LEVERAGING STRUCTURE-ATTRIBUTE MIXED-VIEW AUGMENTATION FOR IMPROVED GRAPH CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Contrastive Learning (GCL) offers a promising self-supervised paradigm for learning node representations, but existing methods typically rely on heuristic augmentations that perturb topology or attributes in isolation, often introducing semantic drift that degrades representation quality. We identify semantic preservation as a key but overlooked property of graph augmentation, and propose SAMA (Structure-Attribute Mixed-View Augmentation), an augmentation framework that jointly models structural connections and node attributes. SAMA integrates deep non-negative matrix factorization with spectral feature filtering to generate multiple semantically aligned augmented views, and adopts a two-stage training scheme of semantic-aware augmentation followed by multi-view contrastive learning. Extensive experiments on six benchmark datasets demonstrate that SAMA consistently outperforms sixteen state-of-the-art baselines in node classification, achieving improvements of up to 3.0 percentage points while maintaining strong performance under extreme label scarcity. These results highlight semantic-preserving augmentation as a representation learning insight for self-supervised graph learning.

1 INTRODUCTION

Graph-structured data, such as social networks and knowledge graphs, are pervasive in modern data management systems. Graph Neural Networks (GNNs) have become the de facto standard for processing such relational data by iteratively aggregating and updating node representations among neighbors via message passing Kipf & Welling (2017); Hamilton et al. (2017); Yang et al. (2023). However, most GNN frameworks rely on supervised learning and require abundant labeled nodes, which is costly and often infeasible in domains such as biomedicine and complex social networks Shchur et al. (2018).

To mitigate dependence on manual labeling, Self-Supervised Graph Learning (SSGL) designs pretext tasks that automatically generate supervisory signals from graph structure, node/edge attributes, and statistical priors. Among these, Graph Contrastive Learning (GCL) has emerged as one of the most active directions due to its transferability and robustness You et al. (2020); Veličković et al. (2019). The core idea of GCL is to construct positive pairs (semantically similar) and negative pairs (semantically dissimilar) and apply a contrastive loss that learns an embedding space consistent across views while discriminative with respect to irrelevant factors. A typical GCL framework decomposes into three interdependent components, each of which we examine below.

(i) Graph data augmentation (GDA). GDA constructs diverse yet semantically consistent views of the original graph, injecting "perspective variance" into the encoder. Existing strategies fall into two categories: *rule-based* approaches (random edge deletion/addition, subgraph sampling, node/feature masking, noise injection) are simple and cheap, but can easily disrupt key relationships; *learnable* approaches (adversarial perturbation, reinforcement learning, gradient-guided augmentation) adapt the perturbation to downstream tasks and better trade off diversity against semantic preservation. The core challenge shared by all strategies is ensuring sufficient diversity to avoid an information bottleneck while maintaining latent semantic consistency, so as not to mislead the encoder into learning task-irrelevant or harmful representations You et al. (2020); Xiong et al. (2025).

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.

(ii) Contrastive objective definition. This component determines how positive and negative pairs are defined. A common practice treats the representations of the same node (or the same graph) across different augmented views as positive pairs, and representations of different nodes (or graphs) as negatives. Efficient and accurate negative selection remains challenging: explicit schemes (e.g., InfoNCE-based) maintain large negative pools that raise memory cost and may introduce false negatives; implicit or negative-free schemes use momentum encoders and symmetric prediction heads to reduce sampling overhead while alleviating convergence instability from semi-hard negatives Thakoor et al. (2022); Miao et al. (2022); Hassani & Khasahmadi (2020).

(iii) Contrastive loss. The loss converts positive and negative pairs into an optimizable objective; the predominant form is InfoNCE, which uses a temperature coefficient τ to control gradient scale, maximizing positive-pair similarity while minimizing the gap between positives and negatives Oord et al. (2018). Subsequent work has enriched loss design with mutual-information objectives, covariance regularization, and multi-granular constraints Veličković et al. (2019); Bai et al. (2021), yet balancing uniformity, separability, and semantic completeness remains an open issue.

The overlooked issue: semantic drift. A semantically consistent augmentation should enrich relational patterns without altering global meaning, whereas an over-aggressive perturbation may indiscriminately drop or rewire edges, yielding views whose semantics no longer correspond to the source graph and undermining the contrastive signal. Figure 1 illustrates this effect on a teacher–student–parent social graph: (a) the original graph preserves role semantics; (b) a semantically consistent evolution keeps new links within the same role clusters; (c) an over-aggressive perturbation causes teachers to appear isolated from students and parents. These observations lead to two design principles: **(P1) semantic consistency**—augmented views should remain within a bounded semantic distance from the original graph; and **(P2) joint structure–attribute modeling**—because semantics emerge from the interplay of topology and node features, perturbations must account for both aspects simultaneously.

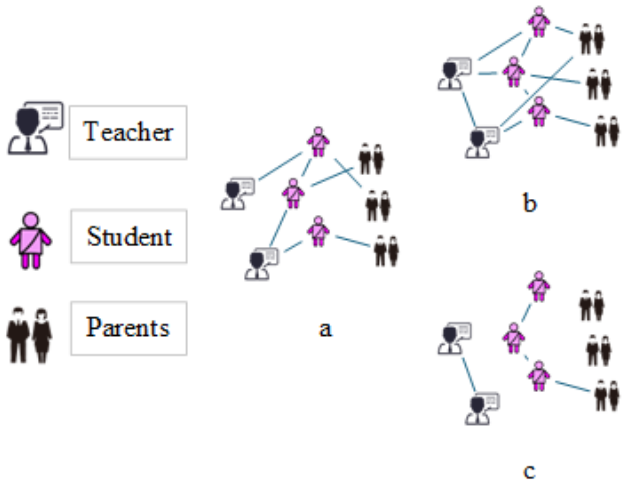


Figure 1: Teacher-student social network: (a) the original graph, (b) the network after a reasonable evolution with semantically similar connections, and (c) the network where the semantics are severely disrupted, resulting in dissimilar connections.

Our approach. We propose SAMA (Structure-Attribute Mixed-View Augmentation), an augmentation framework that operationalizes P1–P2. SAMA employs deep non-negative matrix factorization (DNMF) to learn low-rank latent factors that encode complementary structural and attribute signals, then reconstructs from these factors multiple times to generate anchor views that preserve semantics. A weight-shared GNN embeds each view, and a cross-view InfoNCE loss with a semantic-consistency regularizer drives the model to remain faithful to the source graph while distinguishing informative variants. Figure 2 summarizes the pipeline.

Contributions. (1) We identify semantic preservation as a key overlooked property of graph contrastive augmentation, and formalize it through two design principles. (2) We propose SAMA, a

structure-attribute joint augmentation framework integrating DNMF with spectral filtering to generate semantically consistent anchor views and a two-stage training scheme for multi-view contrastive learning. (3) On six benchmark datasets, SAMA consistently outperforms sixteen strong baselines, achieving improvements of up to 2.9 percentage points under extreme label scarcity.

2 Related Work

2.1 Graph Neural Networks and Contrastive Learning

Classical GNNs such as GCN Kipf & Welling (2017) and GraphSAGE Hamilton et al. (2017) perform message passing via neighborhood aggregation; GAT Veličković et al. (2017) introduces attention to weight neighbors adaptively, while JKNet Xu et al. (2018) and DropEdge Rong et al. (2019) alleviate over-smoothing in deep networks. Despite these advances, two structural issues persist: the *locality bottleneck*—message passing mixes features only within a k-hop neighborhood—and *over-smoothing*—representations of distant nodes converge. Remedies span architectural, optimization, and theoretical perspectives; injecting global structure into local operators remains an open challenge.

GCL learns representations by maximizing the similarity between augmented views. DGI Veličković et al. (2019) maximizes mutual information between global and local representations; GraphCL You et al. (2020) constructs views through node dropping and edge perturbation; GCA Zhu et al. (2021) adaptively selects augmentation strategies; JOAO You et al. (2021) automates strategy selection; HGCL Bai et al. (2021) extends to hypergraphs; GRACE Zhu et al. (2020) applies contrastive learning across randomly augmented views; ASP Chen & Kou (2023) jointly considers attribute and topological consistency; AFGRL Lee et al. (2022) eliminates handcrafted transformations; SUGRL Mo et al. (2022) uses random feature masking and walk-based sampling. Despite these advances, most methods treat topology and attributes as independent channels, leaving semantic consistency during augmentation underexplored.

2.2 Deep Non-Negative Matrix Factorization and Graph Data Management

NMF factorizes a non-negative matrix $\mathbf{V}$ into lower-dimensional non-negative factors $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{V} \approx \mathbf{WH}$, extracting interpretable latent features. DNMF Trigeorgis et al. (2014) extends this to a multi-layer decomposition that captures hierarchical features, and has been applied to image and attribute representation Trigeorgis et al. (2017), bioinformatics Jee et al. (2023), and graph-regularized community detection Ye et al. (2018). Our work leverages DNMF's non-negativity, which aligns with the adjacency matrix, to extract latent semantic representations that preserve graph structure.

Graph data management has been extensively studied in the database community, with research spanning storage, indexing, query processing, and analytics Besta et al. (2024); Coimbra et al. (2025). More recently, the community has recognized the potential of GNNs for enhancing data management systems Li et al. (2025); Khan et al. (2025). These learning-based approaches, however, face label scarcity, noisy attributes, and the semantic drift introduced by heuristic augmentation. Unlike prior methods that focus on system-level optimizations, SAMA operates at the data preparation layer, generating augmented views that retain latent semantics and providing a solution for enhancing downstream analytical quality.

3 Method

3.1 Preliminaries and Semantic Graph Augmentation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be a graph with nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. We define the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ with $\mathbf{D} = \mathrm{Diag}(d_1, \ldots, d_N)$, $d_i = \sum_j A_{ij}$, and its symmetric normalization $\mathbf{L}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ with eigenvectors $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ and eigenvalues $\mathbf{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_N)$. For a discrete graph signal $\mathbf{X}$, the graph Fourier transform is given by $\tilde{\mathbf{X}} = \mathbf{U}^\top\mathbf{X}$ Shuman et al. (2013).

Semantic graph augmentation comprises two components: (1) semantic extraction and (2) semantic-aware graph augmentation. We first decompose both the graph topology and node attributes via DNMF to extract semantics, then introduce a semantic consistency loss to constrain the reconstruction of the adjacency matrix, preventing it from deviating from the semantic structure of the original graph.

Why DNMF? We adopt DNMF as the semantic extractor for two reasons. First, the non-negativity constraint aligns with the adjacency matrix and node features in most real-world graphs (edge weights and feature values are non-negative), and enforces a parts-based, interpretable factorization in which each latent dimension corresponds to a coherent semantic cluster. Second, the multi-layer decomposition captures hierarchical structure that a single-layer NMF cannot represent, which is important for graphs whose semantics span multiple granularities. These properties make DNMF a suitable choice for extracting latent structure that can serve as a semantic anchor during augmentation.

Definition 1 (Latent semantic representations). Let $\mathbf{A} \in \mathbb{R}^{n\times n} \geq 0$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times d} \geq 0$ be the non-negative adjacency and (filtered) attribute matrices, with decomposition depth p. DNMF approximates them as

$$\min_{\mathbf{W}_i,\mathbf{T}_s} \|\mathbf{A} - \mathbf{T}_s\mathbf{W}_1\cdots\mathbf{W}_p\|_F^2, \quad \text{s.t. } \mathbf{T}_s, \mathbf{W}_i \geq 0, \tag{1}$$

$$\min_{\mathbf{Q}_i,\mathbf{F}_s} \|\tilde{\mathbf{X}} - \mathbf{Q}_1\cdots\mathbf{Q}_p\mathbf{F}_s\|_F^2, \quad \text{s.t. } \mathbf{F}_s, \mathbf{Q}_i \geq 0, \tag{2}$$

where $\tilde{\mathbf{X}} = \mathrm{Filter}(\mathbf{X}, \mathbf{A}) = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{X}$ applies spectral filtering to capture global structural signals. $\mathbf{T}_s \in \mathbb{R}^{n\times r_p}$ is the *latent structural semantic representation* and $\mathbf{F}_s \in \mathbb{R}^{r_p\times d}$ the *latent attribute semantic representation*; $\mathbf{W}_i$ and $\mathbf{Q}_i$ are non-negative transformation matrices at layer i. The non-negativity constraints ensure interpretability and sparsity, and align with the adjacency structure to preserve topological semantics.

Theorem 1 (Semantic preservation). Let $\mathbf{A} \in \mathbb{R}^{n\times n} \geq 0$ encode the structural information of a graph whose node set can be partitioned into M semantic clusters $C_1, \ldots, C_M$ such that A_{uv} is relatively large for $u, v \in C_m$. For any decomposition depth p, the learned $\mathbf{T}_s$ from (1) satisfies

$$\forall u, v \in C_m, \quad \|\mathbf{t}_{s,u} - \mathbf{t}_{s,v}\|_2 \ll \|\mathbf{t}_{s,u} - \mathbf{t}_{s,v'}\|_2, \quad \forall v' \notin C_m,$$

where $\mathbf{t}_{s,u}$ denotes the u-th row of $\mathbf{T}_s$. That is, nodes in the same semantic cluster map to similar latent vectors, while nodes in different clusters map farther apart. The proof is provided in Appendix A, which builds upon the equivalence between NMF and spectral clustering Ding et al. (2005); Kuang et al. (2012).

We convert the constrained problems in (1)–(2) to unconstrained ones via a penalty $p(\mathbf{X}) = \sum_i \log(1+\exp(-X_i))^2$, yielding the semantic extraction loss

$$\min_{\mathbf{W}_i,\mathbf{T}_s,\mathbf{F}_s,\mathbf{Q}_i} \mathcal{L}_{\mathrm{Sel}} = T + \lambda p(\{\mathbf{W}_i\}, \mathbf{T}_s) + F + \lambda p(\{\mathbf{Q}_i\}, \mathbf{F}_s), \tag{3}$$

where λ controls the penalty strength, and T and F denote the reconstruction terms in (1) and (2), respectively.

Semantic-aware graph augmentation. Starting from the latent representations, we iteratively refine the graph:

$$\mathbf{S}_i = \mathbf{T}_{s,i}\mathbf{T}_{s,i}^\top, \quad f(\xi) = \alpha\,\xi + (1-\alpha)\,\mathbf{A}_0, \tag{4}$$

$$\mathbf{A}_{i+1} = \frac{1}{2}\big(f(\mathbf{S}_i) + f(\mathbf{S}_i)^\mathsf{T}\big), \quad \mathbf{X}_{i+1} = \mathrm{Filter}(\mathbf{X}, \mathbf{A}_{i+1}), \tag{5}$$

where $0 \leq \alpha \leq 1$. This update retains the latest similarity information while integrating the initial topology $\mathbf{A}_0$, preserving graph symmetry. The process forms a closed loop where $\mathbf{A}_{i+1}$ and $\mathbf{X}_{i+1}$ are reconstructed from the original graph. The complete procedure is summarized in Algorithm 1.

Consistency and regularization losses. To ensure semantic consistency, we penalize deviation from the source graph:

$$\mathcal{L}_{\mathrm{Sga}} = \sum_{i=0}^{K} \mathrm{tr}\big[(\mathbf{A}_0 - \mathbf{A}_i)^\top(\mathbf{A}_0 - \mathbf{A}_i)\big] + \sum_{i=0}^{K} \mathrm{tr}\big[(\mathbf{X}_0 - \mathbf{X}_i)^\top(\mathbf{X}_0 - \mathbf{X}_i)\big]. \tag{6}$$

To preserve intrinsic geometry and sparsity, we add a graph regularizer $\frac{1}{2}\sum_{i,j} A_{ij}\|\mathbf{t}_{s,i} - \mathbf{t}_{s,j}\|_2^2 = \mathrm{tr}(\mathbf{T}_s^\top \mathbf{L}\mathbf{T}_s)$ and an ℓ_1 penalty:

$$\mathcal{L}_{\text{reg}} = \sum_{i=1}^{K} \big(\|\mathbf{T}_{s,i}\|_1 + \|\mathbf{F}_{s,i}\|_1\big) + \sum_{i=1}^{K} \mathrm{tr}(\mathbf{T}_{s,i}^\top \mathbf{L}\mathbf{T}_{s,i}). \tag{7}$$

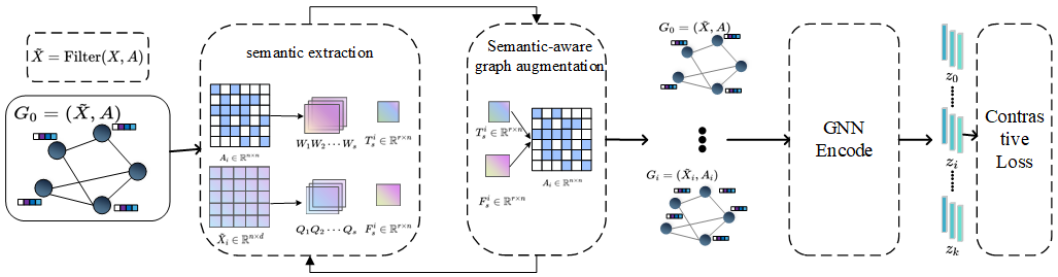


Figure 2: SAMA: multiple contrastive views are generated through iterative semantic extraction and semantic-aware graph augmentation, followed by multi-view contrastive learning.

Algorithm 1: SAMA: Structure–Attribute Mixed–View Augmentation for Graph Contrastive Learning

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, adjacency matrix $\mathbf{A}$, feature matrix $\mathbf{X}$, number of iterations K
Output: Enhanced adjacency matrices $\{\mathbf{A}_1, \ldots, \mathbf{A}_K\}$ and feature matrices $\{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$
Initialize $\mathbf{A}_0 = \mathbf{A}$, $\mathbf{X}_0 = \text{Filter}(\mathbf{X}, \mathbf{A})$, $\mathbf{T}_s, \mathbf{F}_s$ (random initialization);
for *iteration* $i = 1$ *to* K **do**
  Apply deep decomposition: $\mathbf{T}_s, \mathbf{F}_s \leftarrow \text{Decompose}(\mathbf{A}_{i-1}, \mathbf{X}_{i-1})$;
  Compute $\mathbf{S}_i \leftarrow \mathbf{T}_s \mathbf{T}_s^\top$;
  Reconstruct adjacency matrix: $\mathbf{A}_i \leftarrow \frac{1}{2}(f(\mathbf{S}_i) + f(\mathbf{S}_i)^\top)$, where $f(\xi) = \alpha\xi + (1-\alpha)\mathbf{A}_0$;
  Filter feature matrix: $\mathbf{X}_i \leftarrow \text{Filter}(\mathbf{X}, \mathbf{A}_i)$;
return Enhanced matrices $\{\mathbf{A}_1, \ldots, \mathbf{A}_K\}$ and $\{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$;

3.2 Multi-View Contrastive Learning and Optimization

We use the original adjacency $\mathbf{A}$ and the iteratively generated matrices $\mathbf{A}_i$ $(i = 1, \ldots, K)$ as contrastive views. Let $\mathbf{v}_j$ be node v_j's embedding in the original view and $\mathbf{h}_j^{[i]} = f(\mathbf{A}_i, \mathbf{X}, \mathbf{W})$ its embedding in the i-th augmented view.

Positive and negative pairs. For each node v_j, we treat its embeddings across the original view and all K augmented views as *positive pairs*—specifically, we pair the original-view embedding $\mathbf{v}_j$ with each augmented-view embedding $\mathbf{h}_j^{[i]}$. All other node embeddings $\{\mathbf{h}_k^{[i]} : k \neq j\}$ within the same augmented view serve as *negatives*. This view-consistent pairing encourages the encoder to produce embeddings that are invariant to semantic-preserving augmentation while remaining discriminative across nodes.

The contrastive loss is

$$\ell_j^{[i]} = -\log \frac{\exp(\theta(\mathbf{v}_j, \mathbf{h}_j^{[i]})/\tau)}{\exp(\theta(\mathbf{v}_j, \mathbf{h}_j^{[i]})/\tau) + \sum_{k \neq j} \exp(\theta(\mathbf{v}_j, \mathbf{h}_k^{[i]})/\tau)}, \tag{8}$$

where $\theta(\cdot) = s(f(\mathbf{V}), f(\mathbf{H}))$ uses cosine similarity s and temperature τ; the overall objective averages over all nodes and views:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{KN} \sum_{i=1}^{K} \sum_{j=1}^{N} \ell_j^{[i]}. \tag{9}$$

Two-stage optimization. Stage 1 optimizes the augmentation module:

$$\min_{\{\mathbf{W}_i\},\mathbf{T}_s,\{\mathbf{Q}_i\},\mathbf{F}_s} \mathcal{L}_{\text{SAMA}} = \frac{1}{3}(\mathcal{L}_{\text{Sel}} + \mathcal{L}_{\text{Sga}} + \mathcal{L}_{\text{reg}}). \tag{10}$$

Stage 2 optimizes the encoder with

$$\Phi_t = \Phi_{t-1} - \eta_t \nabla \mathcal{L}_{\text{contrastive}}(\Phi_{t-1}), \tag{11}$$

where η_t is the learning rate. The two stages are optimized separately: the augmentation module is first trained to convergence (we monitor the validation loss and stop when the relative change over two consecutive epochs is below 10^{-4}), and its outputs are then used as fixed inputs to drive the contrastive encoder. Stage 1 parameters are frozen during Stage 2.

4 Experiments

4.1 Experimental Settings

Datasets. Six real-world networks: Cora, CiteSeer, PubMed Sen et al. (2008), WikiCS Mernyei & Cangea (2020), Amazon-Photo, and Amazon-Computers Shchur et al. (2018). Table 1 summarizes their statistics.

Table 1: Dataset statistics.

Datasets	Nodes	Edges	Features	Labels
Cora	2708	10556	1433	7
CiteSeer	3327	9104	3703	6
PubMed	19717	88648	500	3
WikiCS	11701	431726	300	10
Amazon-Photo	7650	238162	745	8
Amazon-Computers	13752	491722	767	10

Baselines. We compare against sixteen methods. GCL methods: GCA Zhu et al. (2021), GraphCL You et al. (2020), DGI Veličković et al. (2019), GMI Peng et al. (2020), GRACE Zhu et al. (2020), ASP Chen & Kou (2023), AFGRL Lee et al. (2022), iGCL Li et al. (2024), IDEAL Sun et al. (2024), SUGRL Mo et al. (2022). Random-walk methods: DeepWalk Perozzi et al. (2014), node2vec Grover & Leskovec (2016), LINE Tang et al. (2015). Supervised GNNs: GCN Kipf & Welling (2017), GAT Veličković et al. (2017), AM-GCN Wang et al. (2020). Detailed descriptions of each baseline are provided in Appendix A.

Implementation details. Adam optimizer; experiments are conducted on a single NVIDIA GPU with 16GB memory. GNN encoder layers from $\{1, 2\}$; output dimension from $\{16, 32, 64, 128, 256, 512\}$; temperature τ from $\{0.1, \ldots, 1\}$; views K from $\{1, \ldots, 10\}$; learning rate grid-searched on $\{10^{-5}, \ldots, 5 \times 10^{-1}\}$. Default $\lambda = 300$, $\alpha = 0.5$, decomposition depth $p = 3$; r set to the number of classes. Per-dataset τ: Cora 0.4, CiteSeer 0.9, PubMed 0.7, WikiCS 0.5, Amazon-Photo 0.4, Amazon-Computers 0.3.

4.2 Evaluation Results

We learn node embeddings in an unsupervised manner and feed them into logistic regression. On Cora, CiteSeer, and PubMed, we use 20 nodes/class for training, 500 for validation, and the rest for testing; on Amazon-Photo and Computers, 20/class for training and 30/class for validation; on WikiCS, 20/class for training and 500 for validation. Each dataset uses 20 random splits; averages are reported.

Table 2 reports the results. SAMA achieves the best performance on all six datasets. On Cora, CiteSeer, and Photo, it reaches 82.5%, 73.4%, and 92.5%, respectively, significantly surpassing the strongest baselines. Notably, the improvement is most pronounced on WikiCS (75.5% vs. 72.6% of GCA) and Amazon-Computers (85.2% vs. 83.0% of ASP), where semantic preservation is critical.

Table 2: Node classification accuracy (%) on six datasets. Best in bold, second-best underlined.

Type	Algo.	Cora	CiteSeer	Photo	Computers	WikiCS	PubMed
RW	DeepWalk	71.3±1.6	45.5±1.5	59.7±2.1	61.3±1.8	61.5±1.4	68.3±2.4
	node2vec	75.7±0.3	51.3±1.4	73.8±0.8	63.0±0.8	61.1±0.8	69.1±1.0
	LINE	73.3±0.4	58.6±1.0	75.7±0.4	67.3±0.9	62.0±0.7	70.4±1.2
GCL	GCA	79.2±2.4	65.2±1.8	84.0±1.4	77.7±1.4	72.6±1.5	81.5±2.5
	GraphCL	78.0±1.7	69.2±1.1	86.2±1.7	77.8±1.4	66.5±1.4	78.0±1.3
	DGI	78.4±1.6	68.1±2.4	87.2±0.9	75.9±1.1	71.3±1.6	74.5±1.6
	GMI	79.4±1.4	67.1±2.4	89.2±0.9	74.9±1.1	70.3±1.6	76.5±1.6
	GRACE	81.2±1.6	69.6±1.8	88.6±1.4	79.8±1.3	67.2±1.8	79.5±1.1
	ASP	80.6±1.3	71.6±1.7	90.1±0.7	83.0±1.0	69.8±1.8	81.5±1.9
	AFGRL	81.3±1.0	69.8±1.8	90.5±0.7	81.9±1.6	67.2±0.9	79.6±1.1
	SUGRL	80.2±1.2	67.9±1.8	90.4±0.4	82.0±1.8	67.3±1.1	78.8±1.1
	iGCL	82.1±2.2	60.1±1.1	89.8±0.9	80.3±1.8	68.2±0.9	78.5±1.1
	IDEAL	82.0±1.2	70.6±1.7	91.5±0.7	81.0±1.8	69.2±0.9	80.6±1.1
	Ours	**82.5±1.0**	**73.4±1.0**	**92.5±0.7**	**85.2±0.9**	**75.5±1.5**	**82.5±2.2**
GNN	GCN	79.6±1.6	68.4±1.7	88.4±1.5	78.3±1.7	68.3±1.7	73.5±1.9
	GAT	81.2±1.6	68.9±1.8	86.5±1.5	77.7±1.4	68.4±1.7	73.5±1.9
	AM-GCN	80.0±1.2	70.2±1.5	88.7±1.4	78.3±1.6	69.5±1.5	76.6±2.3

Table 3: Ablation on Cora, WikiCS, PubMed (ACC %).

Method	Cora	WikiCS	PubMed
Ours $[\mathcal{L}_{\mathrm{Sel}}+\mathcal{L}_{\mathrm{Sga}}]$ (w/o $\mathcal{L}_{\mathrm{reg}}$)	75.5±1.0	60.5±1.5	65.5±1.5
Ours (full loss)	**82.5±1.0**	**75.5±1.5**	**82.5±2.2**
Structure-Preserving Only	80.5±1.0	74.5±1.5	80.5±1.5
Attribute-Preserving Only	55.5±1.0	62.5±1.5	54.5±1.5

4.3 Ablation and Parameter Analysis

We ablate the semantic augmentation loss $\mathcal{L}_{\mathrm{Sga}}$ and regularization $\mathcal{L}_{\mathrm{reg}}$, and further decompose the semantic augmentation into structure-only and attribute-only variants. Since $\mathcal{L}_{\mathrm{Sel}}$ is essential for uncovering latent semantics, we exclude it from the ablation.

Table 3 reveals three key findings. First, removing the regularization term $\mathcal{L}_{\mathrm{reg}}$ drops accuracy by 7.0/15.0/17.0 points on Cora/WikiCS/PubMed, confirming its importance for preserving intrinsic geometry and preventing degenerate solutions. Second, comparing the structure-preserving and attribute-preserving variants, we observe that attribute-only variants suffer the largest drops (e.g., 55.5% on Cora, a 27-point decrease), suggesting that structural semantics carry the dominant signal for these citation and co-purchase networks. Third, the full model consistently outperforms both single-channel variants, validating that semantic consistency emerges from the *joint* modeling of topology and attributes rather than from either channel alone.

Parameter analysis. Figure 3 shows accuracy as a function of the number of views K. Performance improves monotonically with K but saturates beyond $K = 5$, while computational cost grows linearly with K; we therefore set $K = 5$ by default. The balancing parameter $\lambda = 300$ effectively preserves non-negativity after several iterations. Detailed sensitivity results for λ and α are provided in Appendix B.

4.4 Scalability Discussion

While the experimental results demonstrate the effectiveness of SAMA in terms of accuracy, it is equally important to consider its practical applicability in real-world graph data management systems, where graph sizes can range from thousands to millions of nodes.

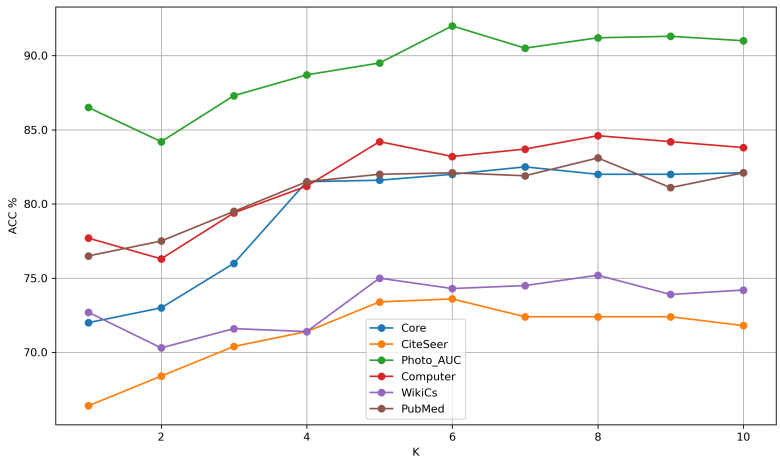


Figure 3: Node classification accuracy of SAMA as a function of the number of augmented views K.

SAMA's computational cost is primarily determined by two factors: (1) the DNMF decomposition performed on both the adjacency and attribute matrices, and (2) the GNN encoding of multiple augmented views. As analyzed in detail in the appendix, the overall time complexity of SAMA is $\mathcal{O}(K \cdot (p \cdot (nr^2 + r^3) + |E|d + nd))$, where K is the number of augmented views, p is the decomposition depth, r is the intermediate embedding dimension, and d is the feature dimension. This complexity grows linearly with the number of views K and remains practical for graphs with up to tens of thousands of nodes. For larger graphs, the dense factorization steps in DNMF may introduce computational bottlenecks, which we identify as a key direction for future optimization (see Section 5).

In practice, the intermediate dimension r is set to the number of classes (typically much smaller than n), which keeps the factorization tractable while providing sufficient capacity to capture the cluster structure; setting $r \gg$ number of classes would introduce redundant parameters without improving semantic preservation in our experiments. The decomposition depth p is fixed at 3 throughout, balancing the expressiveness of multi-layer factorization against the computational overhead. Furthermore, the modular design of SAMA allows easy substitution of DNMF with more efficient sparse or low-rank approximations, enabling adaptation to even larger graph data management scenarios.

5 Conclusion

We presented SAMA, a semantic-aware graph contrastive learning framework that performs mixed augmentation on both graph topology and node attributes while preserving latent semantic consistency. By combining DNMF with spectral feature filtering, SAMA constructs multiple structure–attribute mixed anchor views, enhancing the diversity of contrastive signals without compromising semantic consistency. A two-stage training scheme learns node representations that are discriminative, robust, and semantically faithful. Experiments on six benchmarks show that SAMA consistently outperforms sixteen strong baselines, achieving up to 2.9 percentage points improvement and maintaining strong generalization under extreme label scarcity. These results highlight semantic-preserving augmentation as a representation learning insight, and suggest structure–attribute joint modeling as a direction for self-supervised graph learning.

Limitations. SAMA reconstructs dense latent factors via DNMF, which may bottleneck on million-scale graphs; the current semantic consistency metric (Frobenius norm) may not capture task-specific semantics; and extending to dynamic and heterogeneous graphs is left for future work.

AI USAGE DISCLOSURE

During the preparation of this work, the authors used AI-assisted tools for language polishing. All mathematical formulations, experimental results, and citations were carefully reviewed and verified by the authors. The authors take full responsibility for the final content of this publication.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, we provide detailed descriptions of our proposed method, including all mathematical formulations, algorithm pseudocode, and implementation details in Sections 3 and 4. The source code and data preprocessing scripts will be made publicly available upon acceptance.

ETHICS STATEMENT

This work does not involve human subjects, sensitive data, or any potential ethical concerns. All datasets used in this work are publicly available benchmark datasets.

REFERENCES

Yunsheng Bai, Miao Ding, Yizhou Sun, Wei Wang, and Tianyi Lin. Hypergraph contrastive learning for data-efficient molecular property prediction. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4804–4815, 2021.

Maciej Besta, Robert Gerstenberger, Emanuel Peter, Marc Fischer, Michal Podstawski, Claude Barthels, Gustavo Alonso, and Torsten Hoefler. Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. *ACM Computing Surveys*, 56 (2):1–40, 2024.

Jialu Chen and Gang Kou. Attribute and structure preserving graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7024–7032, 2023.

Miguel E Coimbra, Lucie Svitáková, Alexandre P Francisco, and Luís Veiga. Survey: Graph databases. *arXiv preprint arXiv:2505.24758*, 2025.

Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 606–610, 2005.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1024–1034, 2017.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pp. 4116–4126, 2020.

Dong Jun Jee, Yixin Kong, and Hyonho Chun. Deep nonnegative matrix factorization using a variational autoencoder with application to single-cell rna sequencing data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(2):883–893, 2023. doi: 10.1109/TCBB.2022.3172723.

Arijit Khan, Xiangyu Ke, and Yinghui Wu. Graph data management and graph machine learning: Synergies and opportunities. *ACM SIGMOD Record*, 54(2):28–42, 2025.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2017.

Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 106–117, 2012.

Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7372–7380, 2022.

Haifeng Li, Jun Cao, Jiawei Zhu, Qinyao Luo, Silu He, and Xuying Wang. Augmentation-free graph contrastive learning of invariant-discriminative representations. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):11157–11167, 2024.

Ziming Li, Youhuan Li, Yuyu Luo, Guoliang Li, and Chuxu Zhang. Graph neural networks for databases: A survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 10555–10563, 2025.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

Rui Miao, Yintao Yang, Yao Ma, Xin Juan, Haotian Xue, Jiliang Tang, Ying Wang, and Xin Wang. Negative samples selecting strategy for graph contrastive learning. *Information Sciences*, 613: 667–681, 2022.

Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7797–7805, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *The Web Conference*, pp. 259–270, 2020.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Qingqiang Sun, Kai Wang, Wenjie Zhang, Peng Cheng, and Xuemin Lin. Interdependence-adaptive mutual information maximization for graph contrastive learning. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8556–8567, 2024.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations*, 2022.

George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W Schuller. A deep semi-nmf model for learning hidden representations. In *International Conference on Machine Learning*, pp. 1692–1700, 2014.

George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W Schuller. A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):417–429, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1243–1253, 2020.

Xin Xiong, Xiangyu Wang, Suorong Yang, Furao Shen, and Jian Zhao. Gmni: Achieve good data augmentation in unsupervised graph contrastive learning. *Neural Networks*, 181:106804, 2025.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pp. 5453–5462, 2018.

Liang Yang, Qiuliang Zhang, Runjie Shi, Wenmiao Zhou, Bingxin Niu, Chuan Wang, Xiaochun Cao, Dongxiao He, Zhen Wang, and Yuanfang Guo. Graph neural networks without propagation. In *Proceedings of the Web Conference*, pp. 469–477, 2023.

Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep autoencoder-like nonnegative matrix factorization for community detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1393–1402, 2018.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pp. 12121–12132, 2021.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference*, pp. 2069–2080, 2021.

A Baseline Descriptions

We compare against sixteen methods. **GCA** Zhu et al. (2021) adaptively selects suitable augmentation strategies to better preserve graph semantics. **GraphCL** You et al. (2020) leverages diverse graph augmentations to generate contrastive views. **DGI** Veličković et al. (2019) maximizes the mutual information between local node representations and a global summary vector. **GMI** Peng et al. (2020) estimates and maximizes mutual information at multiple granularities. **GRACE** Zhu et al. (2020) applies contrastive learning directly to node representations across two randomly augmented graph views. **ASP** Chen & Kou (2023) jointly considers both attribute and topological consistency. **AFGRL** Lee et al. (2022) eliminates handcrafted transformations via an internal contrasting mechanism. **iGCL** Li et al. (2024) promotes both invariance and discriminability in learned embeddings. **IDEAL** Sun et al. (2024) adaptively models inter-node dependencies to guide the contrastive objective. **SUGRL** Mo et al. (2022) uses random feature masking and walk-based sampling for unsupervised representation learning. Random-walk methods: **DeepWalk** Perozzi et al. (2014) performs truncated random walks with Skip-Gram; **node2vec** Grover & Leskovec (2016) interpolates

between BFS and DFS strategies; **LINE** Tang et al. (2015) preserves both first- and second-order proximity. Supervised GNNs: **GCN** Kipf & Welling (2017), **GAT** Veličković et al. (2017), and **AM-GCN** Wang et al. (2020).

B Parameter Analysis

We provide supplementary sensitivity results for the penalty factor λ and the interpolation coefficient α. For λ, values below 100 leave the non-negativity constraint insufficiently enforced, producing negative entries in $\mathbf{T}_s$ and $\mathbf{F}_s$ after several iterations; values above 1000 lead to numerical stiffness without further gains. For α, values near 0 over-rely on the initial topology $\mathbf{A}_0$ and limit the contribution of the reconstructed similarity, while values near 1 discard the original structure and destabilize training; $\alpha = 0.5$ consistently delivers the best trade-off. We therefore fix $\lambda = 300$ and $\alpha = 0.5$ across all datasets.

C Proofs of Theorems

C.1 Advantages and Theoretical Guarantees of DNMF in Semantic Extraction

Proof of Theorem 1

Assumptions: Let $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n \times n}$ be a symmetric adjacency matrix that is approximately block-diagonal:

$$\mathbf{A} = \begin{pmatrix} \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B}_M \end{pmatrix} + \mathbf{E},$$

where $\mathbf{B}_k$ corresponds to the k-th cluster block, and $\mathbf{E}$ is a small perturbation such that $\|\mathbf{E}\|_F$ is small. Each block $\mathbf{B}_k$ is non-negative with large intra-block values and near-zero inter-block values. For notational simplicity, we write $\mathbf{T}$ for the structural latent factor in a single-layer decomposition; the extension to deep NMF with $\mathbf{T}_s$ follows by stacking.

Step 1: NMF on Block-diagonal Matrices

We minimize the objective:

$$\min_{\mathbf{T},\mathbf{W} \geq 0} \|\mathbf{A} - \mathbf{T}\mathbf{W}\|_F^2$$

For a perfectly block-diagonal $\mathbf{A}$ ($\mathbf{E} = 0$), it has been shown Ding et al. (2005); Kuang et al. (2012) that the optimal solution $\mathbf{T}^*$ will have its row vectors nearly identical within the same block (up to scaling/rotation).

Since nodes u, v in the same cluster have identical connectivity patterns, setting $\mathbf{T}_u^* = \mathbf{T}_v^*$ minimizes the reconstruction error. Conversely, identical vectors for nodes in different clusters would significantly increase the error.

Step 2: Analysis of Row-wise Distances

Let $c_1, \ldots, c_M$ denote cluster indicator vectors. Then, the rows of $\mathbf{T}^*$ corresponding to the same cluster tend toward c_m (with small perturbations):

$$\|\mathbf{T}_u^* - \mathbf{T}_v^*\|_2 \approx 0, \quad \forall u, v \in \mathcal{C}_k$$
$$\|\mathbf{T}_u^* - \mathbf{T}_w^*\|_2 \gg 0, \quad u \in \mathcal{C}_k,\ w \in \mathcal{C}_{k'},\ k \neq k'$$

Step 3: Stability Under Perturbation

If the perturbation $\mathbf{E}$ is small, by matrix perturbation theory (specifically, the Davis–Kahan theorem on invariant subspaces), $\mathbf{T}^*$ still approximately preserves the block-wise clustering property, with the deviation bounded by $O(\|\mathbf{E}\|_F)$.

Step 4: Extension to Deep NMF

Deep NMF can be regarded as stacking multiple layers of NMF:

$$\mathbf{A} \approx \mathbf{T}_s \mathbf{W}_1 \mathbf{W}_2 \cdots \mathbf{W}_p$$

At each layer, the above clustering property is preserved, and the final $\mathbf{T}_s$ encodes the original block/cluster information, potentially with improved robustness or abstraction.

C.2 Time Complexity Analysis

We analyze the computational complexity of SAMA by considering its two primary components: the semantic-aware augmentation phase and the multi-view contrastive learning phase. During the semantic-aware augmentation stage, the model performs DNMF on both the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and the attribute matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. For each layer of DNMF, the complexity is approximately $\mathcal{O}(nr^2 + r^3)$, where r is the intermediate embedding dimension, typically much smaller than n. With p decomposition layers, the total DNMF cost becomes $\mathcal{O}(p \cdot (nr^2 + r^3))$. In addition, the model applies spectral filtering to the input features to extract global information, which has a complexity of $\mathcal{O}(n^2 d)$ in its exact form, or $\mathcal{O}(k|E|d)$ when using approximate methods such as Chebyshev polynomials, where k is the approximation order and $|E|$ is the number of edges. This entire augmentation procedure is repeated over K iterations to generate K semantically diverse views. Using approximate spectral filtering, the total cost is $\mathcal{O}(K \cdot (p \cdot (nr^2 + r^3) + |E|d))$.

In the contrastive learning stage, each view is passed through a GNN encoder with a typical complexity of $\mathcal{O}(|E|d)$. For K views, this amounts to $\mathcal{O}(K \cdot |E|d)$. The contrastive loss computation considers node-level positive and negative pairs across views. Although the naïve computation scales as $\mathcal{O}(Kn^2 d)$, in practice this can be reduced to $\mathcal{O}(Knd)$ using sampled negatives or in-batch approximations.

Combining both phases, the overall time complexity of SAMA is approximately $\mathcal{O}(K \cdot (p \cdot (nr^2 + r^3) + |E|d + nd))$, assuming approximate spectral filtering and efficient contrastive sampling. The complexity grows linearly with the number of views K, and remains practical for moderate-scale graphs. Further scalability may be achieved through sparse matrix approximations or view pruning strategies.