Exploitation of a Latent Mechanism in Graph Contrastive Learning: Representation Scattering

Dongxiao He[1]**, Lianze Shan**[1]**, Jitao Zhao**[1]**, Hengrui Zhang**[2]**, Zhen Wang**[3]*, **Weixiong Zhang**[4]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, United States
[3]School of Cybersecurity, Northwestern Polytechnical University, Xi'an, China
[4]Department of Computing, Department of Health Technology and Informatics,
The Hong Kong Polytechnic University, Kowloon, Hong Kong
[1]{hedongxiao, shanlz2119, zjtao}@tju.edu.cn,
[2]hzhan55@uic.edu, [3]w-zhen@nwpu.edu.cn, [4]weixiong.zhang@polyu.edu.hk

Abstract

Graph Contrastive Learning (GCL) has emerged as a powerful approach for generating graph representations without the need for manual annotation. Most advanced GCL methods fall into three main frameworks: node discrimination, group discrimination, and bootstrapping schemes, all of which achieve comparable performance. However, the underlying mechanisms and factors that contribute to their effectiveness are not yet fully understood. In this paper, we revisit these frameworks and reveal a common mechanism—representation scattering—that significantly enhances their performance. Our discovery highlights an essential feature of GCL and unifies these seemingly disparate methods under the concept of representation scattering. To leverage this insight, we introduce Scattering Graph Representation Learning (SGRL), a novel framework that incorporates a new representation scattering mechanism designed to enhance representation diversity through a center-away strategy. Additionally, consider the interconnected nature of graphs, we develop a topology-based constraint mechanism that integrates graph structural properties with representation scattering to prevent excessive scattering. We extensively evaluate SGRL across various downstream tasks on benchmark datasets, demonstrating its efficacy and superiority over existing GCL methods. Our findings underscore the significance of representation scattering in GCL and provide a structured framework for harnessing this mechanism to advance graph representation learning. The code of SGRL is at https://github.com/hedongxiao-tju/SGRL.

1 Introduction

Graph Neural Networks (GNNs) have shown impressive performance across various fields, including social networks [1, 2], bioinformatics [3, 4], and fraud detection [5]. However, training GNNs typically requires large datasets with manually annotated labels, which can be both costly and labor-intensive [6]. This limitation restricts their broader application. To address this challenge, Graph Contrastive Learning (GCL) has attracted significant attention [7, 8, 9, 10], focusing on creating proxy tasks from the data itself to enable self-supervised training of GNN encoders [11]. Currently, most GCL research [6, 12, 13] is concentrated on enhancing one of the three main frameworks: InfoNCE-based (i.e., node discrimination) [14], DGI-like (i.e., group discrimination) [7], and BGRL-like (i.e., bootstrapping schemes) [10].

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

The three mainstream graph contrastive learning frameworks differ significantly, particularly in their approach to node-level tasks. InfoNCE-based methods prioritize node-level discrimination [8, 9, 12], treating an anchor node as a reference while considering all other nodes as negative samples to enhance the distinctiveness of each node's representation. In contrast, DGI-like methods adopt a group discrimination paradigm [6, 7, 15], viewing all nodes as positive samples from the same distribution and differentiating them from a noise distribution. BGRL-like methods [10, 13, 16] employ a bootstrapping scheme that eliminates the need for negative samples during training, focusing instead on aligning positive samples. Despite these differences, all three frameworks demonstrate comparable performance, leading us to conjecture **that they may share a common mechanism.** This idea is partially inspired by previous research in visual contrastive learning, which identified uniformity as a key factor in many contrastive learning methods [17]. However, uniformity, which focuses solely on the discriminative properties of node instances, cannot explain the shared underlying factors across these GCL frameworks, as further discussed in Appendix A.

We carried out a detailed analysis of the three GCL frameworks and discovered that representation scattering is a crucial common factor in their success (in Section 3). In Section 3, we provide a formal definition of representation scattering and examine how each framework achieves it. For the DGI framework, we analyze the distributions of original and noise data, proving that after GNN message passing [18, 19], the noise data distribution aligns with the mean distribution of the original data. This indicates that DGI's objective can be interpreted as distinguishing between the local semantics of nodes within the original graph and its mean, which correlates with representation scattering (Section 3.1). In the InfoNCE framework, existing work has demonstrated that the mechanism of negative sampling facilitates a process of achieving uniformity [17]. We extend this further by theoretically proving that the InfoNCE loss serves as an upper bound for representation scattering loss (Section 3.2). Regarding the BGRL framework, we explore the role of Batch Normalization [20] and its connection to representation scattering, showing that it acts as a specific instance of this concept. Notably, removing Batch Normalization significantly degrades BGRL's performance (Figure 2). Our findings confirm that representation scattering is a key mechanism present across all three mainstream GCL frameworks.

However, the existing GCL frameworks have not fully leveraged the latent mechanisms inherent in their designs and neglected the interconnected nature of graphs when implementing representation scattering. This oversight results in inefficiencies and reduced robustness. Firstly, generating augmented views or computing similarities between node pairs for representation scattering [7, 8, 9] incurs significant computational and memory overhead. Secondly, manually defined negative samples may result in the generation of numerous false negative samples, introducing noise [12] that can hinder model training. Lastly, those frameworks that rely solely on positive samples [10] implement representation scattering indirectly through Batch Normalization, which lacks explicit guidance for scattering and can lead to suboptimal performance. Addressing these challenges is crucial for enhancing the effectiveness of GCL methods.

To fully and effectively utilize representation scattering, we propose **S**cattering **G**raph **R**epresentation **L**earning (SGRL). Our approach introduces a Representation Scattering Mechanism (RSM) that embeds node representations into a designated hypersphere, positioning them away from the mean center. This method offers a direct algorithm for representation scattering compared to the existing GCL frameworks, eliminating biases introduced by manually defined negative samples. Additionally, we introduce a Topology-based Constraint Mechanism (TCM) that considers the interconnected nature of graphs. TCM aligns representations derived from structural information with scattered representations, thereby preserving topology information while facilitating scattering. Through these innovations, we aim to enhance the efficiency and robustness of GCL methods. We made the following contributions in this paper:

- We discovered a common representation scattering mechanism in GCLs and showed that the three mainstream GCL frameworks implicitly utilize this mechanism, and importantly can be unified under the concept to representation scattering.
- We showed that the existing methods do not fully exploit the inherent mechanism of representation scattering. We introduced the novel SGRL framework to integrate RSM and TCM, producing an adaptive scattering approach for model training.
- We experimented with various downstream tasks on benchmark datasets and demonstrated the effectiveness and efficiency of SGRL.

2 Preliminary

Graph Data. We define a graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges. The node feature matrix is denoted by $\mathbf{X} \in \mathbb{R}^{N \times D}$, where N is the number of nodes and D is the feature dimension. In addition, the adjacency matrix is indicated by $\mathbf{A} \in \mathbb{R}^{N \times N}$, formulated such that $\mathbf{A}_{ij} = 1$ when an edge (v_i, v_j) exists within the set $\mathcal{E}$, or $\mathbf{A}_{ij} = 0$, otherwise. The degree matrix is denoted as $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_N)$, where each element $d_i = \sum_{j \in \mathcal{V}} \mathbf{A}_{ij}$. The degree-normalized adjacency matrix with self-loops is represented as $\mathbf{A}_{\text{sym}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2}$.

Graph Contrastive Learning. Given a graph's attributes $\mathbf{X}$ and adjacency matrix $\mathbf{A}$, the objective of GCL is to train an encoder $f(\cdot)$ in a self-supervised fashion. The learned encoder $f(\cdot)$ can generate representations $\mathbf{H} = f(\mathbf{X}, \mathbf{A}), \mathbf{H} \in \mathbb{R}^{N \times K}$, which are both topologies decoupled and dense. These representations can be applied to many downstream tasks.

3 Representation Scattering in GCL

We now examine the shared elements that contribute to the effectiveness of popular Graph Contrastive Learning (GCL) frameworks. Upon revisiting three widely used baseline GCL frameworks, we find that they all inherently utilize the mechanism of representation scattering, which plays a crucial role in their success. Here, we formally define Representation Scattering:

Definition 1. *(Representation Scattering) In a d-dimensional embedding space $\mathbb{R}^d$ comprising n vectors organized into a matrix $\mathbf{V} \in \mathbb{R}^{n \times d}$, consider a subspace $\mathbb{S}^k$ $(1 \leq k \leq d)$ of $\mathbb{R}^d$ and a scatter center $\mathbf{c}$. Representation scattering is a process satisfying two constraints, (i) Center-Away Constraint: Node representations are encouraged to be distant from the scattered center $\mathbf{c}$, and (ii) Uniformity Constraint: Node representations are uniformly distributed over the subspace $\mathbb{S}^k$.*

According to Definition 1, achieving representation scattering requires identifying a scattered center $\mathbf{c}$ within the subspace $\mathbb{S}^k$, and simultaneously satisfying the Center-Away and Uniformity Constraints. We will study its relationship with the popular GCL frameworks in the following section.

3.1 DGI-like methods

DGI-like methods generate negative samples through random permutation of nodes. They employ a mutual information discriminator $\mathcal{D}(\cdot)$, which maximizes the mutual information between nodes and their source graphs to train the model [6, 7, 15]. Here, we show that the objective function of DGI is a special case of representation scattering. To facilitate the proof, we present the following assumption:

Assumption 1. *(a) The normalized propagation matrix $\tilde{\mathbf{A}}$ is defined as $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\hat{\mathbf{A}}$, where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. (b) DGI generates the corrupted graph by randomly shuffling the entities in the feature matrix $\mathbf{X}$, while keeping the adjacency matrix $\mathbf{A}$ unchanged. (c) The original data are class-balanced, i.e., for any classes k and j, num(k) = num(j).*

The following results do not strictly require Assumption 1 to be satisfied. Assumption 1 represents a common scenario and serves to simplify the proof. A discussion on the validity of Assumption 1, and proofs of the subsequent results without Assumption 1, can be found in Appendix C.

Theorem 1. *At the node level, minimizing the DGI loss is equivalent to maximizing the Jensen-Shannon (JS) divergence between the local semantic distribution in the original graph and its average distribution.*

Proof. Let p_{data} denote the distribution of all nodes in the original graph, characterized by a mean μ and variance σ^2. To analyze the local distribution of node v_i after its embedding aggregation by GNN encoders, we define p_i as the distribution of node v_i and its first-order neighbors with mean μ_i and variance σ_i^2. We make use of a conclusion in [6] and introduce the following lemma:

Lemma 2. *Minimizing the DGI loss, denoted as $\mathcal{L}_{DGI}$, equals to maximizing the Jensen-Shannon (JS) divergence between the distribution of the original graph $\mathcal{G}$ and the corrupted graph $\tilde{\mathcal{G}}$, i.e., Min$(\mathcal{L}_{DGI}) \Leftrightarrow$ Max$(JS(\mathcal{G} \parallel \tilde{\mathcal{G}}))$.*

We discuss Lemma 2 in Appendix D.1. Lemma 2 establishes the relationship between the DGI loss and the distributions of the original and corrupted graphs. To investigate the distribution of representations in original and corrupted graphs, we focus on the case of a single-layer GNN [21] and have the following formulation:

$$\mathbf{H} = \mathrm{GNN}(\mathbf{A}, \mathbf{X}) = \tilde{\mathbf{A}}\mathbf{X}\mathbf{W}', \mathbf{h}_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{x}_j \cdot \mathbf{W}' = \sum_{j \in \mathcal{N}_i} \frac{1}{d_i} \mathbf{x}_j \cdot \mathbf{W}', \tag{1}$$

where $\mathbf{W}' = \xi(\mathbf{W}), \mathbf{W} \in \mathbb{R}^{D \times K}$ with ξ being an activation function like ReLU [22] for ease of understanding, $\mathcal{N}_i$ denotes the set of first-order neighbors of node v_i, inclusive of v_i itself. In Eq. 1, $\forall j \in \mathcal{N}_i, \mathbf{x}_j \sim p_i(\mu_i, \sigma_i^2)$. For node v_i, subsequent to GNN message passing, we compute the mean and variance of the aggregated representation $\mathbf{h}_i$ as follows:

$$\begin{aligned} \mathbb{E}[\mathbf{h}_i] &= \mathbb{E}\left[\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{x}_j \cdot \mathbf{W}'\right] = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbb{E}[\mathbf{x}_j] \cdot \mathbf{W}' = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mu_i \cdot \mathbf{W}', \\ \mathrm{Var}(\mathbf{h}_i) &= \mathrm{Var}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{x}_j \cdot \mathbf{W}'\right) = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathrm{Var}(\mathbf{x}_j) \cdot \mathbf{W}'^2 = \sigma_i^2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^2 \cdot \mathbf{W}'^2. \end{aligned} \tag{2}$$

Given that $\tilde{\mathbf{A}}$ is a normalized propagation matrix with $\sum_{j \in \mathcal{N}(i)} \alpha_{ij} = 1$ for the aggregated representation $\mathbf{h}_i$ in the original graph, the mean of representation $\mathbf{h}_i$'s distribution equals $\mu_i \cdot \mathbf{W}'$. We then analyze the distribution of the aggregated representation $\hat{\mathbf{h}}_i$ in the corrupted graph. Based on Assumption 1, the adjacency matrix $\mathbf{A}$ is unaltered and the feature vector $\tilde{\mathbf{x}}_i$ is selected randomly from the feature matrix $\mathbf{X}$ of the original graph, following the distribution $p_{\text{data}}(\mu, \sigma^2)$. According to Eq.2, the mean of the distribution of $\hat{\mathbf{h}}_i$ is $\mu \cdot \mathbf{W}'$, and the variance is $\sigma^2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^2 \cdot \mathbf{W}'^2$. Consequently, we have $\mathbf{h}_i \sim p_i'(\mu_i \cdot \mathbf{W}', \sigma_i^2 \cdot \mathbf{W}'^2)$ and $\hat{\mathbf{h}}_i \sim p_{\text{data}}'(\mu \cdot \mathbf{W}', \sigma^2 \cdot \mathbf{W}'^2)$, which directly prove that maximizing the JS divergence between the original and corrupted graph distributions at the node level maximizes, in effect, the JS divergence between the local semantic distribution of node v_i and the mean distribution of the nodes in the original graph. □

Corollary 3. *Taking the mean of the original graph as the center* $\mathbf{c}$*, and the original representation space as a subspace* $\mathbb{S}^k$*, the objective of DGI can be described as follows: within the subspace* $\mathbb{S}^k$*, DGI increases the distance between the nodes of the original graph and its center* $\mathbf{c}$*, achieving the objective of representation scattering.*

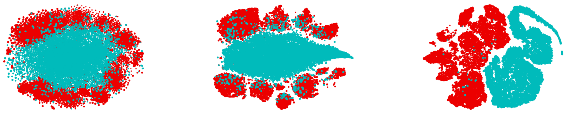

(a) Random Init (b) GNN-Layer #1 (c) GNN-Layer #2

Figure 1: t-SNE embedding of DGI on Co.CS dataset. The blue points represent the embeddings of the perturbed negative samples, and the red points denote that of positive nodes. As can be seen in Figures (a) and (b) of model random initialization and the layer of the trained encoder, the DGI-like methods essentially maximize the JS divergence between node embedding and embedding mean.

Corollary 3 reveals, for the first time, that the primary goal of DGI-like methods is to position node representations away from a central point to encourage a uniform distribution of the nodes. To illustrate these theoretical insights, we performed a visualization experiment. As shown in Figure 1 (a) and (b), both a randomly initialized GNN and a trained single-layer GNN demonstrated that their representations are distanced from the mean of the original graph nodes. Additionally, Figure 1 (b) and (c) provide clearer evidence that minimizing the $\mathcal{L}_{\text{DGI}}$ is equivalent to maximizing the Jensen-Shannon (JS) divergence between the positive and negative samples, i.e., maximizing the JS divergence between the local semantic distribution in the original graph and its mean distribution. However, the row-wise shuffling mechanism may introduce potential disturbances. In a graph with n nodes, each node v retains its local semantic distribution with a probability of $1/k$ during the shuffling process, where k denotes the number of distinct node types, assuming class balance. Despite this, the non-discriminatory nature of the perturbation means that these unchanged nodes could still be mistakenly classified as negative samples. Consequently, genuine node representations may be incorrectly labeled as negatives, leading to bias in the learning process. The overlapping nodes depicted in Figure 1 intuitively support this perspective.

3.2 InfoNCE-based methods

We now show that the mechanism of negative sampling in the InfoNCE-based methods is equivalent to representation scattering.

Theorem 4. *Let $\bar{\mathbf{h}} = 1/n \sum_{i=1}^{n} \mathbf{h}_i$, and sim$(\cdot)$ be the cosine similarity function (here, $\mathbf{h}_i$ represents the encoded representation of node v_i). For node v_i, the lower bound of the InfoNCE loss $\mathcal{L}_{InfoNCE}(\mathbf{h}_i)$ exists: $\mathcal{L}_{InfoNCE}(\mathbf{h}_i) \geq sim(\mathbf{h}_i, \bar{\mathbf{h}}) + \ln(2n)$.*

A detailed proof is given in Appendix D.3. Theorem 4 indicates that when minimizing $\mathcal{L}_{\text{InfoNCE}}(\mathbf{h}_i)$, the similarity between node v_i and mean node $\bar{v}$, i.e., $\text{sim}(\mathbf{h}_i, \bar{\mathbf{h}})$ is also minimized. Taking the mean of all nodes as the scattered center $\mathbf{c}$, and the hypersphere as the subspace $\mathbb{S}^k$, the mechanism of negative sampling is equivalent to representation scattering. However, the InfoNCE loss function is inefficient for representation scattering as it needs to compute and reduce the similarity of each negative pair. Moreover, it indiscriminately treats all negative samples, ignoring the distinctions among them, which leads to inappropriate scattering of negative samples and potential bias from false negatives. Consequently, many recent methods have incurred additional computational overhead by manually and intuitively defining positive and negative samples [12, 23].

3.3 BGRL-like methods

All BGRL-like methods incorporate a key component: Batch Normalization (BN). For a feature vector $\mathbf{x}_i$ and its corresponding batch statistics, mean μ and variance σ^2, the BN is applied as follows: $\text{BN}(\mathbf{x}_i) = \gamma(\mathbf{x}_i - \mu)/\sqrt{\sigma^2 + \epsilon} + \beta$, where γ and β are learnable parameters that scale and shift the normalized value, and ϵ is a small constant added for numerical stability.

Theorem 5. *The process of data normalization by batch normalization can be seen as a special case of representation scattering.*

A detailed proof can be found in Appendix D.2. To empirically assess the impact of batch normalization on BGRL, we conducted experiments comparing both BGRL and BGRL w/o BN across four benchmark datasets. The results, shown in Figure 2, reveal a significant drop in accuracy for BGRL across all datasets without batch normalization, highlighting its critical role in the bootstrapping framework. While BGRL incorporates representation scattering through batch normalization, its training process lacks explicit guidance or a dedicated mechanism to efficiently manage this scattering. The absence of direct supervision during the representation scattering phase can lead to an unoptimized distribution of representation within the embedding space, resulting in suboptimal performance.

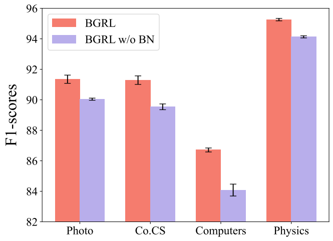


Figure 2: The impact of Batch Normalization in BGRL.

In summary, our analysis of the three mainstream GCL frameworks theoretically proved that they all inherently utilize representation scattering but also fail to fully utilize this effective mechanism. This analytic result has motivated us to design a more effective representation scattering method tailored for learning on graphs.

4 Methodology

The proceeding sections have revealed the importance of representation scattering in GCLs. Based on the findings, we have designed a novel method, namely Scattering Graph Representation Learning, short-handed as SGRL (Figure 3). We introduce the components of SGRL and provide discussion in the following sub-sections.

4.1 Representation Scattering Mechanism (RSM)

To address the shortcomings in the application of representation scattering within the three mainstream graph contrastive learning frameworks, we design RSM to explicitly guide the target encoder in learning scattered representations. Following Definition 1, we introduce a subspace $\mathbb{S}^k$ and a scattered

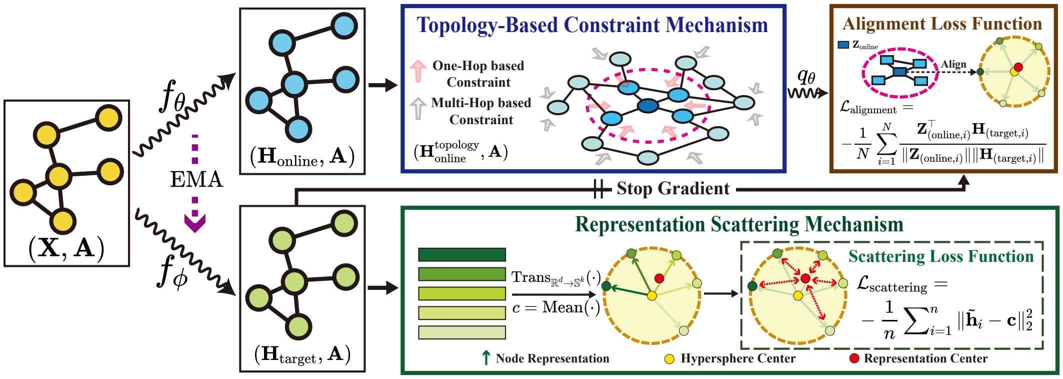

f_θ
f_ϕ
EMA
$(\mathbf{X}, \mathbf{A})$
$(\mathbf{H}_{\text{online}}, \mathbf{A})$
$(\mathbf{H}_{\text{target}}, \mathbf{A})$
Topology-Based Constraint Mechanism
One-Hop based Constraint
Multi-Hop based Constraint
$(\mathbf{H}_{\text{online}}^{\text{topology}}, \mathbf{A})$
q_θ
Alignment Loss Function
$\mathbf{Z}_{\text{online}}$
Align
$\mathcal{L}_{\text{alignment}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{Z}_{(\text{online},i)}^\top \mathbf{H}_{(\text{target},i)}}{\|\mathbf{Z}_{(\text{online},i)}\|\|\mathbf{H}_{(\text{target},i)}\|}$
Stop Gradient
Representation Scattering Mechanism
$\text{Trans}_{\mathbb{R}^d \to \mathbb{S}^k}(\cdot)$
$c = \text{Mean}(\cdot)$
Scattering Loss Function
$\mathcal{L}_{\text{scattering}} = -\frac{1}{n}\sum_{i=1}^{n}\|\tilde{\mathbf{h}}_i - \mathbf{c}\|_2^2$
Node Representation
Hypersphere Center
Representation Center

Figure 3: The overview of SGRL. Consider a graph $\mathcal{G}$ processed using two distinct encoders (online encoder and target encoder): $f_\theta(\cdot)$ with parameters θ and $f_\phi(\cdot)$ with ϕ, aimed at generating node representations $\mathbf{H}_{\text{online}}$ and $\mathbf{H}_{\text{target}}$, respectively. For $\mathbf{H}_{\text{target}}$, the mean representation of all nodes is calculated to serve as the scattered center $\mathbf{c}$. The parameters of $f_\phi(\cdot)$ are updated via RSM that encourages node representations to diverge from $\mathbf{c}$. $\mathbf{H}_{\text{online}}$ is processed through TCM to incorporate topology information, resulting in $\mathbf{H}_{\text{online}}^{\text{topology}}$. Subsequently, $\mathbf{H}_{\text{online}}^{\text{topology}}$ is embedded through a predictor q_θ to predict $\mathbf{H}_{\text{target}}$, and the parameters in $f_\theta(\cdot)$ is updated through back-propagation while stopping the gradient of $f_\phi(\cdot)$. Both channels are trained simultaneously. At the end of each epoch, we employ an Exponential Moving Average (EMA) to update parameters ϕ. Finally, the representations generated by $f_\theta(\cdot)$ are employed across various downstream tasks.

center $\mathbf{c}$ to effectively perform representation scattering. For the subspace $\mathbb{S}^k$, a transformation function $\text{Trans}(\cdot)$ is introduced to transform representations from the original space $\mathbb{R}^d$ into $\mathbb{S}^k$. Specifically, we apply ℓ_2 normalization to each row vector $\mathbf{h}_i$ in the matrix $\mathbf{H}_{\text{target}}$:

$$\tilde{\mathbf{h}}_i = \text{Trans}_{\mathbb{R}^d \to \mathbb{S}^k}(\mathbf{h}_i) = \frac{\mathbf{h}_i}{\text{Max}(\|\mathbf{h}_i\|_2, \varepsilon)}, \quad \mathbb{S}^k = \{\tilde{\mathbf{h}}_i : \|\tilde{\mathbf{h}}_i\|_2 = 1\}, \tag{3}$$

where $\mathbf{h}_i$ is representation for node $v_i \in \mathcal{V}$, generated by the target encoder, $\|\tilde{\mathbf{h}}_i\|_2 = (\sum_{j=1}^{k} \tilde{\mathbf{h}}_{ij}^2)^{\frac{1}{2}}$, and ϵ is a small value to avoid division by zero. As defined by Eq. 3, the representations of all nodes are distributed on a hypersphere $\mathbb{S}^k$. This mapping prevents arbitrary scattering of representations in the space, avoiding instability and optimization difficulties during training.

Next, we define the scattered center $\mathbf{c}$ and introduce a representation scattering loss function $\mathcal{L}_{\text{scattering}}$ in $\mathbb{S}^k$ to push node representations away from the center $\mathbf{c}$, as formulated as follows:

$$\mathcal{L}_{\text{scattering}} = -\frac{1}{n}\sum_{i=1}^{n}\|\tilde{\mathbf{h}}_i - \mathbf{c}\|_2^2, \quad \mathbf{c} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\mathbf{h}}_i. \tag{4}$$

Through Eq. 4, SGRL achieves uniformity of representations globally across the entire dataset, without emphasizing local uniformity. Specifically, RSM enables representations of different semantics to be globally scattered across the hypersphere while accommodating representations of the same semantics to aggregate locally.

Discussion of RSM. We now provide a theoretical analysis demonstrating that the proposed representation scattering mechanism outperforms the three graph contrastive learning frameworks. RSM achieves representation scattering more effectively by encouraging distances between node representations and the scattered center, eliminating the dependence on manually designed negative samples. Most traditional methods [7, 8, 9, 15] rely on negative samples to indirectly promote representation scattering, which is inefficient and introduces biases. DGI-like methods, as discussed in Section 3.1, aim to maximize the Jensen-Shannon (JS) divergence between the distribution of the original graph and its mean distribution. Based on this, generating additional noise graph is inefficient. Moreover, some negative samples generated through random shuffling may align with the distribution of positive samples, resulting in false negative samples that hinder model training. InfoNCE-based methods consider all nodes as negative samples, except the matching ones in two augmented views. While pushing nodes away from each other ensures the discriminability of each node, it also results in significant computational overhead. Moreover, due to the potential conflict between the encoder's message-passing mechanism and the InfoNCE loss function, many negative samples can not effectively distance themselves from each other [24]. Therefore, by employing a center-away strategy, RSM effectively reduces the additional computational overhead and mitigates the biases caused by manually designed negative samples.

4.2 Topology-based Constraint Mechanism(TCM)

After obtaining the scattered representations $\mathbf{H}_{\text{target}} = f_\phi(\mathbf{A}, \mathbf{X})$ through the target encoder, it is necessary to consider the differences in the degree of scattering of different node representations.

Consider the interconnected nature of graphs, the representations of two nodes connected topologically should be closer in space $\mathbb{S}^k$. Specifically, $\forall v_i, v_j \in V$, $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{S}^k$, given a threshold d: $\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 < d$, if $(i,j) \in \mathcal{V}$ and $\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 > d$, if $(i,j) \notin \mathcal{V}$. Intuitively, to this end, a simple way is to replace the individually scattered representations with the aggregated representations $\mathbf{H}_{\text{target}}^{\text{topology}}$ from first-order neighbors: $\mathbf{H}_{\text{target}}^{\text{topology}} = \mathbf{A}\mathbf{H}_{\text{target}}$. However, attempting to consider the topology information and achieve representation scattering through the same encoder may lead to conflicts.

To address this issue, we propose a Topology-based Constraint Mechanism (TCM). Specifically, we separate the process of constraint from the process of scattering by letting the online encoder generate topologically aggregated representations instead of the target. The online encoder enhances its representations by summing the original representations $\mathbf{H}_{\text{online}}$ with the topologically aggregated representations of its k-order neighbors $\hat{\mathbf{A}}^k\mathbf{H}_{\text{online}}$, which can be described as:

$$\mathbf{H}_{\text{online}}^{\text{topology}} = \hat{\mathbf{A}}^k\mathbf{H}_{\text{online}} + \mathbf{H}_{\text{online}}, \tag{5}$$

where k represents the order of neighbors and $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops. By separating scattering and constraints, SGRL can effectively achieve a balance between representation scattering and topology aggregation adaptively, rather than setting the scattering distance empirically. Next, the topology representations $\mathbf{H}_{\text{online}}^{\text{topology}}$ are fed into a predictor $q_\theta(\cdot)$ to generate the predicted representations $\mathbf{Z}_{\text{online}} = q_\theta(\mathbf{H}_{\text{online}}^{\text{topology}})$. Our objective is to align the predicted topology representations $\mathbf{Z}_{\text{online}}$ closely to the scattered representations $\mathbf{H}_{\text{target}}$, enhancing the model's effectiveness in capturing the essential semantic details of the graph. Based on this, the alignment loss $\mathcal{L}_{\text{alignment}}$ is defined as follows:

$$\mathcal{L}_{\text{alignment}} = -\frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{Z}_{(\text{online},i)}^\top \mathbf{H}_{(\text{target},i)}}{\|\mathbf{Z}_{(\text{online},i)}\|\|\mathbf{H}_{(\text{target},i)}\|}, \tag{6}$$

where $\mathbf{Z}_{\text{online}}$ and $\mathbf{H}_{\text{target}}$ represent the predicted and scattered representations, respectively. During this process, the online encoder's parameters θ are updated and the target encoder's parameters ϕ stop gradient propagation. Compared to directly aligning constrained and scattered representations, this predictive objective can act as a buffer, allowing the online encoder to adaptively learn scattered representations and topology information. Furthermore, to make the target encoder consider topological semantic information into the process of representation scattering, instead of solely focusing on scattering, we employ an Exponential Moving Average at the end of each training epoch:

$$\phi \leftarrow \tau\phi + (1-\tau)\theta, \tag{7}$$

where τ is a target decay rate and $\tau \in [0, 1]$. Eq. 7 effectively mitigates the adversarial interactions between RSM and TCM, while also facilitating the integration of topology information into the representation scattering process. Moreover, RSM enhances the discriminability of representations, while TCM incorporates topology information into the representations. The interaction between these two mechanisms effectively mimics the role of data augmentation, i.e., train encoders to learn the invariance of data to perturbations. Consequently, SGRL obviates the need for explicitly designing data augmentation strategies, which leads to additional computational overhead and heavy reliance on the choice of augmentation techniques.

5 Experiment

We evaluated SGRL on the five of the most widely used benchmark datasets, including Amazon-Photo (Photo) and Amazon-Computers (Computers) [25], WikiCS [26], Coauthor-CS (Co.CS) and Coauthor-Physics (Co.Physics) [27]. Detailed information of these datasets is in Appendix B.2. We compared SGRL with four types of methods: (1) Three mainstream baselines: GRACE [8], DGI [7], and BGRL [10]. (2)Six recently advanced algorithms: GCA [9], ProGCL [12], AFGRL [13], iGCL [28], GBT[29], MVGRL [15]. (3)Two classic graph representation learning methods: Node2vec [30] and Deepwalk [31]. (4)The semi-supervised training baseline GCN [18]. We utilized the representations generated by the online encoder for downstream tasks. For node classification, we followed the evaluate scheme from [10]. We trained a simple linear model using only the representations from a logistic regression loss with an ℓ_2 regularization and no backpropagation of any gradient to the graph encoder network. Specifically, we trained the downstream classifier using 10% of the data and tested the classifier on the remaining 90%. We ran SGRL 20 times and report here the average and standard deviation of the F1-score. For node clustering, we adopted the evaluation method from [13]. The testing was conducted on the learned representations at each epoch, and the best performance is reported below. More details of the experiments can be found in Appendix B.

Table 1: Performance on node classification. OOM signifies out-of-memory on 24GB RTX 3090. X, A, Y denote the node attributes, adjacency matrix, and labels in the datasets. Optimal results are shown in bold.

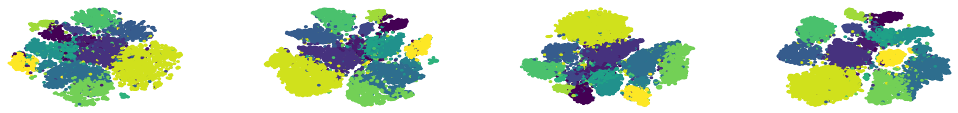
Method	**Available Data**	**WikiCS**	**Computers**	**Photo**	**Co.CS**	**Co.Physics**
Raw Features	X	71.98 ± 0.00	73.81 ± 0.00	78.53 ± 0.00	90.37 ± 0.00	93.58 ± 0.00
Node2vec	A	71.79 ± 0.05	84.39 ± 0.08	89.67 ± 0.12	85.08 ± 0.03	91.19 ± 0.04
DeepWalk	A	74.35 ± 0.06	85.68 ± 0.06	89.44 ± 0.11	84.61 ± 0.22	91.77 ± 0.15
GRACE	X, A	77.97 ± 0.63	86.50 ± 0.33	92.46 ± 0.18	92.17 ± 0.04	OOM
DGI	X, A	75.35 ± 0.14	83.95 ± 0.47	91.61 ± 0.22	92.15 ± 0.63	94.51 ± 0.52
BGRL	X, A	76.86 ± 0.74	89.69 ± 0.37	93.07 ± 0.38	92.59 ± 0.14	95.48 ± 0.08
GBT	X, A	76.65 ± 0.62	88.14 ± 0.33	92.63 ± 0.44	92.95 ± 0.17	95.07 ± 0.17
MVGRL	X, A	77.52 ± 0.08	87.52 ± 0.11	91.74 ± 0.07	92.11 ± 0.12	95.33 ± 0.03
GCA	X, A	77.94 ± 0.67	87.32 ± 0.50	92.39 ± 0.33	92.84 ± 0.15	OOM
ProGCL	X, A	78.45 ± 0.04	89.55 ± 0.16	93.64 ± 0.13	93.67 ± 0.12	OOM
AFGRL	X, A	77.62 ± 0.49	89.88 ± 0.33	93.22 ± 0.28	93.27 ± 0.17	95.69 ± 0.10
iGCL	X, A	78.83 ± 0.08	89.41 ± 0.06	93.02 ± 0.06	93.52 ± 0.04	94.77 ± 0.20
SGRL(Ours)	X, A	**79.40 ± 0.10**	**90.23 ± 0.03**	**93.95 ± 0.03**	**94.15 ± 0.04**	**96.23 ± 0.01**
Supervised GCN	X, A, Y	77.19 ± 0.12	86.51 ± 0.54	92.42 ± 0.22	93.03 ± 0.31	95.65 ± 0.16

5.1 Performance Analysis

Overall evaluation. Table 1 presents the averages and standard deviations of the F1-scores for all methods on the node classification task. Some statistics for the existing methods are reported from either their original papers or [13]. As shown in the table, the proposed SGRL exhibits superior performance across all five datasets tested, achieving the highest accuracy. Compared to the three baselines, BGRL, GRACE, and DGI, our new method outperforms them by 1.23%, 2.14%, and 3.26%, respectively, showing SGRL's effectiveness in more effectively exploiting representation scattering. Moreover, we observe that SGRL outperforms methods like GCA, AFGRL and iGCL, which focus on improving upon data augmentation. This demonstrates the validity of employing a graph contrastive framework based on representation scattering and topology aggregation, instead of using data augmentation. SGRL also outperforms the advanced negative sampling method, ProGCL. We attribute this to SGRL's representation scattering, which does not explicitly define negative samples, achieving better performance than manually defined negative samples. When evaluated on node clustering tasks using the scheme in [13], it can be observed that SGRL achieves the best or second-best accuracy on most datasets. Although SGRL attempts to make node representations scattered, which seems unsuitable for clustering tasks, it still manages to deliver competitive performance. This demonstrates that TCM can preserve the topology information of the original graph in the process of representation scattering, ensuring that while

Table 2: Performance on Clustering in terms of NMI and homogeneity. Optimal results are shown in bold and suboptimal results are underlined.

		GRACE	DGI	BGRL	**SGRL**
WikiCS	NMI	0.4282	**0.4312**	0.3969	0.4188
	Hom.	0.4423	**0.4498**	0.4156	0.4369
Amazon-Computers	NMI	0.4793	0.4630	0.5364	**0.5380**
	Hom.	0.5222	0.4836	**0.5869**	0.5705
Amazon-Photo	NMI	0.6513	0.5487	**0.6841**	0.6788
	Hom.	0.6657	0.5557	**0.7004**	0.6786
Co-CS	NMI	0.7562	0.7162	0.7732	**0.7961**
	Hom.	0.7909	0.7428	0.8041	**0.8216**
Co-Physics	NMI	OOM	0.6540	0.5568	**0.7232**
	Hom.	OOM	0.6868	0.6018	**0.7366**



(a) GRACE (b) DGI (c) BGRL (d) SGRL

Figure 4: t-SNE embeddings of nodes in CS dataset.

performing representation scattering, it maintains the local structure and global semantic consistency of the nodes.

Visualization. To more intuitively show the advantages of the representations learned by SGRL, we employ t-SNE [32] for visualizing the learned representations from the Co-CS dataset. Each point represents a node, with the color indicating the node's label. As shown in Figure 4, SGRL shows clearer inter-class boundaries compared to the other three methods, indicating that RSM achieves superior scattering. Moreover, we observe better intra-class clustering in SGRL. This highlights that SGRL, which does not require manually defined negative samples, prevents the inappropriate distancing of intra-class nodes. It achieves both global scattering and local semantic aggregation, demonstrating the effectiveness of adaptive scattering methods based on topology constraints.

Table 3: Ablation study on node classification. Optimal results are shown in bold.

Variant	WikiCS	Amazon-Computers	Amazon-Photo	Co-CS	Co-Physics
SGRL w/o RSM and TCM	76.86 ± 0.74	89.69 ± 0.37	93.07 ± 0.38	92.59 ± 0.14	95.48 ± 0.08
SGRL w/o TCM	78.55 ± 0.08	89.54 ± 0.10	93.58 ± 0.05	94.08 ± 0.03	96.19 ± 0.04
SGRL w/o EMA	79.36 ± 0.08	90.03 ± 0.07	93.92 ± 0.02	93.89 ± 0.07	96.16 ± 0.07
SGRL (Ours)	**79.40 ± 0.13**	**90.23 ± 0.03**	**93.95 ± 0.03**	**94.15 ± 0.04**	**96.23 ± 0.01**

5.2 Model Analysis

Hyperparamter Analysis. In this subsection, we investigate the sensitivity of the hyperparameter k in SGRL, as shown in Eq. 5. The parameter k represents the order of neighbors aggregated, which directly influences the strength of the topology constraints. In our experiments, we adjust k in the range $0, 1, 2, \cdots, 7$ to evaluate the impact of different constraint strengths on SGRL. The results are shown in in Figure 5: when $k = 0$, i.e., "SGRL w/o TCM", the lack of topology constraints leads to a decrease in model performance. As k increases, the constraint ability of TCM gradually strengthens, and the model performance exhibits a unimodal shape with respect to changes in k. This highlights that weak topology cannot preserve adequate topological information for adaptive representation scattering, whereas strong topology may excessively restrict the scattering of representations, which aligns with the perspectives proposed in Sections 3 and 4.

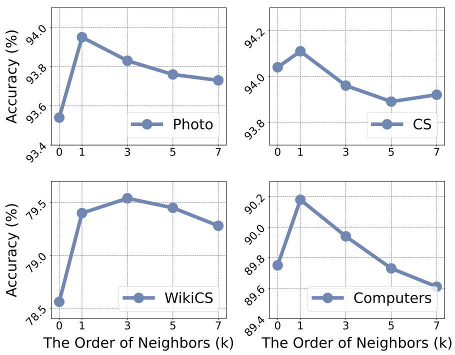


Figure 5: Hyperparameter Analysis on k

Ablation Studies. To verify the effectiveness of each component of SGRL, we conducted ablation studies on five datasets. As shown in Table 3, "SGRL w/o TCM and TCM" demonstrates significant performance improvements on four datasets with the addition of RSM alone, i.e., "SGRL w/o TCM", confirming its effectiveness. We observe that there is a slight performance decrease on the Computers dataset, which we attribute to over-scattering resulting from the absence of constraints during representation scattering. Additionally, a comparison between SGRL and "SGRL w/o TCM" reveals that SGRL achieves improved accuracy across all five datasets, particularly on the Computers dataset. This indicates that the TCM effectively constrain scattering, achieving an adaptive representation scattering, verifying our view in Section 4.2. Finally, we performed an ablation study on EMA. Given the potential adversarial relationship between RSM and TCM, we employed EMA to balance this effect. The experimental results in Table 3 highlight the necessity of incorporating EMA to mitigate the adversarial interaction between RSM and TCM.

6 Conclusion

In this paper, we made two significant contributions to Graph Contrastive Learning (GCL), an actively researched subject with numerous applications across diverse domains. First, through a comprehensive analysis of the three popular GCL frameworks, we discovered a common latent mechanism – representation scattering – that underlies these distinct contrastive methods. This discovery highlights an essential feature of GCL and unifies these seemingly disparate frameworks under the concept of representation scattering. Despite their popularity, the existing methods have not fully leveraged this mechanism, leaving the potential of representation scattering largely untaped. However, applying representation scattering directly to GCL poses technical challenges. Our second contribution addressed this issue by introducing the Representation Scattering Mechanism (RSM) and the Topology-based Constraint Mechanism (TCM), which we integrate into a novel GCL approach named Scattering Graph Representation Learning (SGRL). In SGRL, we effectively balance the adversarial relationship between RSM and TCM using an Exponential Moving Average (EMA) strategy. Extensive experimental results on benchmark datasets validate the effectiveness of our proposed method. Future work will further explore the broader implications of representation scattering beyond GCL, discussed in Appendices F and G.

Acknowledgement

This work was supported by the National Natural Science Foundation of China [No. 62422210, No. U22B2036, No. 62276187], the National Science Fund for Distinguished Young Scholarship (No. 62025602), the XPLORER PRIZE, the Hong Kong RGC theme-based Strategic Target Grant (STG1/M-501/23-N), the Hong Kong Global STEM Professorship Scheme, and the Hong Kong Jockey Club Charity Trust.

References

[1] Sanjay Kumar, Abhishek Mallik, Anavi Khetarpal, and Bhawani Sankar Panda. Influence maximization in social networks using graph embedding and graph neural network. *Inf. Sci.*, 607:1617–1636, 2022.

[2] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S Yu. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*, pages 2141–2145, 2021.

[3] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. Graph neural networks and their current applications in bioinformatics. *Frontiers in genetics*, 12:690049, 2021.

[4] Hai-Cheng Yi, Zhu-Hong You, De-Shuang Huang, and Chee Keong Kwoh. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1):bbab340, 2022.

[5] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1569–1572, 2020.

[6] Yizhen Zheng, Shirui Pan, Vincent Lee, Yu Zheng, and Philip S Yu. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, 35:10809–10820, 2022.

[7] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[8] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *CoRR*, abs/2006.04131, 2020.

[9] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2069–2080. ACM / IW3C2, 2021.

[10] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[11] Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z Li. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4216–4235, 2021.

[12] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 24332–24346. PMLR, 2022.

[13] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7372–7380. AAAI Press, 2022.

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

[15] Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4116–4126. PMLR, 2020.

[16] Wangbin Sun, Jintang Li, Liang Chen, Bingzhe Wu, Yatao Bian, and Zibin Zheng. Rethinking and simplifying bootstrapped graph latents. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 665–673, 2024.

[17] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[19] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[20] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021.

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[23] Hengrui Zhang, Qitian Wu, Yu Wang, Shaofeng Zhang, Junchi Yan, and Philip S Yu. Localized contrastive learning on graphs. *arXiv preprint arXiv:2212.04604*, 2022.

[24] Dongxiao He, Jitao Zhao, Cuiying Huo, Yongqi Huang, Yuxiao Huang, and Zhiyong Feng. A new mechanism for eliminating implicit conflict in graph contrastive learning. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 12340–12348. AAAI Press, 2024.

[25] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 43–52. ACM, 2015.

[26] Peter Mernyei and C Wiki-CS Cangea. A wikipedia-based benchmark for graph neural networks. arxiv 2020. *arXiv preprint arXiv:2007.02901*, 2007.

[27] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (MAS) and applications. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 243–246. ACM, 2015.

[28] Haifeng Li, Jun Cao, Jiawei Zhu, Qinyao Luo, Silu He, and Xuying Wang. Augmentation-free graph contrastive learning of invariant-discriminative representations. *IEEE Trans. Neural Networks Learn. Syst.*, 35(8):11157–11167, 2024.

[29] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V. Chawla. Graph barlow twins: A self-supervised representation learning framework for graphs. *Knowl. Based Syst.*, 256:109631, 2022.

[30] 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, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM, 2016.

[31] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710. ACM, 2014.

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

[35] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9:e15, 2020.

[36] Yuefeng Zhang. An introduction to matrix factorization and factorization machines in recommendation system, and beyond. *arXiv preprint arXiv:2203.11026*, 2022.

[37] Gregory F Lawler and Vlada Limic. *Random walk: a modern introduction*, volume 123. Cambridge University Press, 2010.

[38] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 385–394, 2017.

[39] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA*, pages 1024–1034, 2017.

[40] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

[41] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[42] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

A Discussion of Uniformity

In visual contrastive learning, uniformity is recognized as a critical factor contributing to the effectiveness of contrastive methods. In [17], uniformity is defined as the condition where feature vectors should be roughly uniformly distributed on the unit hypersphere, preserving as much information of the data as possible. This concept is based on the InfoNCE loss. Due to the similarities in implementation mechanisms between the SimCLR framework and the Deep InfoMax (DIM) framework, DIM can be considered a special case of SimCLR when the number of negative samples within a batch is limited to one. Consequently, this concept can be seen as a key factor contributing to the success of most visual contrastive learning methods.

However, this concept fails to adequately unify the frameworks within graph contrastive learning, particularly for DGI framework. In DGI, all nodes of the original graph are treated as a single distribution, whereas the distribution generated from a randomly shuffling noise graph is treated as another. The encoder is trained by maximizing the Jensen-Shannon (JS) divergence between these two distributions. First, the representations in DGI do not conform to a distribution on the unit hypersphere. Moreover, considering all nodes of the original graph as one distribution more closely aligns with a process of aggregation rather than facilitating a uniform distribution of nodes throughout the space.

Furthermore, while the negative sampling mechanism of InfoNCE-based methods implements uniformity, the interconnected nature of graphs can lead to conflicts during message passing within the encoder, which hinders the effective distancing of negative samples [24]. Therefore, we conducted a comprehensive theoretical analysis of three mainstream graph contrastive frameworks (Section 3) and introduced the concept of representation scattering. Specifically, the center-away constraint within representation scattering provides a more detailed elucidation of the operational mechanisms underlying the DGI framework than uniformity. Moreover, this constraint effectively mitigates the previously discussed conflicts, thereby facilitating the integration of representations and topological information and enabling adaptive representation scattering.

B Experiment Details

B.1 Hyperparameters Settings

Table 4: Detailed hyperparameters of SGRL.

Dataset	Hidden dim	online learning rate	target learning rate	Training epochs	Activation	momentum
WikiCS	1024	0.00001	0.00001	500	PReLU	0.99
Amazon-Computers	1024	0.001	0.001	700	PReLU	0.99
Amazon-Photo	1024	0.001	0.001	700	PReLU	0.99
Coauthor-CS	1024	0.001	0.001	700	PReLU	0.99
Coauthor-Physics	1024	0.0001	0.00001	1000	PReLU	0.99

B.2 Datasets

We use Wiki-CS, Amazon-Computers (Computers), Amazon-Photo (Photo), Coauthor-CS and Coauthor-Physics to evaluate the performance of SGRL. The detailed statistics of all the used datasets are in Table 5

Table 5: Statistics of datasets used in this paper.

Dataset	Type	Nodes	Edges	Attributes	Classes
Wiki-CS	reference	11,701	216,123	300	10
Amazon-Computers	co-purchase	13,381	245,778	767	10
Amazon-Photo	co-purchase	7,487	119,043	745	8
Coauthor-CS	co-author	18,333	81,894	6,805	15
Coauthor-Physics	co-author	34,493	247,962	8,415	5

Wiki-CS [26] is a directed graph dataset sourced from Wikipedia. It comprises nodes that represent articles in the field of computer science, with edges corresponding to the hyperlinks interconnecting these articles. Each article is categorized into one of 10 associated subfields. The attributes of the nodes are derived as the average of the text embeddings for the respective articles, making it the sole dataset characterized by dense attributes.

Amazon-Computers and Amazon-Photo [25] are networks that map out the co-purchase relationships among products on the Amazon platform. In these networks, nodes represent products, and edges are established between products that are frequently bought together. Products within each network are categorized into 10 and 8 distinct classes respectively, according to their product categories, and node features are represented using a bag-of-words model from the product reviews.

Coauthor-CS and Coauthor-Physics [27] represent two academic networks based on the Microsoft Academic Graph, illustrating the co-authorship links among academics. In these networks, nodes correspond to individual authors, while edges reflect collaborative authorship between them. Authors in each network are categorized into 15 and 5 research fields respectively, with node features encapsulated as a bag-of-words representation derived from the keywords of their publications.

B.3 Environment Configurations

All experiments were carried out on an NVIDIA GeForce GTX 3090 GPU, which comes equipped with 24GB of memory. For model development, we utilized PyTorch version 1.13.1 [33], along with PyTorch Geometric version 2.3.0 [34], which also served as the source for all the datasets used in our study. The detailed hyperparameters for node classification and clustering are shown in Table 4

B.4 Implementation Details

Our method employs GCN[18] as the encoder. The encoding process can be described as follows:

$$\mathbf{H}_k^{(l)} = \text{GCN}^{(l)}(\mathbf{H}_{k-1}^{(l)}, \mathbf{A}), \tag{8}$$

where $\mathbf{H}_k^{(l)}$ represents the node embedding matrix at the l-th layer during the k-th epoch of encoding and $\mathbf{H}_0^{(l)}$ is initialized by graph feature matrix $\mathbf{X}$,and $\mathbf{A}$ is the adjacency matrix without self-loops. In addition, the encoder structure is defined as:

$$\text{GCN}^{(l)}(\mathbf{H}_k^{(l)}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{H}_{k-1}^{(l)}\mathbf{W}_{k-1}^{(l)}), \tag{9}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\sigma(\cdot)$ is the PReLU activation function, $\hat{\mathbf{D}}$ is the degree matrix of $\hat{\mathbf{A}}$, and $\mathbf{W}^{(l)}$ is the trainable weight matrix for the l-th layer. In our experiments, we set $l = 1$.

C Detailed Discussion on Assumption 1

Assumption 1.(a) In assumption 1.(a), we define the normalized propagation matrix as $\tilde{A} = D^{-1}A$. When the encoder is GCN [18], assumption 1.(a) holds. Next, we will discuss whether assumption 1.(a) holds in Corollary 3 when the encoder is GAT [19]. The message-passing formula for GAT is similar to that of GCN. For node v_i, $\mathbf{h}_i = \text{GAT}(A, X) = \sum_{\text{neighbor } j} \alpha_{ij}\mathbf{x}_{ij} \cdot \mathbf{W}$, but the method for obtaining the propagation matrix is different:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{x}_i \| \mathbf{W}\mathbf{x}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{x}_i \| \mathbf{W}\mathbf{x}_k]))}, \tag{10}$$

where $\mathbf{a}$ is the weight vector used to compute the attention scores, $\| \cdot \|$ denotes the concatenation operation, $\mathbf{W}$ is the weight matrix, and LeakyReLU is the activation function. In the original graph, node v_i and its first-order neighbors follow a distribution with mean μ_i and variance σ_i^2, i.e., $\forall j \in \mathcal{N}_i, x_j \sim p_i(\mu_i, \sigma_i^2)$. Subsequent to GAT aggregation, the representation of node v_i, i.e., $\mathbf{h}_i$ follows a local semantic distribution with a mean given by $\mathbb{E}\left(\sum \alpha_{ij}x_{ij}\right) = \sum \alpha_{ij}\mu_i$ and a variance of $\sigma_i^2 \sum \alpha_{ij}^2$. This is consistent with the conclusion in Section 3.1. In the perturbed graph, the method for calculating the aggregated distribution differs from that of GCN. For GCN, the propagation matrix $\alpha_{ij} = 1/d_i$ is independent of the distribution of node v_i and its neighbors. In contrast, for GAT, as shown in Eq. 10, α_{ij} reflects the correlation between node v_i and its neighbor v_j. Therefore, in the perturbed graph, there are two distributions present among node v_i and its first-order neighbors: the local semantic distribution p_i and the original data distribution p_{data}. Let $a = \sum \alpha_{ik}$, where $x_k \sim p_i(\mu_i, \sigma_i^2)$. Let $b = \sum \alpha_{ik'}$, where $x_{k'} \sim p_{data}(\mu, \sigma^2)$. After aggregation through GAT, the mean of the distribution that node v_i follows is $a \cdot \mu_i + b \cdot \mu$. In this case, minimizing the DGI loss is equivalent to maximizing the local semantic distribution of node v_i with a mean of $a \cdot \mu_i + b \cdot \mu$ and a variance of $\sigma_i^2 \sum \alpha_{ik}^2 + \sigma^2 \sum \alpha_{ik'}^2$.

In our problem, we aim to find a subspace $\mathbb{S}^k$ and a scattered center $\mathbf{c}$ to achieve representation scattering, and this goal can be accomplished with either GCN or GAT. To this end, Corollary 3 also holds when the encoder is GAT.

Assumption 1.(b) For assumption 1.(b), there are two ways to generate a corrupted graph in DGI [7]: one is by randomly shuffling the feature matrix $\mathbf{X}$, and the other is by keeping $\mathbf{X}$ unchanged and shuffling the adjacency matrix $\mathbf{A}$. Here, we discuss whether Theorem 1 still holds when the assumption is the latter.

We have already demonstrated in Section 3.1 that, in the original graph $\mathcal{G}$, the aggregated features of node v_i follow a distribution with mean μ_i and variance $\sigma_i^2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^2$. In the corrupted graph $\tilde{G}$, randomly shuffling the adjacency matrix $\mathbf{A}$ is equivalent to randomly re-wiring n nodes, where $\sum_{i=1}^n \sum_{j=1}^n \mathbf{A}_{ij} = \sum_{i=1}^n \sum_{j=1}^n \hat{\mathbf{A}}_{ij}$. The first-order neighbors of node v_i still derive their features from the original graph distribution, i.e., $\hat{x}_{ij} \sim p_{\text{data}}(\mu, \sigma^2)$, and in the corrupted graph, the new topology still satisfies $\sum(\alpha_{ij}) = 1$. Therefore, the aggregated representation follows a distribution with mean μ and variance $\sigma^2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^2$, which is consistent with the conclusion derived from assumption (b).

Assumption 1.(c) In our theoretical analysis, we assume class balance, which is common in many theoretical proofs [7]. When the classes are imbalanced, the conclusion is similar to what we explained in assumption 1. (a): the scattered center changes from the mean to $a \cdot \mu_i + b \cdot \mu$, i.e., it leans towards the classes with a larger number of instances. This often occurs in long-tail distribution problems. We will further explore this aspect in our future work.

D Proof

D.1 Proof of Lemma 2

Proof. Here, we rewrite the proof from work [6] for ease of understanding Lemma 2. We replace the summary vector s with an all-ones vector. By removing the weight parameters $\mathbf{W}$ in the discriminator $\mathcal{D}$, we obtain:

$$\begin{aligned}
\mathcal{L}_{DGI} &= \frac{1}{2N} \left(\sum_{i=1}^{N} \log \mathcal{D}(\mathbf{h}_i, \mathbf{s}) + \log(1 - \mathcal{D}(\tilde{\mathbf{h}}_i, \mathbf{s})) \right) \\
&= \frac{1}{2N} \left(\sum_{i=1}^{N} \log(\mathbf{h}_i \cdot \mathbf{s}) + \log(1 - \tilde{\mathbf{h}}_i \cdot \mathbf{s}) \right) \\
&= \frac{1}{2N} \left(\sum_{i=1}^{N} \log(\text{sum}(\mathbf{h}_i)) + \log(1 - \text{sum}(\tilde{\mathbf{h}}_i)) \right),
\end{aligned} \tag{11}$$

where sum$(\cdot)$ is the summation function for every dimension in vector $\mathbf{h}_i$. As described by Eq. 11, minimizing DGI loss can be viewed as a binary classification problem distinguishing between sum(h_i) from the original graph and sum$(\tilde{h}_i)$ from the perturbed graph. Therefore, minimizing DGI loss is equivalent to maximizing the JS divergence between the original and perturbed graphs. □

D.2 Proof of Theorem 5

Proof. Consider the output of a single-layer Graph Neural Network (GNN) as $\mathbf{H} \in \mathbb{R}^{N \times K}$, where N represents the batch size and K denotes the dimension of representations. Each column $\mathbf{H}_j$ of $\mathbf{H}$ represents all samples for a particular feature, with mean μ_j and standard deviation σ_j.

Batch Normalization (BN) processes each feature dimension j independently by calculating the mean μ_j and variance σ_j^2, and then applying the following transformation:

$$\mathbf{H}'_j = \mathrm{BN}(\mathbf{H}_j) = \frac{\mathbf{H}_j - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}, \quad \mu_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{H}_{ij}, \tag{12}$$

where ϵ is a small value to prevent division by zero. The initial representations $\mathbf{H}$ are distributed on an arbitrary space characterized by a mean center $\mu = \{\mu_1, \ldots, \mu_K\}$. After batch normalization, the space of the transformed representations $\mathbf{H}'$ is standardized to a space with zero mean and unit variance. This transformation effectively constitutes a center-away displacement, i.e., each feature vector is shifted away from the center of the original distribution. Moreover, Batch Normalization ensures that each feature dimension exhibits a variance of unity, thereby facilitating a more uniform distribution across the transformed representations. Although the primary objective of batch normalization is not representation scattering, it can be inferred from the aforementioned analysis that BN's adjustment of data distribution indirectly facilitates a more uniform scattering of representations across space. □

D.3 Proof of Theorem 4

Proof. It is well-known that the InfoNCE loss (with $\ell 2$ norm) is enforcing the embeddings to be uniformly distributed in a hypersphere. It is exactly representation scattering. We begin with analysing the formula of the InfoNCE [14] loss:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{h}_i) = -\log(\frac{e^{\theta(\mathbf{h}_i, \mathbf{h}'_i)}}{e^{\theta(\mathbf{h}_i, \mathbf{h}'_i)} + \sum_{k=1, k\neq i}^{n}(e^{\theta(\mathbf{h}_i, \mathbf{h}_k)} + e^{\theta(\mathbf{h}_i, \mathbf{h}'_k)})}), \tag{13}$$

where $\mathbf{h}_i$ denotes the representation of node v_i, obtained from one augmented view of the original graph, and $\mathbf{h}'_i$ represents the representation of the corresponding node v'_i from the other augmented view. Here, $\theta(\mathbf{h}_i, \mathbf{h}'_i) = \mathrm{Sim}(\mathbf{h}_i, \mathbf{h}'_i)/\tau$, where $\mathrm{Sim}(\cdot)$ denotes the cosine similarity between node representations. Specifically, $\mathrm{Sim}(\mathbf{h}_i, \mathbf{h}'_i)$ is computed as the normalized dot product: $\mathrm{Sim}(\mathbf{h}_i, \mathbf{h}'_i) = (\mathbf{h}_i^\top \mathbf{h}'_i)/(\|\mathbf{h}_i\|\|\mathbf{h}'_i\|)$. Eq. 13 states that minimizing the loss function requires the maximization of similarity among positive samples and the simultaneous minimization of similarity among negative samples.

To simplify the proof, we define $\theta(\mathbf{h}_i, \mathbf{h}'_i)$ as the cosine similarity between node pairs, omitting the hyper-parameter τ. Furthermore, we replace the logarithm with the natural logarithm $\ln$ in Equation (13), yielding the expression below:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{h}_i) = -\theta(\mathbf{h}_i, \mathbf{h}'_i) + \ln \textstyle\sum_{j=1}^{n} \left(e^{\theta(\mathbf{h}_i, \mathbf{h}_j)} + e^{\theta(\mathbf{h}_i, \mathbf{h}'_j)}\right). \tag{14}$$

To facilitate a clearer analysis of Equation (14) and given the convex nature of the exponential function, we simplify the expression using Jensen's Inequality:

$$\frac{1}{2n}\textstyle\sum_{j=1}^{2n} \mathrm{e}^{\theta_j} \geq \mathrm{e}^{\frac{1}{2n}\sum_{j=1}^{2n}\theta_j}, \tag{15}$$

where θ_j represents the cosine similarity between the anchor node v_i and other nodes that form negative pairs. According to Eq. (15), the sum of exponentials can be transformed into the exponential of a sum, which can then cancel out with the logarithm, thereby simplifying the equation:

$$\mathcal{L}_{\text{InfoNCE}}(h_i) \geq \frac{1}{2n}\sum_{j=1}^{2n-1} \theta_j + \ln(2n). \tag{16}$$

In Eq. (16), it is demonstrated that the InfoNCE loss associated with the anchor node v_i serves as an upper bound to the sum of cosine similarities between v_i and all negative samples. To elucidate the relationship between representation scattering and InfoNCE loss more clearly, the following lemma is presented:

Lemma 6. *Let v_i be the anchor node, and let $\{v_1, v_2, v_3, \cdots, v_n\}$ be the set of nodes, excluding v_i. The average of the sum of cosine similarities between v_i and all other nodes in the set is equivalent to the cosine similarity between the ℓ_2-normalized vector of v_i and the ℓ_2-normalized mean vector obtained from all other nodes.*

The proof is provided in Appendix D.4. Lemma 6 demonstrates that the minimization of Eq. (13) for the anchor node v_i entails a concurrent reduction in the cosine similarity between the ℓ_2-normalized vector of v_i and the ℓ_2-normalized centroid of all other nodes. This process can be viewed as transforming the set of nodes onto a hypersphere with a radius of 1 and distancing the anchor node from the centroid of other nodes. We consider this hypersphere as subspace $\mathbb{S}^k$, and the centroid as the scattered center $\mathbf{c}$, which aligns with our definition of representation scattering. Thus, minimizing the InfoNCE loss effectively promotes representation scattering. □

D.4 Proof of Lemma 6

Proof. Given two vectors $\mathbf{v_i}$ and $\mathbf{v_j}$, their cosine similarity is defined as $\frac{\mathbf{v}_i\mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}$, which is the cosine of the angle between them

$$\sum_{j=1,j\neq i}^{2n-1} \theta_{ij} = \sum_{j=1,j\neq i}^{2n-1} \frac{\mathbf{v_i v_j}}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}. \tag{17}$$

For ℓ_2-normalized vectors $\hat{\mathbf{v}}_\mathbf{i} = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$, this is equivalent to their dot product: $\hat{\mathbf{v}}_\mathbf{i} \cdot \hat{\mathbf{v}}_\mathbf{j}$:

$$\sum_{j=1,j\neq i}^{2n-1} \frac{\mathbf{v_i v_j}}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|} = \sum_{j=1,j\neq i}^{2n-1} \hat{\mathbf{v}}_\mathbf{i} \cdot \hat{\mathbf{v}}_\mathbf{j}. \tag{18}$$

Since the dot product of vectors possesses the distributive property of multiplication:

$$\sum_{j=1,j\neq i}^{2n-1} \hat{\mathbf{v}}_\mathbf{i} \cdot \hat{\mathbf{v}}_\mathbf{j} = \hat{\mathbf{v}}_\mathbf{i} \cdot \sum_{j=1,j\neq i}^{2n-1} \hat{\mathbf{v}}_\mathbf{j}. \tag{19}$$

Finally, we compute the mean of Eq. 19 :

$$\frac{1}{2n} \sum_{j=1,j\neq i}^{2n-1} \theta_{ij} = \hat{\mathbf{v}}_\mathbf{i} \cdot \frac{1}{2n} \sum_{j=1,j\neq i}^{2n-1} \hat{\mathbf{v}}_\mathbf{j} = \hat{\mathbf{v}}_\mathbf{i} \cdot \mathbb{E}_{v_j \in \mathcal{V}, j\neq i}(\hat{\mathbf{v}}_\mathbf{j}). \tag{20}$$

As a consequence, the mean of the sum of cosine similarity between the anchor node v_i and all other nodes can be regarded as the expected similarity of the ℓ_2-normalized v_i with all other ℓ_2 normalized points. □

E Related Work

Graph Representation Learning (GRL) has attracted substantial attention in recent years due to the broader application of graph data across various real-world scenarios [35]. The primary goal of GRL is to embed graph data, which is inherently high-dimensional, sparse, and non-Euclidean, into a lower-dimensional, dense, Euclidean space to facilitate easier processing in downstream tasks. Early GRL works rely on Matrix Factorization [36] and Random Walk [37], failing to simultaneously utilize the structural and attribute information of graphs [38]. In recent years, Graph Neural Networks (GNNs) have achieved great success, such as GCN [18], GAT [19], GraphSAGE [39], which effectively extract graph attribute and structural information simultaneously through message passing and aggregation mechanisms. However, the training of GNNs requires a large amount of manually annotated labels, which is expensive and labor-intensive [6]. To overcome this limitation, Graph Contrastive Learnings (GCLs) have emerged as a promising and increasingly popular learning paradigm for handling unlabeled graph data.

Currently, there are three mainstream graph contrastive frameworks, each inspired by methods from visual contrastive learning. DGI [7], inspired by Deep InfoMax (DIM) [40], generates a noise distribution through random shuffling and learns node representations by maximizing the Jensen-Shannon (JS) divergence between the semantic distribution of the original graph and the noise distribution. GRACE [8], a representative method based on InfoNCE loss and inspired by SimCLR [41], generates two correlated graph views through augmentation techniques. This framework employs the InfoNCE loss to enhance the similarity of identical nodes across the two views while minimizing the similarity among different nodes. BGRL [10], inspired by BYOL [42], utilizes two distinct encoders: an online encoder and a target encoder. The online encoder is trained to predict the output of the target encoder, while the target encoder is updated by Exponential Moving Average (EMA).

Although the three mainstream graph contrastive learning frameworks: GRACE, DGI, and BGRL, have achieved notable success in addressing the challenges of unlabeled graph data, the underlying mechanisms driving their success have not been fully explored. Current research predominantly focuses on the implementation and optimization of these frameworks [6, 9, 12, 13], with insufficient understanding of their theoretical foundations and intrinsic principles. In visual contrastive learning, studies have revealed that "uniformity", which refers to the uniform distribution of representations in hypersphere, is a key factor contributing to improved generalization and discriminative power of models [17]. However, due to the non-Euclidean structure of graphs and the complex dependencies among nodes, this theory cannot be directly applied to graph contrastive learning. The unique characteristics of graphs necessitate a reevaluation and adaptation of the uniformity concept to develop a corresponding theoretical framework suitable for graphs. In this paper, we reanalyze the three mainstream graph contrastive learning frameworks and propose a new mechanism: representation scattering, which is prevalent and important in these frameworks. Furthermore, considering the interconnected nature of graphs, we design a constraint mechanism to achieve an adaptive representation scattering.

F A Detailed Discussion on the Limitations

- **Application of Heterogeneous Datasets.** In this work, we focused exclusively on homogeneous datasets and achieved satisfactory results. Our experiments demonstrated the effectiveness of the proposed topology-based constraint under the assumption of homogeneity. However, this constraint may face challenges when applied to heterogeneous datasets. Heterogeneous datasets often exhibit varying structures and characteristics, which could undermine the performance of our method. Consequently, addressing these challenges and adapting our approach to handle heterogeneous datasets effectively is a crucial direction for future research. We plan to investigate and refine our method to ensure its robustness and applicability across diverse and complex datasets.
- **The Methods of Balancing RSM and TCM.** RSM aims for the representations of nodes to be scattered from each other, while TCM prevents excessive scattering among nodes with the same semantics. These two mechanisms inherently have an adversarial relationship. We use EMA to balance this adversarial relationship, which is just one of many possible methods.

G Broader Impacts of SGRL

As a preliminary exploration in the field of GCL, SGRL has the following impacts:

- Uncovering potential success factors in GCL: SGRL explores three different mainstream graph contrastive methods and identifies their common underlying factors, providing a new perspective for future GCL research.
- Applications in many fields: Compared to existing methods, SGRL offers higher accuracy and can more efficiently process graph data. It can be widely applied in various fields such as social networks, biological networks, and recommendation systems.
- Negative social impacts: There are no negative social impacts foreseen.

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 in this paper accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA 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 NA 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: We discuss the limitations of the work in Appendix F

   Guidelines:

   - The answer NA 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: In this paper, all proofs of theorems are provided and all assumptions are clearly stated or referenced in the statement of any theorems.

Guidelines:

- The answer NA 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: All the results in this paper can be reproduced.

   Guidelines:

   - The answer NA 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: We provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specify all the training and test details necessary to understand the results.

   Guidelines:

   - The answer NA 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: We report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments in this paper.

   Guidelines:

   - The answer NA 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: For each experiment, the paper provide sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA 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 research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: The paper discuss both potential positive societal impacts and negative societal impacts of the work performed in Appendix G.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: The creators or original owners of assets used in the paper are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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.