MR	D&D	IMDB-B	RDT-B	ogbg-molhiv
GIN Xu et al. (2019)	558	1,194	560	4,964	596
G^2GNN Wang et al. (2022)	580	3,636	646	13,388	OOM
TopoImb Zhao et al. (2022)	604	5,362	790	3,864	OOM
SOLT-GNN Liu et al. (2022)	576	2,254	580	2,416	OOM
RAHNet Mao et al. (2023)	544	936	568	3,640	656
DataDec Zhang et al. (2023)	570	1,664	692	3,712	OOM
ImGKB Tang & Liang (2023)	678	8,284	1,016	21,348	1,314
GDeR Zhang et al. (2024)	558	1,502	634	5,516	582
ImbGNN Xu et al. (2024)	575	3,593	634	13,293	OOM
C^3GNN Ju et al. (2025)	544	982	570	3,784	14,286
KDEX Mao et al. (2025)	546	1,102	572	3,790	808
SIMBA Qin et al. (2025a)	552	1,270	588	4,312	23,960
SimGraph	344	1,270	414	5,446	8,100

M Limitations and Broader Impacts

Limitations. Our framework treats graph size primarily as a nuisance factor, which is appropriate when size variation is driven by data collection artifacts or when labels are determined by local substructures (e.g., mutagenicity determined by functional groups). However, in domains where graph size directly encodes the property of interest (e.g., polymer chain length predicting material properties), SAIN's normalization could suppress genuinely predictive signals. As a practical guideline, we recommend computing the size-class correlation $\hat{\Delta}_{CS}^{train}$ on the training set: when this correlation is low and size variation is high, SAIN is beneficial; when size is a strong class-predictive feature with minimal distribution shift, SAIN can be disabled. SimGraph's modular design supports this flexibility.

Broader Impacts. SimGraph improves equity in graph classification by preventing models from systematically favoring structurally dominant (typically larger) graphs, which is particularly relevant in applications such as drug discovery (where small molecules may be underrepresented), social network analysis (where smaller communities may be marginalized), and protein function prediction. However, in high-stakes applications where graph size carries substantive domain-specific meaning, practitioners should carefully evaluate whether size normalization is appropriate before deployment. We do not foresee direct negative societal impacts from this work.