Boosting Graph Contrastive Learning via Manifold-Guided Representation Disentanglement

Zhiqiang Li[1], **Jianqing Liang**[2]*, **Jie Wang**[3], **Junbiao Cui**[1], **Zhiqiang Wang**[1], **Jiye Liang**[1]
[1]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[3]School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China
lizhiqiang2@sxu.edu.cn, liangjq@seu.edu.cn, wangjie@tyust.edu.cn, {cjb, wangzq, ljy}@sxu.edu.cn

Abstract

Graph contrastive learning constructs augmented views and aligns corresponding instances across views to learn view-invariant representations. However, existing methods usually enforce cross-view consistency within a single representation space, lacking an explicit characterization of view-specific information. This may suppress task-relevant complementary information, reduce embedding diversity, and induce dimensional collapse. To this end, we propose Manifold-Guided Representation Disentanglement (MGRD), a plug-and-play framework for boosting graph contrastive learning. MGRD decomposes graph representations into shared and complementary subspaces. The shared subspace captures view-invariant semantics while preserving local graph structure through a graph-anchored manifold prior, whereas the complementary subspace captures residual view-specific information beyond the shared representation and is regularized by asymmetric consistency and subspace decorrelation. By jointly exploiting shared and complementary representations, MGRD balances cross-view invariance and representation diversity. Experiments on node-level and graph-level benchmarks show that MGRD consistently improves multiple GCL backbones, while theoretical and empirical analyses demonstrate its effectiveness in mitigating dimensional collapse.

1 Introduction

Graph Contrastive Learning (GCL) has become a representative unsupervised paradigm for graph representation learning. By constructing augmented graph views and maximizing the consistency between corresponding instances, GCL learns effective node-level and graph-level representations without labels [1–4]. However, existing GCL methods usually enforce cross-view consistency in a single representation space, implicitly assuming that shared information across views is sufficient for downstream tasks, while treating view-specific discrepancies as noise. In fact, augmentations such as edge dropping, feature masking, and subgraph sampling may not only introduce perturbations, but also preserve complementary structural and semantic cues. For example, in molecular graphs, toxicity may be associated with a nitro group $-\mathrm{NO}_2$, whereas hydrophobicity may depend on the benzene ring. A feature-masked view may weaken nitro-related information while preserving aromatic structures, whereas a subgraph-sampled view may retain the nitro neighborhood but lose part of the benzene ring. Thus, view discrepancies are not merely noise; they may encode complementary chemical cues, with one view emphasizing aromatic-ring information and the other preserving nitro-related structures.

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

If the model treats such discrepancies as augmentation noise and enforces global consistency across views, it may retain only coarse-grained skeleton information while ignoring discriminative local structures such as the nitro group and benzene ring. As semantically inconsistent views are continuously pulled together in a single space, useful view-specific information can be suppressed, reducing representation diversity and inducing dimensional collapse, where embeddings degenerate into a low-rank subspace with extremely small variance [5–7]. This collapse weakens representation expressiveness and downstream generalization. Existing studies mitigate this issue through refined augmentations [8–10], variance-based regularization [5], or asymmetric encoder architectures [11]. However, manually designed augmentations may destroy task-relevant structures; variance regularization lacks explicit graph-geometric modeling; and asymmetric architectures cannot explicitly preserve view-specific complementary information. More importantly, most methods still emphasize cross-view invariance while overlooking the complementary semantic value of view discrepancies.

To this end, we propose MGRD, a lightweight plug-and-play framework for boosting graph contrastive learning. MGRD decomposes graph representations into a shared subspace and a complementary subspace. The shared subspace is used to capture view-invariant semantics across augmented views, and is constrained by a graph-anchored manifold prior constructed via Locally Linear Embedding (LLE) [12], thereby preserving the local geometric structure of the original graph and avoiding excessive compression of representations. The complementary subspace is used to model residual view-specific information that cannot be sufficiently explained by the shared representation, and is encouraged by asymmetric consistency and subspace decorrelation constraints to remain stable while being effectively separated from the shared subspace. In addition, MGRD operates in a fully feedforward manner, introduces only small computational overhead, and requires no modification to the original GCL architecture. It can be seamlessly integrated into representative GCL backbones such as GRACE [13], GCA [8], AUGCL [14], and E2Neg [15]. Extensive experiments on node-level and graph-level benchmarks show that MGRD consistently improves the performance of various GCL methods. Theoretical and empirical analyses further verify its effectiveness in mitigating dimensional collapse. Our main contributions are summarized as follows:

• We propose MGRD, a lightweight plug-and-play framework for enhancing graph contrastive learning, which decomposes graph representations into a shared subspace and a complementary subspace to jointly model view-invariant semantics and view-specific residual information.

• We introduce a graph-anchored manifold prior for the shared subspace, and design a complementary learning mechanism based on residual modeling, asymmetric consistency, and subspace decorrelation for the complementary subspace, thereby preserving useful complementary information while maintaining the local geometric structure of the graph.

• We demonstrate from both theoretical and experimental perspectives that MGRD can effectively mitigate dimensional collapse and consistently improve the performance of different GCL backbones on multiple node-level and graph-level benchmarks.

2 Related Work

2.1 Graph Contrastive Learning

Graph Contrastive Learning (GCL) learns node-level and graph-level representations by constructing augmented graph views and aligning corresponding instances across views. Representative methods include GRACE [13], BGRL [11], SimGRACE [16], GCA [8], and GTCA [17], which commonly use edge perturbation, feature masking, subgraph sampling, or adaptive augmentation to generate different views [18–21]. However, most GCL methods enforce consistency in a single representation space and tend to treat view-specific discrepancies as noise. Such over-alignment may suppress useful complementary information, reduce representation diversity, and weaken downstream generalization [7, 6]. Different from prior methods that mainly refine augmentations or alignment objectives, MGRD decomposes graph representations into shared and complementary subspaces to preserve both view-invariant semantics and view-specific residual information.

2.2 Dimensional Collapse and Manifold Regularization

Dimensional collapse is a major challenge in self-supervised learning [22, 23]. In GNNs, oversmoothing makes node representations increasingly similar after repeated message passing [24, 25].

In GCL, strong cross-view alignment may further compress representations into a low-rank subspace with small variance, causing dimensional collapse [5, 7]. Existing methods mitigate collapse through variance regularization [5], asymmetric encoder architectures [11], or improved augmentations [8], but they often lack explicit graph-geometric awareness or fail to preserve complementary view-specific cues. Manifold learning methods such as LLE [12] and Laplacian Eigenmaps [26] preserve local geometry. ML^2-GCL [27] introduces manifold priors into GCL, while CL-GCL [28] combines graph coarsening with manifold learning to capture community-level structural semantics and local geometric relations. In contrast, MGRD applies a graph-anchored manifold prior to the shared subspace while modeling residual information in the complementary subspace, mitigating collapse while maintaining representation diversity.

3 Method

3.1 Problem Setup

Let $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ denote a graph with node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ and adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$. Graph contrastive learning aims to learn a GNN encoder $f_\theta(\mathbf{X}, \mathbf{A})$ without task labels. Given two stochastic augmentations, we obtain two graph views $\mathcal{G}_u = (\mathbf{X}_u, \mathbf{A}_u)$ and $\mathcal{G}_v = (\mathbf{X}_v, \mathbf{A}_v)$, which are encoded as $\mathbf{Z}_u = f_\theta(\mathbf{X}_u, \mathbf{A}_u), \mathbf{Z}_v = f_\theta(\mathbf{X}_v, \mathbf{A}_v)$.

Although most GCL methods learn view-invariant representations through cross-view consistency, enforcing all information within a single representation space may suppress view-specific cues, reduce embedding diversity, and further induce dimensional collapse. To address this issue, we propose MGRD, a manifold-guided representation disentanglement framework, as shown in Figure 1.

MGRD decomposes each view representation into a shared component and a complementary component. The shared components $\mathbf{Z}_u^S$ and $\mathbf{Z}_v^S$ capture view-invariant semantics under a graph-anchored manifold prior, while the complementary components $\mathbf{Z}_u^C$ and $\mathbf{Z}_v^C$ preserve residual view-dependent information that is not sufficiently explained by the shared subspace. In MGRD, the contrastive loss is applied to the shared representations: $\mathcal{L}_{\text{gcl}} = \frac{1}{2}\left[\ell(\mathbf{Z}_u^S, \mathbf{Z}_v^S) + \ell(\mathbf{Z}_v^S, \mathbf{Z}_u^S)\right]$, where

$$\ell(\mathbf{Z}_u^S, \mathbf{Z}_v^S) = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(p(\mathbf{z}_{u,i}^S), p(\mathbf{z}_{v,i}^S))/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(p(\mathbf{z}_{u,i}^S), p(\mathbf{z}_{v,j}^S))/\tau)}. \tag{1}$$

Here, $p(\cdot)$ is a projection head, $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and τ is the temperature parameter.

3.2 Graph-Anchored Manifold Prior

To prevent cross-view consistency from suppressing local structural diversity in the shared subspace, we introduce a graph-anchored manifold prior to regularize $\mathbf{Z}_u^S$ and $\mathbf{Z}_v^S$, enabling them to capture view-invariant semantics while preserving the local geometry of the original graph. Different from traditional Locally Linear Embedding (LLE) [12], which mines local manifold structure mainly from Euclidean neighborhoods in the feature space, our method incorporates the topology of graph data. Specifically, for each node v_i, we first construct its K-hop topological neighborhood $\mathcal{N}_K(i)$ from the original graph. Then, within this topology-constrained neighborhood, we select a local neighbor set $\mathcal{P}_i \subseteq \mathcal{N}_K(i)$ and approximate the feature of v_i as a convex combination of its neighbors:

$$\min_{\{w_{ij}\}_{j\in\mathcal{P}_i}} \sum_{i=1}^{N} \Big(\big\|\mathbf{x}_i - \sum_{j\in\mathcal{P}_i} w_{ij}\mathbf{x}_j\big\|_2^2 + \lambda\|\mathbf{w}_i\|_2^2 \Big), \quad \text{s.t. } w_{ij} \geq 0,\ \sum_{j\in\mathcal{P}_i} w_{ij} = 1. \tag{2}$$

Here, $\mathbf{w}_i = \{w_{ij}\}_{j\in\mathcal{P}_i}$ denotes the reconstruction weights of node v_i, and λ is a regularization coefficient. The resulting weights encode topology-aware local manifold relationships, since the reconstruction is restricted to nodes within the K-hop graph neighborhood rather than arbitrary feature-space neighbors.

We transfer the mined local manifold structure into the shared representation space by requiring $\mathbf{Z}_u^S$ and $\mathbf{Z}_v^S$ to preserve the same reconstruction relationships:

$$\mathcal{L}_{\text{mpr}} = \frac{1}{2N}\sum_{i=1}^{N}\Big(1 - \cos\Big(\mathbf{Z}_{u_i}^S, \sum_{j\in\mathcal{P}_i} w_{ij}\mathbf{Z}_{u_j}^S\Big)\Big) + \frac{1}{2N}\sum_{i=1}^{N}\Big(1 - \cos\Big(\mathbf{Z}_{v_i}^S, \sum_{j\in\mathcal{P}_i} w_{ij}\mathbf{Z}_{v_j}^S\Big)\Big). \tag{3}$$

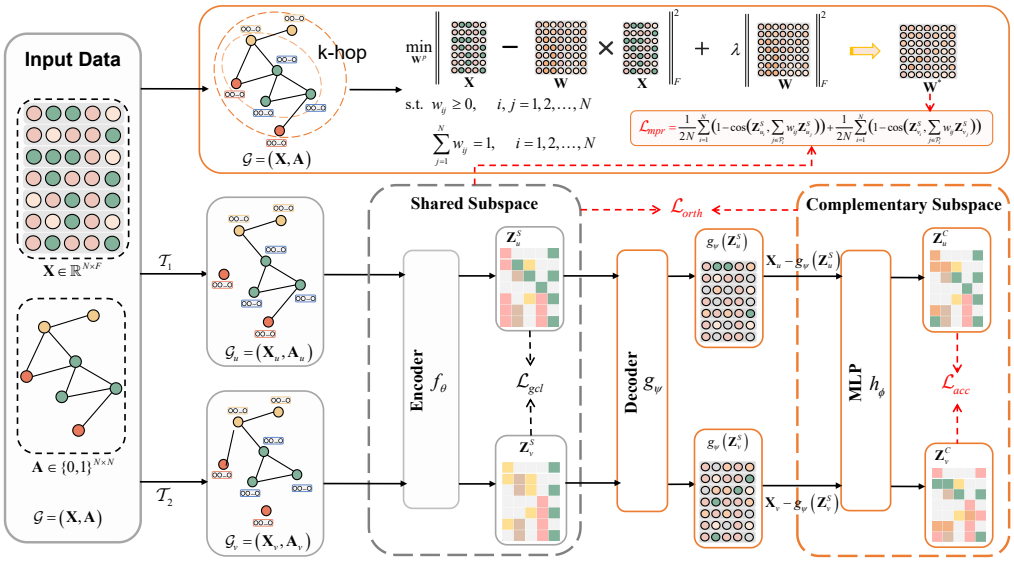

Figure 1: Overview of MGRD. MGRD decomposes graph representations into shared and complementary subspaces, where the shared subspace is regularized by a graph-anchored manifold prior and the complementary subspace captures residual view-specific information. Asymmetric consistency and subspace decorrelation stabilize the decomposition and help mitigate dimensional collapse.

This loss encourages the shared representations to preserve the topology-aware local reconstruction structure derived from the original graph.

By introducing $\mathcal{L}_{\text{mpr}}$, the shared subspace is anchored to the local graph manifold rather than being solely optimized by cross-view alignment. Together with the contrastive objective, this regularization encourages preservation of local reconstruction relationships and empirically helps mitigate dimensional collapse.

3.3 Residual-Based Complementary Subspace

The shared representations $\mathbf{Z}_u^S$ and $\mathbf{Z}_v^S$ are designed to capture view-invariant semantics across augmented views. However, forcing all information to be encoded and aligned in the shared subspace may suppress useful view-dependent cues introduced or preserved by graph augmentations. To preserve such complementary information, MGRD further constructs a residual-based complementary subspace to model feature variations that are not fully captured by the shared representations.

Formally, for each augmented view, we map the shared representation back to the input feature space to estimate the feature component explained by the shared subspace:

$$\widehat{\mathbf{X}}_u = g_\psi(\mathbf{Z}_u^S), \qquad \widehat{\mathbf{X}}_v = g_\psi(\mathbf{Z}_v^S), \tag{4}$$

where $g_\psi(\cdot)$ is a lightweight MLP projection module. The residual signals are then defined as:

$$\mathbf{R}_u = \mathbf{X}_u - \widehat{\mathbf{X}}_u, \qquad \mathbf{R}_v = \mathbf{X}_v - \widehat{\mathbf{X}}_v. \tag{5}$$

Here, the residual signals are not required to faithfully reconstruct the input features. Instead, they serve as a feature-space residual pathway that exposes information not explicitly encoded in the shared representations and is optimized end-to-end by the complementary learning objectives. We encode the residual signals into complementary representations:

$$\mathbf{Z}_u^C = h_\phi(\mathbf{R}_u), \qquad \mathbf{Z}_v^C = h_\phi(\mathbf{R}_v), \tag{6}$$

where $h_\phi(\cdot)$ is a residual encoder implemented by an MLP.

Although the residual encoder captures view-dependent residual signals, these signals may also contain augmentation-induced noise. Without proper constraints, the complementary subspace may

become unstable or degenerate into noisy representations. Therefore, we introduce asymmetric consistency constraint to stabilize complementary representation learning:

$$\mathcal{L}_{\text{acc}} = \frac{1}{2N} \left\| \mathbf{Z}_u^C - \text{sg}(\mathbf{Z}_v^C) \right\|_F^2 + \frac{1}{2N} \left\| \mathbf{Z}_v^C - \text{sg}(\mathbf{Z}_u^C) \right\|_F^2 . \tag{7}$$

Here, $\text{sg}(\cdot)$ denotes the stop-gradient operation, making each consistency term asymmetric in terms of gradient propagation. This design encourages the complementary representations from two augmented views to reside in a comparable latent space, while preventing each branch from being simultaneously optimized toward the other in the same gradient path. As a result, the complementary subspace captures stable residual semantics beyond the shared view-invariant representation, rather than merely fitting augmentation noise.

Although the asymmetric consistency constraint stabilizes the complementary subspace, it does not explicitly separate it from the shared subspace. To reduce redundancy between $\mathbf{Z}^S$ and $\mathbf{Z}^C$ and encourage effective disentanglement, we introduce a sample-wise orthogonality constraint:

$$\mathcal{L}_{\text{orth}} = \frac{1}{2N} \sum_{i=1}^{N} \left[\cos^2(\mathbf{z}_{u,i}^S, \mathbf{z}_{u,i}^C) + \cos^2(\mathbf{z}_{v,i}^S, \mathbf{z}_{v,i}^C) \right] , \tag{8}$$

where $\cos(\cdot, \cdot)$ denotes the cosine similarity. This constraint encourages the shared and complementary subspaces to encode less redundant and more complementary information.

3.4 Overall Objective and Inference

The overall training objective is defined by jointly optimizing the standard graph contrastive loss, the graph-anchored manifold prior, the asymmetric complementary consistency loss, and the subspace orthogonality regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{gcl}} + \alpha \mathcal{L}_{\text{mpr}} + \beta \mathcal{L}_{\text{acc}} + \gamma \mathcal{L}_{\text{orth}}, \tag{9}$$

where α, β, and γ are trade-off hyperparameters controlling the contribution of each regularization term.

After pre-training, MGRD evaluates on the original graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$. The shared representation is obtained by $\mathbf{Z}^S = f_\theta(\mathbf{X}, \mathbf{A})$, and the complementary representation is computed from the residual signal as $\mathbf{Z}^C = h_\phi\big(\mathbf{X} - g_\psi(\mathbf{Z}^S)\big)$. The final representation is $\mathbf{Z} = [\mathbf{Z}^S \| \mathbf{Z}^C]$, which is directly used for node-level evaluation or pooled by a readout function for graph-level evaluation. The detailed training procedure is summarized in Algorithm 1 in Appendix D.

3.5 Theoretical Analysis

In graph contrastive learning, the encoder produces node representations $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^\top \in \mathbb{R}^{N \times d}$. A desirable embedding space should distribute information across multiple dimensions. However, when representation collapse occurs, embeddings may concentrate around a few dominant directions or degenerate to nearly identical vectors. Let $\Sigma_Z = \text{Cov}(\mathbf{Z})$ denote the covariance matrix of the embeddings. Dimensional collapse can be reflected by a low-rank or low-variance covariance structure:

$$\text{rank}(\Sigma_Z) \ll d \quad \text{or} \quad \text{tr}(\Sigma_Z) \to 0. \tag{10}$$

Under the Gaussian assumption, the differential entropy of $\mathbf{Z}$ is

$$H(\mathbf{Z}) = \frac{1}{2} \log\big((2\pi e)^d \det(\Sigma_Z)\big) . \tag{11}$$

When Σ_Z becomes nearly singular, $\det(\Sigma_Z)$ decreases toward zero, leading to entropy degeneration. Therefore, mitigating dimensional collapse can be interpreted as preserving the effective volume of the embedding distribution.

MGRD improves representation diversity through a disentangled representation design. The shared subspace $\mathbf{Z}^S$ captures view-invariant semantics and is regularized by the graph-anchored manifold prior to preserve local graph geometry. The complementary subspace $\mathbf{Z}^C$ captures residual information beyond the shared representation, while the orthogonality constraint reduces redundancy between the two subspaces. The final representation is formed by concatenation: $\mathbf{Z} = [\mathbf{Z}^S \| \mathbf{Z}^C]$.

We formalize this intuition under a mild non-degeneracy condition. Specifically, assume that the shared and complementary subspaces retain non-trivial variance in their effective dimensions, and that their cross-correlation is bounded:

$$\lambda_{\min}(\Sigma_S) \geq \epsilon_S, \quad \lambda_{\min}(\Sigma_C) \geq \epsilon_C, \quad \left\| \Sigma_S^{-\frac{1}{2}} \Sigma_{SC} \Sigma_C^{-\frac{1}{2}} \right\|_2 \leq \rho < 1, \tag{12}$$

where $\Sigma_S = \mathrm{Cov}(\mathbf{Z}^S)$, $\Sigma_C = \mathrm{Cov}(\mathbf{Z}^C)$, and $\Sigma_{SC} = \mathrm{Cov}(\mathbf{Z}^S, \mathbf{Z}^C)$. This condition is encouraged jointly by the manifold prior, residual complementary learning, and orthogonality regularization, rather than guaranteed by any single component alone.

Proposition 3.1 (Entropy Preservation of MGRD)**.** *Let $\mathbf{Z} = [\mathbf{Z}^S \| \mathbf{Z}^C]$ be the final representation, where d_S and d_C are the effective dimensions of the shared and complementary subspaces, respectively. Under the above non-degeneracy condition, the entropy of $\mathbf{Z}$ satisfies*

$$H(\mathbf{Z}) \geq \frac{1}{2} \log\Big((2\pi e)^d \epsilon_S^{d_S} \epsilon_C^{d_C} (1-\rho^2)^m \Big), \tag{13}$$

where $d = d_S + d_C$ and $m = \min(d_S, d_C)$.

This result suggests that MGRD preserves the entropy of the learned representation by maintaining non-trivial variance in both shared and complementary subspaces while reducing their redundancy. Compared with conventional GCL methods that enforce consistency in a single representation space, MGRD introduces an additional residual-aware subspace and explicitly encourages subspace decorrelation, thereby alleviating low-rank degeneration and improving representation diversity.

4 Experiments

We conduct comprehensive experiments to investigate the following research questions: **Q1:** Does MGRD improve the performance of the GCL method on different tasks and datasets? **Q2:** How well does MGRD demonstrate generalization, efficiency, and scalability across diverse datasets and GCL models? **Q3:** Can MGRD alleviate dimensional collapse in graph contrastive learning? **Q4:** What are the individual and combined effects of the manifold prior, complementary encoder, and orthogonality constraint in MGRD?

4.1 Experimental Setup

We evaluate MGRD on graph-level and node-level classification benchmarks. The graph-level benchmarks include MUTAG, DD, NCI1, PROTEINS, COLLAB, REDDIT-B, and IMDB-B, while the node-level benchmarks include Cora, CiteSeer, PubMed, DBLP, Amazon-Computers, Amazon-Photo, and Ogbn-Arxiv. Dataset statistics are summarized in Table 6, with detailed descriptions in Appendix A.1. We adopt a standard two-stage protocol of unsupervised pretraining followed by linear evaluation. For graph classification, MGRD is plugged into GraphCL [29] with a GIN encoder [30], and a linear SVM is trained on frozen graph embeddings using 10-fold cross-validation with 5 repeats. For node classification, MGRD is integrated into GCA [8] and evaluated following the DGI protocol [18], with a logistic regression classifier trained on frozen node embeddings over 20 runs. All models are implemented in PyTorch and trained on an NVIDIA RTX 3090 GPU, using the same encoder architectures, training epochs, and augmentations as the corresponding baselines for fair comparison. Additional evaluation and implementation details are provided in Appendix A.2.

4.2 Graph and Node Classification (Q1)

We evaluate MGRD on both graph-level and node-level classification tasks. For graph-level evaluation, we compare MGRD with classical baselines, including GK [31], WL [32], DGK [33], node2vec [34], sub2vec [35], and graph2vec [36], as well as recent GCL-based methods such as InfoGraph [37], JOAOv2 [38], SimGRACE [16], GraphCL [29], DCL [39], HCL [40], ProGCL [41], AUGCL [14], and E2Neg [15]. For node-level evaluation, we include representative traditional and deep learning baselines, including DeepWalk [42], GAE [43], VGAE [43], GAT [44], DGI [18], GRACE [13], GCA [8], GDCL [45], ProGCL [41], AUGCL [14], and E2Neg [15].

As shown in Table 1 and Table 2, MGRD achieves consistent improvements across diverse datasets and backbones. In particular, the manifold-prior guided shared subspace preserves structural consistency, while the complementary and orthogonal components capture augmentation-specific variations and reduce redundancy, leading to more robust and expressive representations.

Table 1: Graph classification accuracy (%) on multiple benchmark datasets using the GraphCL backbone. The proposed MGRD consistently enhances the baseline performance across diverse datasets, demonstrating its effectiveness in improving graph-level representation learning. Best results are highlighted in **bold**.

Method	MUTAG	DD	NCI1	PROTEINS	COLLAB	REDDIT-B	IMDB-B
GK	81.7±2.11	-	-	-	-	77.3±0.18	65.9±0.98
WL	80.7±3.00	-	80.0±0.50	72.9±0.56	-	68.8±0.41	72.3±3.44
DGK	87.4±2.72	-	80.3±0.46	73.3±0.82	-	78.0±0.39	67.0±0.56
node2vec	72.6±10.20	-	54.9±1.61	57.5±3.57	-	-	-
sub2vec	61.1±15.80	-	52.8±1.47	53.0±5.55	-	71.5±0.41	55.3±1.54
graph2vec	83.2±9.25	-	73.2±1.81	73.3±2.05	-	75.8±1.03	71.1±0.54
InfoGraph	89.0±1.13	72.9±1.78	76.2±1.06	74.4±0.31	70.7±1.11	82.5±1.42	73.0±0.87
JOAOv2	87.7±0.79	77.4±1.15	78.4±0.53	74.1±1.10	69.3±0.86	86.4±1.45	70.8±0.25
SimGRACE	89.0±1.31	77.4±1.11	79.1±0.44	75.4±0.09	71.7±0.84	89.5±0.89	71.3±0.77
GraphCL	87.2±0.86	78.0±0.77	78.3±0.39	74.4±0.44	71.5±1.06	90.1±0.70	71.2±0.72
DCL	88.3±1.75	76.8±1.24	77.6±0.67	74.7±0.39	70.4±0.99	89.9±0.72	70.6±0.58
HCL	88.9±1.22	76.8±1.15	78.2±0.53	74.4±0.77	70.4±0.85	90.1±0.47	71.4±0.62
ProGCL	88.7±1.40	79.2±0.90	76.9±0.47	74.5±0.37	70.5±0.82	90.5±0.49	71.6±0.59
AUGCL	89.2±1.08	80.1±0.54	80.2±0.34	75.8±0.39	72.1±0.65	91.2±0.44	72.5±0.80
E2Neg	89.5±0.95	79.7±0.43	79.3±0.56	75.5±0.36	71.5±0.52	90.9±0.35	72.1±0.73
GraphCL + MGRD	**89.7±0.75**	**81.5±0.62**	**80.5±0.53**	**76.9±0.43**	**72.8±0.35**	**92.3±0.53**	**73.5±0.56**

Table 2: Node classification accuracy (%) on multiple benchmark datasets using the GCA backbone. The integration of the proposed MGRD module consistently improves performance across diverse datasets, demonstrating its effectiveness in enhancing node-level representation learning. Best results are highlighted in **bold**.

Method	Cora	CiteSeer	PubMed	DBLP	Computers	Photo
DeepWalk	70.7±0.25	51.4±0.36	74.3±0.18	75.9±0.10	85.6±0.07	89.4±0.05
GAE	71.5±0.39	60.6±0.43	72.1±0.27	81.2±0.07	85.2±0.13	91.6±0.06
VGAE	78.9±0.26	61.2±0.59	79.0±0.35	81.7±0.05	86.3±0.15	92.2±0.07
GAT	83.0±0.30	72.5±0.65	79.0±0.40	81.5±0.03	85.2±0.09	91.2±0.05
DGI	83.8±0.36	72.0±0.72	76.8±0.45	83.2±0.15	83.9±0.05	91.6±0.09
GRACE	83.2±0.75	72.1±1.51	86.7±0.19	84.1±0.34	86.8±0.32	91.8±0.15
GCA	83.7±0.79	72.5±1.29	86.8±0.21	84.2±0.23	87.5±0.35	92.2±0.43
GDCL	84.5±0.59	72.8±1.13	81.5±0.35	82.1±0.42	85.9±0.50	90.1±0.59
ProGCL	83.7±0.45	72.7±1.09	85.9±0.26	83.7±0.36	88.1±0.43	93.2±0.32
AUGCL	84.2±0.43	72.6±1.25	85.5±0.35	84.1±0.26	**88.9±0.39**	93.6±0.53
E2Neg	85.1±0.51	72.9±0.36	86.7±1.09	84.5±0.52	88.7±0.96	93.3±0.76
GCA + MGRD	**85.9±0.42**	**73.5±0.93**	**87.9±0.32**	**85.7±0.25**	88.7±0.35	**94.2±0.45**

4.3 Generalization (Q2)

To examine the general applicability of the proposed MGRD module, we conduct comprehensive experiments by integrating it into six widely adopted GCL backbones: GRACE [13], GCA [8], GDCL [45], ProGCL [41], AUGCL [14], and E2Neg [15]. For each method, MGRD retains the backbone encoder architecture and augments training with the proposed auxiliary modules and regularization terms. The results in Table 3 demonstrate that MGRD consistently improves node classification accuracy across all datasets and backbones. Its plug-and-play design further ensures easy adoption in future contrastive graph learning paradigms, offering both stability and scalability benefits.

4.4 Efficiency and Scalability (Q2)

We evaluate the efficiency and scalability of MGRD on PubMed and Ogbn-Arxiv in terms of accuracy, training time, and GPU memory usage. MGRD is integrated into GRACE [13], CCA-SSG [46], and StructComp [47], and compared with their original versions under identical settings. As shown in Table 7 in Appendix B.1, MGRD improves StructComp on Ogbn-Arxiv with only marginal additional memory and time costs, showing strong scalability and efficiency. Moreover, manifold-prior construction after StructComp compression introduces a modest one-time preprocessing cost.

Table 3: Node classification accuracy (%) on multiple benchmark datasets using MGRD. Improvements over the base GCL methods are highlighted in **bold**, demonstrating the effectiveness of the proposed plugin module.

Method	Cora	CiteSeer	PubMed	DBLP	Computers	Photo
GRACE	83.2±0.75	72.1±1.51	86.7±0.19	84.1±0.34	86.8±0.32	91.8±0.15
GRACE + **MGRD**	**85.3±0.35**	**73.3±0.38**	**87.5±0.26**	**84.9±0.23**	**87.9±0.27**	**92.6±0.20**
GCA	83.7±0.79	72.5±1.29	86.8±0.21	84.2±0.23	87.5±0.35	92.2±0.43
GCA + **MGRD**	**85.9±0.42**	**73.5±0.93**	**87.9±0.32**	**85.7±0.25**	**88.7±0.35**	**94.2±0.45**
GDCL	84.5±0.59	72.8±1.13	81.5±0.35	82.1±0.42	85.9±0.50	90.1±0.59
GDCL + **MGRD**	**85.1±0.51**	**73.3±1.06**	**82.6±0.30**	**83.2±0.36**	**87.2±0.45**	**91.3±0.47**
ProGCL	83.7±0.45	72.7±1.09	85.9±0.26	83.7±0.36	88.1±0.43	93.2±0.32
ProGCL + **MGRD**	**84.6±0.39**	**73.5±1.01**	**87.1±0.23**	**84.5±0.31**	**88.9±0.36**	**93.8±0.29**
AUGCL	84.2±0.43	72.6±1.25	85.5±0.35	84.1±0.26	88.9±0.39	93.6±0.53
AUGCL + **MGRD**	**85.3±0.38**	**73.2±1.11**	**86.9±0.31**	**84.8±0.22**	**89.3±0.34**	**94.0±0.46**
E2Neg	85.1±0.51	72.9±0.36	86.7±1.09	84.5±0.52	88.7±0.96	93.3±0.76
E2Neg + **MGRD**	**85.7±0.43**	**73.3±0.56**	**87.4±0.35**	**85.2±0.33**	**89.0±0.45**	**93.9±0.56**

The resulting weights are cached and reused throughout training, so construction is not repeated at each epoch. Detailed preprocessing costs are reported in Table 8 in the Appendix B.1.

4.5 Evidence for Collapse Mitigation (Q3)

Effective Rank Analysis. We measure the Effective Rank (ER) [48] of learned embeddings to evaluate representation collapse. A higher ER indicates that information is distributed across more feature dimensions, while a lower ER suggests low-rank degeneration. As shown in Figure 2, GRACE and GCA show a clear decline in ER during training, indicating increasing dimensional collapse. In contrast, MGRD maintains a higher and more stable ER on both datasets, demonstrating its ability to preserve representation diversity.

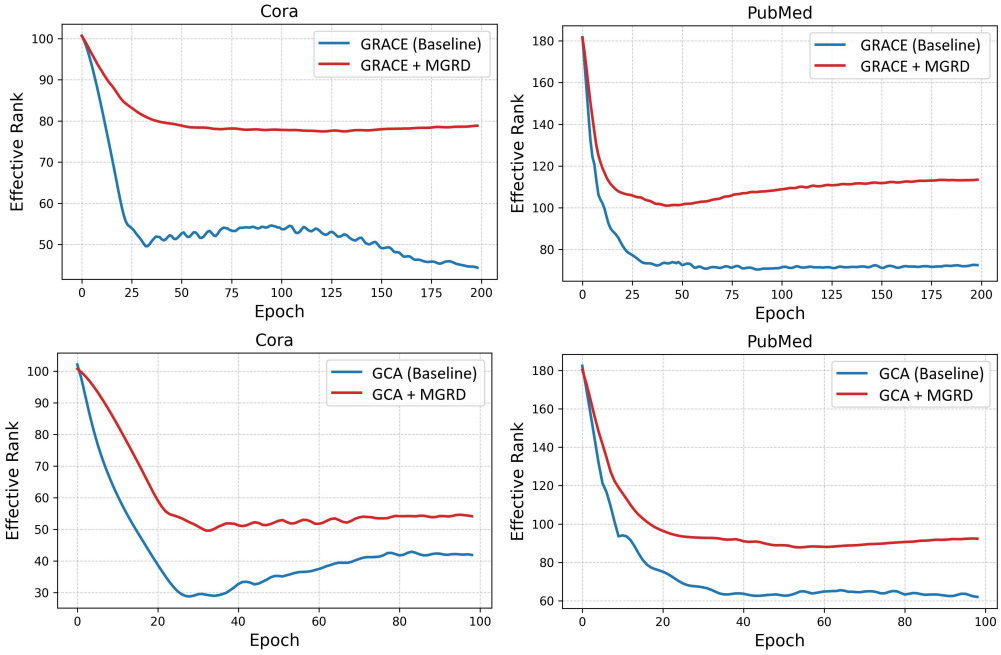

Figure 2: Evolution of ER during training on Cora and PubMed with GRACE and GCA backbones.

Gaussian Entropy Analysis. We further measure the Gaussian entropy of learned embeddings to evaluate representation diversity. Higher entropy indicates a more dispersed embedding distribution,

while lower entropy suggests potential dimensional collapse. For visualization, entropy values are normalized to $[0, 1]$ across training epochs. As shown in Figure 3, GRACE and GCA converge

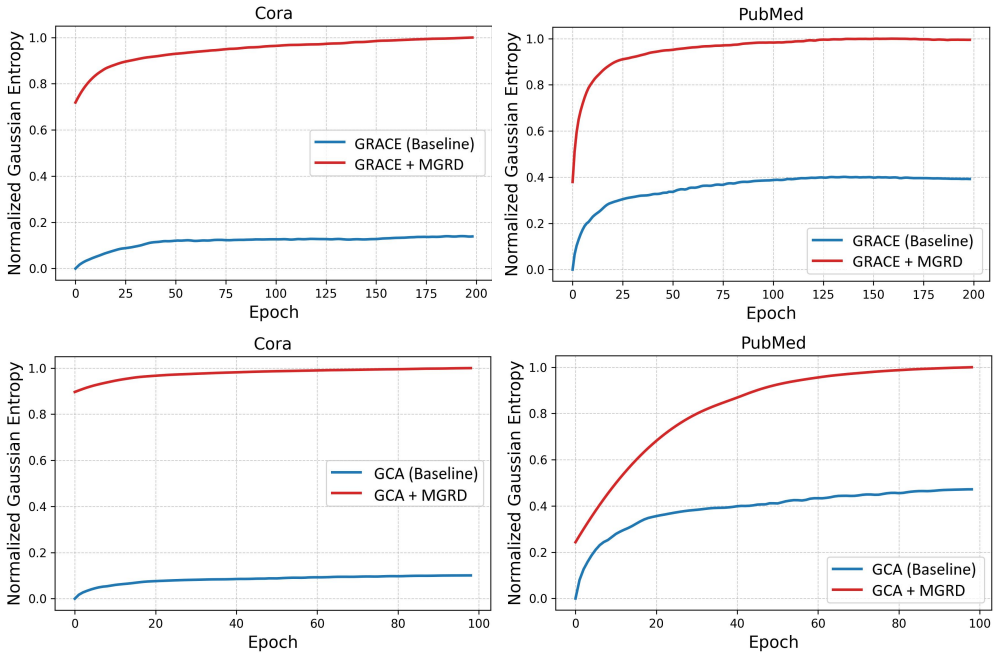

Figure 3: Normalized Gaussian entropy evolution on Cora and PubMed with GRACE and GCA backbones.

to relatively lower entropy, suggesting that their embeddings may concentrate in fewer dominant directions. In contrast, MGRD maintains consistently higher entropy on both datasets, demonstrating its effectiveness in preserving embedding diversity and mitigating dimensional collapse.

4.6 Ablation Study (Q4)

Component-level ablation. To evaluate the contribution of each component in MGRD, we perform ablation studies on Cora, CiteSeer, PubMed, and Photo. We compare three variants: (1) w/o mpr, removing manifold prior regularization; (2) w/o acc, removing the residual-based complementary encoder; and (3) w/o orth, removing the orthogonality constraint. As shown in Figure 4, all components consistently improve performance. In particular, removing $\mathcal{L}_{\text{mpr}}$ leads to the largest drop, indicating the importance of structure-preserving regularization.

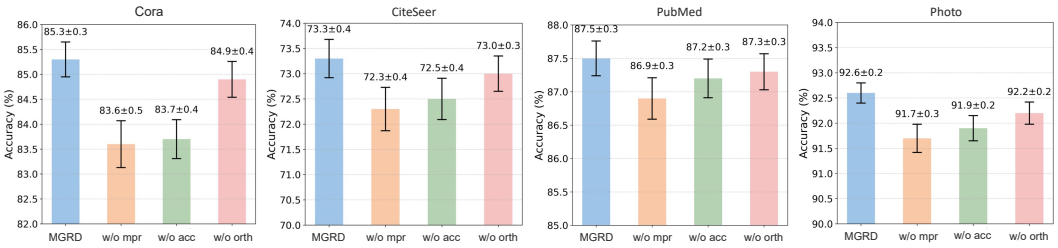

Figure 4: Ablation study of MGRD on node classification using the GRACE backbone across Cora, CiteSeer, PubMed, and Photo.

Representation-level ablation. To examine the predictive contributions of the shared and complementary subspaces, we separately evaluate Z^S, Z^C, and their concatenation $[Z^S \| Z^C]$ under the linear evaluation protocols in Section 4.1. Table 4 presents results on selected graph and node

Table 4: Representation-level ablation on selected benchmarks. Graph and node classification use GraphCL + MGRD and GCA + MGRD, respectively. Accuracy (%) is reported, with the best result per dataset in bold. Δ denotes the gain of Combined over Shared in percentage points (pp).

Representation	Graph classification				Node classification		
	MUTAG	NCI1	PROTEINS	REDDIT-B	Cora	PubMed	Photo
Shared (Z^S)	88.1	73.7	72.1	86.7	84.8	87.1	91.7
Complementary (Z^C)	85.7	70.2	67.4	81.8	83.9	85.7	89.4
Combined ($[Z^S \| Z^C]$)	**89.7**	**80.5**	**76.9**	**92.3**	**85.9**	**87.9**	**94.2**
Δ (pp) $\uparrow$	+1.6	+6.8	+4.8	+5.6	+1.1	+0.8	+2.5

classification benchmarks; results on the remaining datasets are provided in Table 9 in Appendix B.2. Across all evaluated datasets, Z^S consistently outperforms Z^C, while their concatenation achieves the best performance, improving over Z^S alone by 1.6–6.8 percentage points on graph classification and 0.8–2.5 percentage points on node classification. These results support the predictive value of the complementary subspace beyond the shared representation.

4.7 Hyperparameter Sensitivity Analysis (Q4)

We evaluate the sensitivity of MGRD to three hyperparameters: α for $\mathcal{L}_{\mathrm{mpr}}$, β for $\mathcal{L}_{\mathrm{acc}}$, and γ for $\mathcal{L}_{\mathrm{orth}}$. Following standard practice, we vary one hyperparameter in $\{0.1, 0.5, 1.0, 2.0, 5.0\}$ while fixing the others to 1.0, and report node classification accuracy on *Cora*. As shown in Table 5, performance generally improves when increasing α, β, or γ up to 1.0, confirming the effectiveness of each term. However, overly large α, β, or γ degrades accuracy, indicating that excessive regularization may harm representation learning.

Table 5: Hyperparameter sensitivity of MGRD on Cora with GCA backbone.

α	Acc.	Δ	β	Acc.	Δ	γ	Acc.	Δ
0.1	83.4	-2.5	0.1	83.7	-2.2	0.1	84.0	-1.9
0.5	84.5	-1.4	0.5	84.6	-1.3	0.5	84.8	-1.1
1.0	**85.9**	**0.0**	**1.0**	**85.9**	**0.0**	**1.0**	**85.9**	**0.0**
2.0	84.9	-1.0	2.0	85.3	-0.6	2.0	85.0	-0.9
5.0	84.0	-1.9	5.0	83.9	-2.0	5.0	84.6	-1.3

5 Conclusion

We presented MGRD, a manifold-guided representation disentanglement framework to mitigate over-alignment and representation collapse in graph contrastive learning. MGRD decomposes embeddings into shared and complementary subspaces, where the shared subspace is regularized by a structure-aware manifold prior and the complementary subspace preserves augmentation-specific variations via asymmetric consistency and orthogonality constraints. We further provided theoretical insights suggesting that MGRD can help mitigate entropy collapse and low-rank degeneration under suitable conditions. Extensive experiments on node- and graph-level benchmarks validate consistent improvements across diverse GCL backbones.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (U25A20529, 62376142, 62272285, 62306205), and the Shanxi "Sanjin Talents" Support Program (Young Top-notch Talent in Scientific and Technological Innovation). We extend our heartfelt gratitude to the anonymous reviewers for their invaluable and constructive comments.

References

[1] Hanchen Wang, Jean Kaddour, Shengchao Liu, Jian Tang, Joan Lasenby, and Qi Liu. Evaluating self-supervised learning for molecular graph embeddings. In *Proceedings of the Advances in Neural Information Processing Systems*, 2023.

[2] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 21872–21884, 2021.

[3] Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. Simple and asymmetric graph contrastive learning without augmentations. In *Proceedings of the Advances in Neural Information Processing Systems*, 2023.

[4] Yuena Lin, Gengyu Lyu, Haichun Cai, Deng-Bao Wang, Haobo Wang, and Zhen Yang. Simplified graph contrastive learning model without augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *The International Conference on Learning Representations*, 2022.

[6] Yifei Zhang, Hao Zhu, Menglin Yang, Jiahong Liu, Rex Ying, Irwin King, and Piotr Koniusz. Understanding and mitigating hyperbolic dimensional collapse in graph contrastive learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1984–1995, 2025.

[7] Jingyu Liu, Huayi Tang, and Yong Liu. Perfect alignment may be poisonous to graph contrastive learning. In *Proceedings of the International Conference on Machine Learning*, pages 31890–31916, 2024.

[8] 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*, pages 2069–2080, 2021.

[9] Haonan Zhang, Yuyang Ren, Luoyi Fu, Xinbing Wang, Guihai Chen, and Chenghu Zhou. Multi-scale self-supervised graph contrastive learning with injective node augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):261–274, 2023.

[10] Xin Zhang, Qiaoyu Tan, Xiao Huang, and Bo Li. Graph contrastive learning with personalized augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6305–6316, 2024.

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

[12] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

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

[14] Chaoxi Niu, Guansong Pang, and Ling Chen. Affinity uncertainty-based hard negative mining in graph contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11681–11691, 2024.

[15] Yongqi Huang, Jitao Zhao, Dongxiao He, Di Jin, Yuxiao Huang, and Zhen Wang. Does gcl need a large number of negative samples? enhancing graph contrastive learning with effective and efficient negative sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17511–17518, 2025.

[16] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *The ACM Web Conference*, pages 1070–1079, 2022.

[17] Jianqing Liang, Xinkai Wei, Min Chen, Zhiqiang Wang, and Jiye Liang. Gnn-transformer cooperative architecture for trustworthy graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18667–18675, 2025.

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

[19] 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 *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1150–1160, 2020.

[20] Yue Yu, Xiao Wang, Mengmei Zhang, Nian Liu, and Chuan Shi. Provable training for graph contrastive learning. In *Proceedings of the Advances in Neural Information Processing Systems*, 2023.

[21] 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*, 2024.

[22] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[23] Junlin He, Jinxiao Du, and Wei Ma. Preventing dimensional collapse in self-supervised learning via orthogonality regularization. *Advances in Neural Information Processing Systems*, 37:95579–95606, 2024.

[24] Jin Li, Qirong Zhang, Wenxi Liu, Antoni B. Chan, and Yang-Geng Fu. Another perspective of over-smoothing: Alleviating semantic over-smoothing in deep gnns. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6897–6910, 2025.

[25] Hugo Attali, Davide Buscaldi, and Nathalie Pernelle. Delaunay graph: Addressing over-squashing and over-smoothing using delaunay triangulation. In *Proceedings of the International Conference on Machine Learning*, 2024.

[26] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[27] Jianqing Liang, Zhiqiang Li, Xinkai Wei, Yuan Liu, and Zhiqiang Wang. Ml ^{2}-gcl: Manifold learning inspired lightweight graph contrastive learning. In *Proceedings of the International Conference on Machine Learning*, 2025.

[28] Jianqing Liang, Xinkai Wei, and Zhiqiang Li. Cl-gcl: Comprehensive and lightweight graph contrastive learning. In *Proceedings of the International Conference on Machine Learning*, 2026.

[29] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Proceedings of the Advances in Neural Information Processing Systems*, 2020.

[30] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *The International Conference on Learning Representations*, 2019.

[31] Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 5, pages 488–495, 2009.

[32] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

[33] Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.

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

[35] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B. Aditya Prakash. Sub2vec: Feature learning for subgraphs. In *Proceedings of the Advances in Knowledge Discovery and Data Mining - Pacific-Asia Conference*, volume 10938, pages 170–182, 2018.

[36] 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.

[37] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *The International Conference on Learning Representations*, 2020.

[38] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *Proceedings of the International Conference on Machine Learning*, volume 139, pages 12121–12132, 2021.

[39] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Proceedings of the Advances in Neural Information Processing Systems*, 2020.

[40] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *The International Conference on Learning Representations*, 2021.

[41] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In *Proceedings of the International Conference on Machine Learning*, volume 162, pages 24332–24346, 2022.

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

[43] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

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

[45] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. Graph debiased contrastive learning with joint representation clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3434–3440, 2021.

[46] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *Proceedings of the Advances in Neural Information Processing Systems*, 34:76–89, 2021.

[47] Shengzhong Zhang, Wenjie Yang, Xinyuan Cao, Hongwei Zhang, and Zengfeng Huang. Structcomp: Substituting propagation with structural compression in training graph contrastive learning. In *The International Conference on Learning Representations*, 2024.

[48] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610, 2007.

A Additional Experimental Details

A.1 Dataset Descriptions

Graph classification datasets. We consider seven widely-used graph classification datasets spanning bioinformatics and social network domains. MUTAG, DD, NCI1, and PROTEINS are classical bioinformatics benchmarks, where each graph corresponds to a molecule or protein structure. Nodes represent atoms or amino acids, and edges denote chemical bonds or protein interactions. The classification task is to predict molecular properties (e.g., mutagenicity) or protein families. COLLAB, REDDIT-B, and IMDB-B are social network benchmarks. In COLLAB, graphs represent ego-networks of researchers in different scientific fields. REDDIT-B contains graphs where nodes are users and edges represent interactions in online threads, with the goal of classifying discussion communities. IMDB-B consists of ego-networks of actors/actresses, where nodes denote performers and edges indicate co-appearances in movies, and the task is to classify graphs into movie genres. These datasets are diverse in scale and density: bioinformatics graphs are typically small and sparse but contain rich structural motifs, while social network graphs are larger, noisier, and exhibit high structural variability.

Node classification datasets. We evaluate on seven widely-used node classification benchmarks, covering citation networks and co-purchase networks. Cora, CiteSeer, PubMed, and DBLP are citation graphs, where nodes represent academic papers and edges denote citation relationships. Node features are bag-of-words representations of paper content, and the task is to classify papers into research topics. Amazon-Computers and Amazon-Photo are product co-purchase graphs, where nodes correspond to products and edges indicate that two products are frequently bought together. Node features are bag-of-words representations of product reviews, and labels denote product categories. Ogbn-Arxiv is a large-scale citation network constructed from all Computer Science papers on the arXiv platform. Each node represents an individual paper, and each directed edge corresponds to a citation relationship between two papers. These datasets vary significantly in terms of graph size, node degree distribution, and feature sparsity, providing a comprehensive testbed for evaluating the generalization and stability of MGRD under different structural and feature settings.

Table 6: Statistics of Datasets Used in Experiments

Graph-level Datasets					Node-level Datasets				
Dataset	**Task**	**Nodes**	**Edges**	**Graphs**	**Dataset**	**Task**	**Nodes**	**Edges**	**Graphs**
MUTAG	Graph	17.93	19.79	188	Cora	Node	2,708	5,429	1
DD	Graph	284.32	715.66	1,178	CiteSeer	Node	3,327	4,732	1
NCI1	Graph	29.87	32.30	4,110	PubMed	Node	19,717	44,338	1
PROTEINS	Graph	39.06	72.82	1,113	DBLP	Node	17,716	52,867	1
COLLAB	Graph	74.49	2,457.78	5,000	Computers	Node	13,752	245,861	1
REDDIT-B	Graph	429.63	497.75	2,000	Photo	Node	7,650	119,081	1
IMDB-B	Graph	19.77	96.53	1,000	Ogbn-Arxiv	Node	169,343	1,166,243	1

A.2 Evaluation Protocol and Implementation Details

Graph classification. For graph-level benchmarks, we integrate MGRD into GraphCL [29], which uses a GIN encoder [30] and the same augmentations as the original baseline. After unsupervised pretraining, we extract graph embeddings and train a linear SVM classifier. We adopt 10-fold cross-validation with 5 independent repeats, and report the mean accuracy with standard deviation.

Node classification. For node-level benchmarks, we integrate MGRD into GCA [8] and follow the evaluation protocol of DGI [18]. Specifically, node embeddings obtained from the pretrained model are fixed, and a logistic regression classifier is trained to predict node labels.

Implementation details. All models are implemented in PyTorch, and experiments are conducted on an NVIDIA RTX 3090 GPU. We keep the encoder architecture, training epochs, and data augmentations identical to those used in the baseline methods to ensure fair comparison. Hyperparameters such as learning rate, batch size, and weight decay are tuned within a fixed search range based on validation performance.

B Additional Experimental Results

B.1 Efficiency and Scalability Experimental Details

This section provides detailed empirical results on the computational efficiency and scalability of the proposed MGRD framework. Overall, MGRD incurs only marginal computational overhead without introducing additional message-passing steps. Table 7 summarizes the per-epoch training time, GPU memory usage, and classification accuracy of MGRD integrated into different graph contrastive learning backbones under identical training settings.

Table 7: Accuracy (%), time per epoch (s/epoch), and GPU memory usage for GCL backbones with and without the proposed MGRD plugin. For PubMed, memory is measured in MB, and for Ogbn-Arxiv in GB. OOM indicates out-of-memory on GPU.

Method	**PubMed**			**Ogbn-Arxiv**		
	Acc	Mem	Time	Acc	Mem	Time
GRACE	86.7	11677	0.252	–	OOM	–
GRACE + **MGRD**	87.5	11759	0.267	–	OOM	–
CCA-SSG	86.0	825	0.123	–	OOM	–
CCA-SSG + **MGRD**	86.8	859	0.132	–	OOM	–
StructComp	87.2	194	0.009	71.7	11.7	1.2
StructComp + **MGRD**	87.7	203	0.008	72.3	11.9	1.3

We evaluate the one-time cost of constructing the manifold prior on the graph compressed by StructComp. The reconstruction weights are computed once and kept fixed throughout training. Table 8 reports the preprocessing time and additional peak CPU memory, excluding the shared costs of METIS partitioning and graph compression. Total preprocessing time additionally includes CSR conversion, sparse-prior assembly, and function-call overhead.

Table 8: One-time manifold-prior construction cost after StructComp compression. Shared costs of METIS partitioning and graph compression are excluded.

Dataset	Number of nodes		Preprocessing time (s)			Peak CPU overhead (MiB)
	Original	Compressed	Neighborhood	Weights	Total	
PubMed	19,717	1,958	0.47	0.52	0.99	47.52
Ogbn-Arxiv	169,343	19,863	29.31	6.64	36.04	151.95

B.2 Additional Representation-Level Ablation

Table 9 reports representation-level ablation results on the remaining datasets, using the same settings as Table 4. The results exhibit the same pattern: the shared representation outperforms the complementary representation, while concatenating the two consistently achieves the highest accuracy.

Table 9: Representation-level ablation on the remaining benchmarks. Graph and node classification use GraphCL + MGRD and GCA + MGRD, respectively. Accuracy (%) is reported, with the best result per dataset in bold. Δ denotes the gain of Combined over Shared in percentage points (pp).

Representation	Graph classification			Node classification		
	DD	COLLAB	IMDB-B	CiteSeer	DBLP	Computers
Shared (Z^S)	76.5	68.7	71.6	72.1	84.5	87.4
Complementary (Z^C)	73.4	63.9	66.3	71.7	82.9	85.3
Combined ($[Z^S \| Z^C]$)	**81.5**	**72.8**	**73.5**	**73.5**	**85.7**	**88.7**
Δ (pp) $\uparrow$	+5.0	+4.1	+1.9	+1.4	+1.2	+1.3

C Supplementary Proofs

C.1 Proof of Proposition 3.1

We provide the detailed proof of the entropy preservation result in the main text. The final representation of MGRD is formed by concatenating the shared and complementary components:

$$\mathbf{Z} = [\mathbf{Z}^S \| \mathbf{Z}^C] \in \mathbb{R}^{N \times (d_S + d_C)}, \tag{14}$$

where $\mathbf{Z}^S \in \mathbb{R}^{N \times d_S}$ denotes the shared representation and $\mathbf{Z}^C \in \mathbb{R}^{N \times d_C}$ denotes the complementary representation. Let

$$\Sigma_S = \mathrm{Cov}(\mathbf{Z}^S), \qquad \Sigma_C = \mathrm{Cov}(\mathbf{Z}^C), \qquad \Sigma_{SC} = \mathrm{Cov}(\mathbf{Z}^S, \mathbf{Z}^C) \tag{15}$$

denote the covariance matrices of the shared subspace, complementary subspace, and their cross-covariance, respectively. The covariance matrix of the concatenated representation $\mathbf{Z}$ can then be written as the following block matrix:

$$\Sigma_Z = \begin{bmatrix} \Sigma_S & \Sigma_{SC} \\ \Sigma_{CS} & \Sigma_C \end{bmatrix}, \tag{16}$$

where $\Sigma_{CS} = \Sigma_{SC}^\top$.

We consider the effective subspaces of $\mathbf{Z}^S$ and $\mathbf{Z}^C$, where Σ_S and Σ_C are positive definite. Following the non-degeneracy condition in the main text, assume that there exist constants $\epsilon_S > 0$ and $\epsilon_C > 0$ such that

$$\lambda_{\min}(\Sigma_S) \geq \epsilon_S, \qquad \lambda_{\min}(\Sigma_C) \geq \epsilon_C. \tag{17}$$

We further assume that the normalized cross-covariance between the two subspaces is bounded:

$$\left\| \Sigma_S^{-\frac{1}{2}} \Sigma_{SC} \Sigma_C^{-\frac{1}{2}} \right\|_2 \leq \rho, \qquad 0 \leq \rho < 1. \tag{18}$$

By the block determinant identity, the determinant of Σ_Z can be decomposed as

$$\det(\Sigma_Z) = \det(\Sigma_S) \det\left(\Sigma_C - \Sigma_{CS}\Sigma_S^{-1}\Sigma_{SC}\right). \tag{19}$$

The second term is the Schur complement of Σ_S in Σ_Z.

Define the normalized cross-covariance matrix as

$$\mathbf{M} = \Sigma_S^{-\frac{1}{2}}\Sigma_{SC}\Sigma_C^{-\frac{1}{2}}. \tag{20}$$

Then we have

$$\Sigma_{CS}\Sigma_S^{-1}\Sigma_{SC} = \Sigma_C^{\frac{1}{2}}\mathbf{M}^\top\mathbf{M}\Sigma_C^{\frac{1}{2}}. \tag{21}$$

Therefore, the Schur complement can be rewritten as

$$\begin{aligned}\Sigma_C - \Sigma_{CS}\Sigma_S^{-1}\Sigma_{SC} &= \Sigma_C - \Sigma_C^{\frac{1}{2}}\mathbf{M}^\top\mathbf{M}\Sigma_C^{\frac{1}{2}} \\ &= \Sigma_C^{\frac{1}{2}}\left(\mathbf{I} - \mathbf{M}^\top\mathbf{M}\right)\Sigma_C^{\frac{1}{2}}.\end{aligned} \tag{22}$$

From Eq. (18), all singular values of $\mathbf{M}$ are upper-bounded by ρ. Let $\{s_k\}_{k=1}^r$ denote the non-zero singular values of $\mathbf{M}$, where $r \le m = \min(d_S, d_C)$. Then the eigenvalues of $\mathbf{M}^\top\mathbf{M}$ are $\{s_k^2\}_{k=1}^r$ together with possible zeros. Since $s_k \le \rho < 1$, we have

$$1 - s_k^2 \ge 1 - \rho^2 > 0. \tag{23}$$

Thus, $\mathbf{I} - \mathbf{M}^\top\mathbf{M}$ is positive definite on the effective subspace, and

$$\det\left(\mathbf{I} - \mathbf{M}^\top\mathbf{M}\right) = \prod_{k=1}^{r}(1 - s_k^2) \ge (1 - \rho^2)^m. \tag{24}$$

Here, $m = \min(d_S, d_C)$ is an upper bound on the number of nonzero singular values of M. Since $0 < 1 - \rho^2 \le 1$ and $r \le m$, using the exponent m yields a conservative lower bound on $\det(I - M^\top M)$.

Using Eq. (22), we obtain

$$\begin{aligned}\det\left(\Sigma_C - \Sigma_{CS}\Sigma_S^{-1}\Sigma_{SC}\right) &= \det\left(\Sigma_C^{\frac{1}{2}}\left(\mathbf{I} - \mathbf{M}^\top\mathbf{M}\right)\Sigma_C^{\frac{1}{2}}\right) \\ &= \det(\Sigma_C)\det\left(\mathbf{I} - \mathbf{M}^\top\mathbf{M}\right) \\ &\ge \det(\Sigma_C)(1 - \rho^2)^m.\end{aligned} \tag{25}$$

Next, from the eigenvalue lower bounds in Eq. (17), we have

$$\det(\Sigma_S) = \prod_{i=1}^{d_S}\lambda_i(\Sigma_S) \ge \epsilon_S^{d_S}, \qquad \det(\Sigma_C) = \prod_{i=1}^{d_C}\lambda_i(\Sigma_C) \ge \epsilon_C^{d_C}. \tag{26}$$

Substituting Eq. (25) and Eq. (26) into Eq. (19), we obtain

$$\det(\Sigma_Z) \ge \epsilon_S^{d_S}\epsilon_C^{d_C}(1 - \rho^2)^m. \tag{27}$$

Under the Gaussian assumption, the differential entropy of the final representation $\mathbf{Z}$ is

$$H(\mathbf{Z}) = \frac{1}{2}\log\left((2\pi e)^d \det(\Sigma_Z)\right), \qquad d = d_S + d_C. \tag{28}$$

Using the determinant lower bound in Eq. (27), we have

$$H(\mathbf{Z}) \ge \frac{1}{2}\log\left((2\pi e)^d \epsilon_S^{d_S}\epsilon_C^{d_C}(1 - \rho^2)^m\right). \tag{29}$$

This proves the proposition.

The result indicates that when the shared and complementary subspaces both preserve non-trivial variance and their cross-correlation is bounded, the concatenated representation maintains a finite entropy lower bound. Therefore, MGRD can alleviate dimensional collapse by preserving useful variation in both subspaces while reducing their redundancy.

Algorithm 1: Training procedure of MGRD

Input: Graph $\mathcal{G} = (X, A)$ with features X and adjacency A;
Contrastive backbone encoder f_θ;
Decoder g_ψ; Residual encoder h_ϕ.

Mining manifold structures:
for *each node* $i \in \{1, \ldots, N\}$ **do**
  Construct the K-hop topological neighborhood $\mathcal{N}_K(i)$ from A;
  Select $\mathcal{P}_i \subseteq \mathcal{N}_K(i)$ based on input X ;
  Solve LLE optimization: $w_i = \arg\min_w \|x_i - \sum_{j \in P_i} w_{ij} x_j\|_2^2 + \lambda \|w_i\|_2^2$;
  subject to $\sum_{j \in P_i} w_{ij} = 1,\ w_{ij} \geq 0$;
Cache the reconstruction weights and keep them fixed during training;

Training:
while *not converged* **do**
  Sample two augmentations (X_u, A_u) and (X_v, A_v) of $\mathcal{G}$;
  `// 1. Shared representations`
  $Z_u^S \leftarrow f_\theta(X_u, A_u),\ \ Z_v^S \leftarrow f_\theta(X_v, A_v)$;
  $\mathcal{L}_{gcl} \leftarrow$ InfoNCE(Z_u^S, Z_v^S);
  $\mathcal{L}_{\text{mpr}} \leftarrow Eq.\ (3)$;
  `// 2. Complementary representations`
  $\hat{X}_u \leftarrow g_\psi(Z_u^S),\ \ \hat{X}_v \leftarrow g_\psi(Z_v^S)$;
  $R_u \leftarrow X_u - \hat{X}_u,\ \ R_v \leftarrow X_v - \hat{X}_v$;
  $Z_u^C \leftarrow h_\phi(R_u),\ \ Z_v^C \leftarrow h_\phi(R_v)$;
  $\mathcal{L}_{acc} \leftarrow Eq.\ (7)$;
  `// 3. Orthogonality constraint`
  $\mathcal{L}_{orth} \leftarrow Eq.\ (8)$;
  `// 4. Total loss and update`
  $\mathcal{L}_{total} \leftarrow \mathcal{L}_{gcl} + \alpha \mathcal{L}_{mpr} + \beta \mathcal{L}_{acc} + \gamma \mathcal{L}_{orth}$;
  Update parameters θ, ψ, ϕ via gradient descent;

Inference on the original graph:
$\mathbf{Z}^S \leftarrow f_\theta(\mathbf{X}, \mathbf{A})$
$\mathbf{Z}^C \leftarrow h_\phi\big(\mathbf{X} - g_\psi(\mathbf{Z}^S)\big)$
$\mathbf{Z} \leftarrow [\mathbf{Z}^S \| \mathbf{Z}^C]$
Output: Node embeddings Z for downstream tasks.

D Algorithmic Description of MGRD

For completeness and reproducibility, we provide a detailed algorithmic description of the proposed MGRD framework in this appendix. Algorithm 1 summarizes the overall training procedure, including manifold structure mining via locally linear embedding, shared and complementary representation learning, and the integration of manifold prior, orthogonality regularization, and graph contrastive objectives.

E Limitations and Broader Impact

Limitations. Although MGRD consistently improves representative graph contrastive learning backbones, it still has several limitations. First, MGRD is built upon augmented graph views, and its effectiveness may be influenced by the quality of the underlying graph augmentations. Second, the proposed objective introduces additional regularization weights, including α, β, and γ, which require empirical selection for different datasets and backbones. Third, our experiments are mainly conducted on standard node-level and graph-level benchmark datasets. Further validation on larger, dynamic, heterogeneous, or domain-specific graphs remains an important direction for future work.

Broader Impact. MGRD is a general graph representation learning framework designed to improve the robustness and expressiveness of graph contrastive learning. Its potential positive impacts include improving representation learning for scientific graphs, citation networks, social networks, and other graph-structured data. However, like other graph learning methods, MGRD may inherit biases or sensitive information from the underlying graph data when applied to real-world networks. Therefore, deployment in high-stakes scenarios should carefully consider data quality, privacy, fairness, and domain-specific risks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main claims of MGRD, including manifold-guided representation disentanglement, shared and complementary subspaces, mitigation of dimensional collapse, and consistent improvements across GCL backbones. These claims are supported by the method, theoretical analysis, and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix E discusses the limitations of MGRD, including its dependence on graph augmentation quality, the need to tune additional regularization weights, and the need for further validation on larger, dynamic, heterogeneous, or domain-specific graphs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical analysis states the non-degeneracy and bounded cross-correlation assumptions for Proposition 3.1. A detailed proof of the entropy preservation result is provided in Appendix C.1.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the datasets, evaluation protocols, backbone models, training settings, and implementation details in Section 4.1 and Appendix A. Additional results, scalability analysis, and algorithmic details are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments are conducted on publicly available graph benchmark datasets. We provide an anonymized code package in the Supplementary Material, including implementation of MGRD, configuration files, running scripts, and instructions for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes the datasets, backbone models, evaluation protocols, hardware, and implementation details in Section 4.1 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main classification results are reported with mean accuracy and standard deviation. Graph classification uses 10-fold cross-validation with 5 independent repeats, and node classification results are averaged over 20 runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper states that all models are implemented in PyTorch and trained on an NVIDIA RTX 3090 GPU. Appendix B.1 further reports per-epoch training time, GPU memory usage, and accuracy for scalability evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies graph representation learning on public benchmark datasets and does not involve human subjects, private data, sensitive attributes, or high-risk deployment. The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix E discusses the potential positive impacts of MGRD in improving graph representation learning, as well as possible risks related to biased or sensitive graph data in real-world applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, generative models, or systems with obvious dual-use risks. The method is evaluated on standard public graph benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use publicly available graph benchmark datasets and properly cite their original creators and sources.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset, benchmark, or released model asset. The main contribution is a plug-and-play graph contrastive learning framework evaluated on existing public benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, interviews, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, crowdsourcing, or user studies, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not use LLMs as an important, original, or non-standard component. Any possible use of LLMs for writing, editing, or formatting does not affect the methodology, scientific rigor, or experimental results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.