Towards Expansive and Adaptive Hard Negative Mining: Graph Contrastive Learning via Subspace Preserving

Zhezheng Hao
Northwestern Polytechnical University
haozhezheng@outlook.com

Haonan Xin
Northwestern Polytechnical University
noah.haonanxin@gmail.com

Long Wei
Northwestern Polytechnical University
longwei@mail.nwpu.edu.cn

Liaoyuan Tang
Northwestern Polytechnical University
tangly@mail.nwpu.edu.cn

Rong Wang
Northwestern Polytechnical University
wangrong07@tsinghua.org.cn

Feiping Nie*
Northwestern Polytechnical University
feipingnie@gmail.com

ABSTRACT

Graph Neural Networks (GNNs) have emerged as the predominant approach for analyzing graph data on the web and beyond. Contrastive learning (CL), a self-supervised paradigm, not only mitigates reliance on annotations but also has potential in performance. The hard negative sampling strategy that benefits CL in other domains proves ineffective in the context of Graph Contrastive Learning (GCL) due to the message passing mechanism. Embracing the subspace hypothesis in clustering, we propose a method towards expansive and adaptive hard negative mining, referred to as Graph contRastive leArning via subsPace prEserving (GRAPE). Beyond homophily, we argue that false negatives are prevalent over an expansive range and exploring them confers benefits upon GCL. Diverging from existing neighbor-based methods, our method seeks to mine long-range hard negatives throughout subspace, where message passing is conceived as interactions between subspaces. Additionally, our method adaptively scales the hard negatives set through subspace preservation during training. In practice, we develop two schemes to enhance GCL that are pluggable into existing GCL frameworks. The underlying mechanisms are analyzed and the connections to related methods are investigated. Comprehensive experiments demonstrate that our method outperforms across diverse graph datasets and remains competitive across varied application scenarios[1].

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Computing methodologies** → **Learning latent representations**; *Neural networks*; • **Mathematics of computing** → *Graph algorithms*.

---

*Feiping Nie is the corresponding author.
[1]Our code is available at https://github.com/zz-haooo/WWW24-GRAPE.

KEYWORDS

Graph neural networks, Graph contrastive learning, Hard negative mining, Web data mining

1 INTRODUCTION

Graph data is ubiquitous in both real-world and virtual realms, encompassing a broad spectrum of areas such as social networks, molecular structures, trade circulation. Recently, GNNs have witnessed significant strides in analyzing graph data, exhibiting exceptional performance in tasks such as graph classification, node clustering, link prediction, graph generation, etc. Following the pioneering contributions of GCN [27], GraphSAGE [17], GAT [64], etc., numerous GNN architectures have been developed and enhanced. Almost all GNNs are built upon the message passing mechanism between neighbors, where each node acquires feature information from its neighbors and contributes its own feature information. Analogous to most neural networks, GNNs are typically trained in a supervised manner and require an abundance of annotations.

Contrastive Learning (CL), as the avant-garde framework for self-supervised methods, has recently demonstrated a series of state-of-the-art performances in various domains [6, 7, 12, 40, 94]. These studies emphasize that the representations learned by CL perform comparably to supervised learning in downstream tasks. The essence of CL lies in learning representations that retain invariance under a variety of distortions, referred to as "data augmentations" [62, 63]. To achieve this, researchers develop InfoNCE objective [15, 49], which maximizes a lower bound of mutual information between augmented views [2, 20]. The core conception is to draw positive pairs closer while repelling negative pairs apart [16].

The breakthroughs of CL in computer vision have motivated studies to extend the analogous concepts from visual representation learning to graph data, referred to as Graph Contrastive Learning (GCL). These GCL methods achieve sota in both graph-level and node-level tasks [18, 61, 75, 87, 95, 96]. GCL adheres to the typical CL
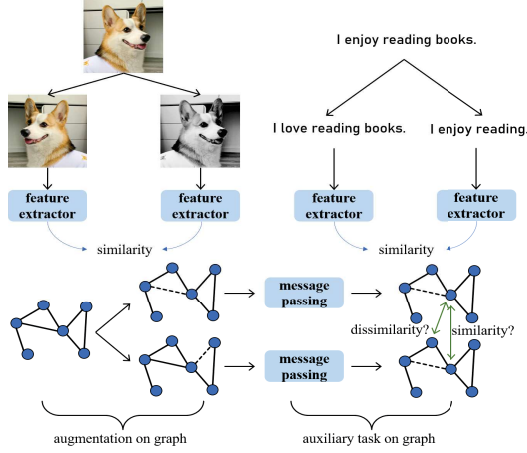
Figure 1: A comparison of CL for visual, textual, and graph data. The irregularity of graph data and the message passing mechanism of GNNs distinguish GCL from CL in other domains. Graph convolutional operator introduces smoothing property among neighbors, while necessitating some technical changes to GCL.

paradigm, albeit with specific variations [57, 65]. In general, the application paradigms of CL in visual, textual, and graph data domains can be illustrated as Figure 1. As demonstrated, existing research in GCL can be summarized into the following two main threads: (1) augmentation on graph [23, 32, 53, 59, 75, 84–86, 89, 93, 97], which aims to adapt semantic-preserving augmentation techniques from visual data to irregular graph data. (2) auxiliary task on graph [18, 33, 58, 76, 87, 89, 96], which explores the loss functions suitable for GNN training within CL framework. Our work falls into the latter category. Unlike other mainstream instance-discriminating backbones where instances do not exhibit explicit interactions, GNNs rely on message passing among instances. A notable issue arises where hard negative sampling techniques, proven to contribute in CL [5, 24, 31, 48, 54, 74], does not confer benefits in GCL and may even impair performance, which has been discussed in [41, 58, 76, 95]. The main concept behind is that hard negatives in GCL are prone to being false negatives, consequently, pushing away the semantically similar representations leads to a degradation in performance.

In this paper, we report that mining **expansive** and **adaptive** hard negatives enhances node-level tasks. To achieve both objectives, we introduce a negative hardness estimation scheme for GCL, aligning with the subspace preservation hypothesis in clustering. The core strength of our method lies in its ability to capture hard negatives beyond the scope of message passing and adjust the hard negatives set in a self-scaled manner. In node-level tasks, the concept of subspace preservation is intuitive. For instance, in a citation network, it can be elucidated as follows: from the semantic perspective, articles with the similar theme tends to share keywords (features); from the structural perspective, mutual citations within the same subfield are frequent whereas cross-domain article citations are limited. Prominent recommendation mechanisms within social or e-commerce networks, which curate personalized content for individual entities, have catalyzed the emergence of subspaces [52, 72, 73]. We provide theoretical and experimental analyses to

illuminate why and how our method works. To the best of our knowledge, our work is the first to address the GCL through subspace techniques.

In summary, the main contributions of this paper can be encapsulate in threefold:

- We show that more expansive and adaptive hard negative mining is promising for enhancing node-level GCL. Embracing this philosophy, we propose GRAPE, a negative hardness estimation method for GCL based on subspace theory.
- In GRAPE, the hard negatives beyond the scope of message passing can be captured and the hard negatives set can be adaptively scaled. Two schemes are devised to alleviate the influence of false negative samples on GCL. Besides, we provide a theoretical exposition of GRAPE's properties and uncover its connection with related methods.
- In comparison to several advanced GCL methods, GRAPE exhibits superior performance on eight widely-used public graph datasets. We conduct comprehensive experiments under various settings to thoroughly analyze the results and behaviors of GRAPE.

The proofs of involved theorems, experimental settings and supplementary experiments are relegated to the appendix.

2 RELATED WORK

In line with the focus of our work, we provide an overview of related works on graph contrastive learning and subspace preserving.

2.1 Graph Contrastive Learning

Amidst the increasing recognition of contrastive learning's expressive capability, DGI [65] and InfoGraph [57] first leverage the maximization of mutual information [20] at the node- and graph-level, respectively, to attain effective representations. In subsequent works, MVGRL [18] utilizes graph diffusion [13] to obtain augmented views and applies contrastive learning at both the node and graph levels. GMI [51] extends mutual information computations from vector spaces to the graph domain and assesses the correlation between input graphs and high-level hidden representations. GRACE [96], GCA [97] employ the InfoNCE-style objective and obtain node representations by treating others as negative samples, which serves as a baseline in follow-up research. To mitigate the sampling bias issue, BGRL [61] extends the BYOL [14] framework to graph. In this strand, CCA-SSG [87] optimizes a feature-level objective derived from classical canonical correlation analysis. SpCo [37] is introduced as a spectral GCL module based on the general graph augmentation rule to enhance existing GCL methods. In another thread, ProGCL [76] estimates the probability of a true negative using a two-component beta mixture model. Empirical studies [95] verify that assigning higher weights to hard negatives or generating hard negatives fails to improve GCL. GDCL [91] jointly performs GCL and DEC [78]; nevertheless, this unsupervised process may lead to training collapse. COSTA [89] advocates generating covariance-preserving augmented features inspired by matrix sketching. HomoGCL [33] proposes utilizing the homophily in graph to filter positive pairs. PHASES [58] employs a progressive negative masking strategy to enhance tolerance between sample pairs. We recommend readers to refer to [39, 77, 79] for a comprehensive overview.

2.2 Subspace Preserving

One underlying tenet in machine learning is that the data contains certain type of structure for intelligent representation. From this, the subspace assumption, which runs through the research journey of machine learning, can be described as follows [35]: high-dimensional data is drawn from a union of multiple affine or linear subspaces. In a simplified perspective, affine subspace is more closely related to manifold learning [3, 19, 47, 55, 60, 82], whereas linear subspace aligns more closely with dictionary learning [22, 30, 44, 70, 71]. Over the past decade, subspace learning based on the self-expression model, which enjoys the benefits of the both, has made significant strides [10, 34, 35, 42]. The main divergence among these methods is the constraints imposed on the self-expression coefficients, such as sparse constraint [10, 50, 69], low-rank constraint [25, 35, 36], connectivity constraint [43, 81, 83], and smooth constraint [4, 21, 29]. We adopt the fundamental principles of such methods to tackle hard negative mining in GCL. Both empirical investigations and theoretical analyses confirm the suitability in the context of GCL. Recent studies in graph dictionary learning [38, 67, 80] focus on sparse encoding for molecules, which are not directly related to our work.

3 METHODOLOGY

3.1 Notations and Preliminaries

Let $G = (A, X)$ denotes a graph with n nodes, where $A \in \{0, 1\}^{n \times n}$ denotes the adjacency matrix and $X \in \mathbb{R}^{n \times d}$ denotes the feature matrix. Let $\hat{A} = A + I_n$ be the adjacency matrix with self-loops. The normalized adjacency matrix is then given by $\tilde{A} = D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}}$, where D is the degree matrix. Vectors and matrices in this paper are denoted by bold lowercase and bold uppercase letters, respectively.

Our objective is to train a GNN encoder $f_\Theta(A, X)$ in a label-scarce scenario, where Θ represents the network parameters. The output node embeddings are supposed to be directly applicable for downstream tasks, such as node classification and node clustering in this paper. Take GCN for example, the layer-wise forward-propagation operation at the l-th layer is formulated as:

$$Z^{(l)} = \sigma\left(\tilde{A} Z^{(l-1)} W^{(l)}\right), \quad (1)$$

where W_l is the trainable weights for feature transformation and Z_l denotes the node embeddings at the l-th layer. Clearly, there is $Z^{(0)} = X$ at the initial layer. $\sigma(\cdot)$ denotes an activation function such as ReLU. In context of GCL, two views $G_1 = (A_1, X_1)$, $G_2 = (A_2, X_2)$ are generated by augmentation strategies [92] each epoch. G_1 and G_2 are fed into a Siamese GNN encoder [7] to produce node embeddings $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^n$, respectively. The contrastive loss can then be computed, followed by backpropagation. The common baseline for graph contrastive loss is the InfoNCE-style loss in GRACE [96]. Specifically, the contrastive loss for u_i is defined as:

$$\ell(u_i) =$$

$$-\log \frac{e^{\theta(u_i, v_i)/\tau}}{\underbrace{e^{\theta(u_i, v_i)/\tau}}_{\text{positive pair}} + \underbrace{\sum_{j \neq i} e^{\theta(u_i, v_j)/\tau}}_{\text{inter-view negative pairs}} + \underbrace{\sum_{j \neq i} e^{\theta(u_i, u_j)/\tau}}_{\text{intra-view negative pairs}}}, \quad (2)$$

where $\theta(\cdot, \cdot)$ denotes cosine similarity and τ is the temperature parameter. The objective $\ell(v_i)$ is defined symmetrically. Then the overall loss is given as:

$$\mathcal{L} = \frac{1}{2n} \sum_{i=1}^n \left(\ell(u_i) + \ell(v_i)\right). \quad (3)$$

It can be observed in Eq. (2) that by minimizing loss $\mathcal{L}$, embeddings of the same sample under two augmentations are pulled closer (**positives**), while embeddings of different samples are repelled away (**negatives**). For simplicity, in this paper, we represent $\ell(u_i)$ in the following form

$$\ell(u_i) = -\log \frac{\text{pos}(u_i)}{\text{pos}(u_i) + \text{neg}(u_i)}. \quad (4)$$

3.2 An Empirical Investigation for GCL

Recent studies [76, 95] report that considering all samples other than the anchor itself as negatives (Eq. (2)) unduly distances **false negatives** (i.e., samples of the same class as the anchor). This so-called "class collisions" phenomenon makes the marriage of CL and GNNs seem subtle and, as a result, leads to performance deterioration. Hard negative mining provides a remedy to rectify this deficiency, where **hard negatives** refer to negatives that are most similar to the anchor (possibly false negatives). Let Φ_i be the hard negatives set of the i-th sample. With $\{\Phi_i\}_{i=1}^n$ identified, hard negative mining mainly employs two forms of loss: one explicitly treats Φ_i as positives [9, 33](referred to as "Positive" strategy), i.e., modifying the "pos" term in Eq. (4) to:

$$\text{pos}(u_i) = e^{\theta(u_i, v_i)/\tau} + \sum_{j \in \Phi_i} e^{\theta(u_i, v_j)/\tau} + \sum_{j \in \Phi_i} e^{\theta(u_i, u_j)/\tau}; \quad (5)$$

another strategy masks Φ_i within the negatives [8, 76] (referred to as "Mask" strategy), i.e., modifying the "neg" term in Eq. (4) to:

$$\text{neg}(u_i) = e^{\theta(u_i, v_i)/\tau} + \sum_{j \notin \Phi_i} e^{\theta(u_i, v_j)/\tau} + \sum_{j \notin \Phi_i} e^{\theta(u_i, u_j)/\tau}. \quad (6)$$

While both strategies are intuitive, their effectiveness on graphs remains to be thoroughly explored. We employ GRACE with two-layer GCN as a baseline and probe the quality of hard negative mining on three datasets. The results are shown in Table 1, with each value representing the average of 10 repeated runs. The different settings are explained as follows: "*w/o MP*" denotes GRACE without message passing, "*x-hop*" denotes randomly selecting neighbors within x-hop as hard negatives, "*x-hop**" denotes randomly selecting false negatives (labels available) within x-hop as hard negatives, and "*all-hop**" denotes randomly selecting hard negatives from all false negatives. Here, 1-hop includes own neighbors for each node, while 2-hop encompasses the neighbors of neighbors and so forth. The "all-hop*" setting is an extreme scenario with all labels available in which $\ell(u_i)$ is akin to the tuplet loss in metric learning [26, 56]. To prevent label leakage, the number of selected hard negatives in all settings are equal.

Through empirical study, we make the following observations: (1) the performance of GCL significantly deteriorates in the absence of message passing; (2) "*x-hop*" settings provide limited benefits and occasionally detrimental to GCL; (3) training under "*x-hop**" setting improves GCL performance; (4) "*x-hop**" setting with larger x leads to more noticeable performance improvements. A follow-up query

Table 1: Empirical study (node classification accuracy in percentage) on hard negative mining in GCL.

Datasets	Cora		CiteSeer		PubMed	
Settings	Positive	Mask	Positive	Mask	Positive	Mask
GRACE	81.05		71.27		79.57	
w/o MP	44.34		60.52		72.94	
1-hop	81.71	82.20	71.09	71.57	79.76	79.98
2-hop	80.68	81.95	69.43	71.52	77.41	78.80
3-hop	79.06	81.52	68.47	70.14	74.89	76.86
*1-hop**	82.49	81.83	71.42	71.06	80.54	80.31
*2-hop**	83.86	83.49	71.93	72.02	81.42	80.96
*3-hop**	84.37	85.17	72.64	72.61	83.26	82.78
*all-hop**	86.93	86.60	76.19	75.16	85.60	85.25

arises as to whether it is feasible to narrow the gap between "x-hop" and "x-hop*"? Drawing insights from above observations: (2) and (3) inspire us to capture more "*precise*" hard negatives, while (4) encourages capturing "*expansive*" hard negatives. In unsupervised scenarios, the notion of "*precise*" may appear impractical, and thus, we pivot towards the pursuit of an "*adaptive*" solution.

To further elucidate our perspective, i.e., the close neighbors of an anchor inherently hard to distance, we conducted the following experiment to show the proximity of embeddings at different hops within a two-layer GCN backbone. The results on three datasets are depicted in Figure 2, where the vertical axis represents the average cosine similarity of all pairs of points within a given hop.

(a) Cora	(b) CiteSeer	(c) PubMed
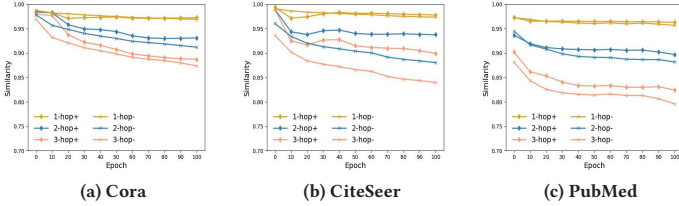
Figure 2: Similarity of embeddings at different hops.

The line named with '-' corresponds to the results trained with a small τ ($\tau = 0.05$), while the line named with '+' corresponds to the results trained with a larger τ ($\tau = 5$). As analyzed in [68], within CL, a small τ exerts strong repulsive forces on neighboring points, whereas a larger τ exerts weaker repulsive forces. This observation is corroborated in 2-hop and 3-hop scenarios. However, this phenomenon is relatively subtle in the case of GCL's 1-hop scenario. As can be observed, the gap between the results of "$1 - hop+$" and "$1 - hop-$" is quite narrow. We attribute this to the fact that 1-hop neighbors often share the identical neighbors, and the frequent message passing makes them hard to be separated.

Besides, we observed that in the context of training with a small τ (strong repulsive forces on neighbors), the similarity rankings across different hops remained undisturbed, and the differences across similarity at different hops even increase. This empirical evidence weakens the motivation that "GCL may push close neighbors further apart" [33, 48]. Moreover, it appears imprudent to address this issue solely from graph structure due to the widespread heterophily on graphs.

The above empirical investigation gives rise to the following concern: on the one hand, capturing more expansive false negatives

approximates the performance under "all labels" setting; on the other hand, it is essential to prevent the capture of true negatives and thus avert the occurrence of 'x-hop' scenario. In other words, this is promising intuitively and entails practical risks.

3.3 Graph Contrastive Learning via Subspace Preserving

Beyond well-known graph homophily [45], we employ subspace preserving techniques to remedy this issue. The essence behind is to mine hard negatives across the entire subspace, rather than limiting it to graph-structured neighbors. Next, we provide the brief definition of subspace preserving.

DEFINITION 1. *(Subspace Preserving) The given data $\{x_i\}_{i=1}^n$ is drawn from a union of an unknown number k of subspaces $\{S_j\}_{j=1}^k$ with unknown dimensions $\{d_i\}_{i=1}^k$. S_j is subspace preserving if $\forall x_i \in S_j$ can be expressed as a linear combination of other points in S_j.*

Based on the so-called self-expressiveness property [10], the coefficients representing the contribution to the anchor can be obtained by solving the optimization problem:

$$\min_{c} \|z - Hc\|_2^2 + \lambda\Omega(c), \qquad (7)$$

where $z \in \mathbb{R}^d$ is the representation of the anchor, and matrix $H \in \mathbb{R}^{d \times m}$ is formed by concatenating the representations of m hard negatives of the anchor. Ω corresponds to a specific constraint on c. Note that the anchor in problem (7) represents any sample and we omit subscript i for simplicity. Upon comparative analysis, we opt for elastic net [98] as Ω in this paper, which is a combination of the ℓ_1 and ℓ_2-norms widely used in machine learning [11, 83, 88, 90]. ℓ_1-penalty encourages sparsity, while ℓ_2-penalty promotes the connectivity. Furthermore, we expect to capture the consistent contribute from each hard negative throughout the entire process. The hard negatives selection for anchor z turns out to be:

$$\min_{c} \sum_{l=1}^{L} \frac{1}{2d_l} \|f_l(z) - \sum_{j \in \Phi} f_l(x_j)c_j\|_2^2 + \lambda\left(\mu\|c\|_1 + \frac{1-\mu}{2}\|c\|_2^2\right) \quad (8)$$

where L is the number of network layers and d_l is the dimension of the l-th layer. There is $z_l \in \mathbb{R}^{d_l}$ and $H_l \in \mathbb{R}^{d_l \times m}$. $\lambda > 0$ is the regularization parameter and $\mu \in [0, 1]$ controls the trade-off between two terms in the elastic net regularizer. As GNN performs message passing between neighbors at each layer, the subspaces at each layer may shift. Therefore, each forward propagation can be regarded as interactions between subspaces: some nodes are drawn into certain subspaces, while some are pushed out of their original subspaces. Scalar $\frac{1}{2d_l}$ is for scale equilibrium. The interpretation of the first term in Eq. (8) is to seek consistent coefficients c across training layers. In other words, if a node consistently resides in the same subspace as the anchor, it is highly likely to be a false negative of the anchor. The magnitude of this possibility depends on the magnitude of self-expression coefficients.

Due to the sparse constraints, problem (8) can not be computed in closed form by SVD. Multiple solutions are provided below.
Full parameterization: If each position of c is considered as a parameter, then problem (8) can be solved in fully parametric way, such as Iterative Shrinkage Thresholding Algorithm (ISTA) [1].
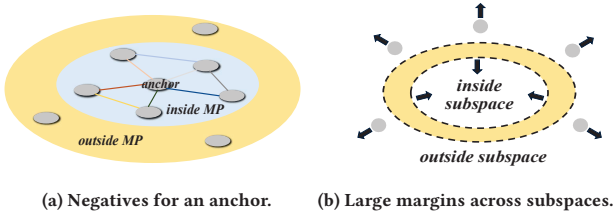
(a) Negatives for an anchor. (b) Large margins across subspaces.

Figure 3: Qualitative schematic of our method.

Moreover, c can also be solved by gradient-based training. Since each sample serves as an anchor, the number of parameters in this strategy is $\sum_{i=1}^{n} m_i$. Updating these parameters during training may bring computational burdens on large-scale data.

MLP parameterization: In this scheme, self-expression coefficients can be computed on the lower-dimensional representations output from MLP. For example, SENet proposed in [88] employs a lightweight query and key network to parameterize the self-expression coefficients. Since MLP parameters does not depend on n, such methods alleviate computational overhead.

Attentive parameterization: Attentive models, such as GAT [64], presuppose varying contributions of distinct features. These models also utilize dimension-related memory to parameterize c.

The number of parameters in the above three ways decreases in order. Correspondingly, the expressive power decreases and the efficiency increases. Since problem (8) is strongly convex, such accelerated proximal gradient mothed or linearized alternating direction method can be applied for seeking unique solution. Selecting non-zero indices in solution c and applying strategies like Eq. (5) and (6) may help attract false negatives within the subspace. Ideally, as shown in Figure 3b, true negatives are pushed farther away while false negatives are masked or explicitly drawn closer. It encourages the emergence of clear class boundaries, i.e. the golden band is stretched in Figure 3b. While the solution process is straightforward, problem (8) itself may not be static; in other words, the pre-selected matrix H may not be optimal. This naturally prompts the question: how is the hard negatives set H selected? Moreover, when dealing with large-scale data, it is extravagant to employ the self-representation of all samples on one single sample, we use a subset instead. Instead, we necessitate the adaptive selection of a subset.

Therefore, we aim to seek an adaptive matrix H which can be self-scaled during the training process. The selected indices are expected to effectively preserve hard negative samples without becoming excessively large thus causing training difficulties. Remark that problem (8) is independent for each sample. Next we introduce the definition of *Adaptive Hard Negative Set* for individual anchor.

DEFINITION 2. *(Adaptive Hard Negative Set) Assume $\tilde{c}(\Phi)$ is the optimal solution of problem (8) with the i-th sample as the anchor. Φ^* is the adaptive hard negatives set of the i-th sample if the following conditions are satisfied:*

$$(a) \ \forall j \notin \Phi^*, \ \tilde{c}^T(\Phi \cup \{j\}) = \left[\tilde{c}^T(\Phi), 0\right],$$
$$(b) \ \forall j \in \Phi^*, \ \tilde{c}^T(\Phi \cup \{j\}) = \left[q^T(\Phi), \alpha_j\right], \quad (9)$$

where $\left[q^T(\Phi), \alpha_j\right]$ denotes the solution vector with scalar $\alpha_j \neq 0$.

The interpretation of this definition is intuitive: j within Φ^* make a contribution to the self-expression of the anchor (i.e., the optimal corresponding coefficient α_j are not zero), while j outside Φ^* will not (i.e., the corresponding optimal coefficient equals to zero).

Inspired by the active-set in subspace clustering [83], Φ^* can be computed via the following theorem.

THEOREM 1. *Assume $\tilde{c}(\Phi)$ is the optimal solution of problem (8) with the i-th sample as the anchor. The auxiliary function is defined as*

$$g(k) = \sum_{l=1}^{L} \frac{1}{d_l} f_l(x_k)^T \left(f_l(z) - \sum_{j \in \Phi} f_l(x_j) c_{ij} \right). \quad (10)$$

Then hard negatives set can be computed by $\Phi^* = \{k \mid |g(k)| > \lambda\mu\}$.

We can now give an understanding of what kind of samples are "hard" for a given anchor in the subspace framework. Theorem 1 implies that a sample is indispensable for subspace preserving if its representation sufficiently resembles the residual of existing self-expression. This diverges from homophily and similarity-based methods. Hence, our method exhibits "adaptive" in two aspects: On the one hand, as evident from the proof, it is clear that $c_j = 0$ is equivalent to $j \notin \Phi^*$. Therefore, it can be removed from the adaptive hard negatives set by updating $\Phi \rightarrow \Phi^*$ once. On the other hand, throughout the training process, updating $\Phi \rightarrow \Phi^*$ continuously expands the hard negatives set for the i-th sample.

We aim for the gradual expansion of Φ with the training process. Beginning with an initial set, Φ can be periodically updated every few epochs to reduce additional time overhead while capturing expansive hard negatives. In addition, to avoid large-scale computations, the size of Φ can be controlled by confining hard negatives within a specified K-hop radius. The hyperparameter K dictates the range of selectable hard negatives.

Combining the solutions of all subproblems, the self-expression matrix can be defined as $C = [c_1, \cdots, c_n]$, where C_{ij} reflects the hardness of i with respect to j. To incorporate the subspace information into the contrastive loss, the self-expression coefficients C_{ij} is supposed to be mapped to the probability that j serves as a false negative for i. This can be done through either a softmax operation or a linear mapping as follows:

$$(a) \ S_{ij} = \frac{\exp\left(|C_{ij}|/\sigma\right)}{\sum_{k \in \Phi_j} \exp\left(|C_{kj}|/\sigma\right)}, \quad (b) \ S_{ij} = \min\left\{\frac{|C_{ij}|}{\zeta}, \rho\right\}. \quad (11)$$

S_{ij} in (a) satisfies probabilistic properties and σ is tunable. S_{ij} in (b) is proportionally scaled from C_{ij}, where ζ is the maximum value within a sampled subset $\left\{C_{ij}\right\}_{(i,j)}$. The truncated parameter ρ controls the ceiling of S_{ij} and is set to 1 by default. In turn, S can be symmetrized by $S \leftarrow (S^T + S)/2$. Then two schemes can be developed to enhance the performance of GCL based on the obtained S.

GRAPE$_{mask}$: GRACE in Eq. (2) treats all samples except itself as negatives, whose negatives set for anchor i can be denoted as $\tilde{N}_i = [n] \setminus \{i\}$. While GRAPE estimates negatives' hardness and obtains the probability S for false negatives in turn, it can subsequently excluded the highly probable false negatives from $\tilde{N}_i$. Specifically, in each epoch, j is included in the false negatives set $\mathcal{F}_i$ for anchor
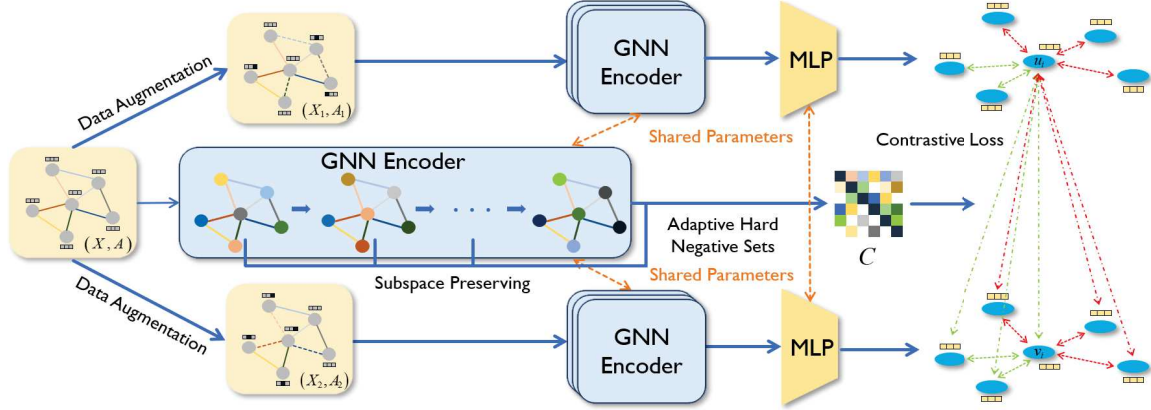
Figure 4: The model architecture of GRAPE. The two views are generated through data augmentation of the initial graph. These three are fed into the parameter-sharing GNN encoder, where the projection header is alternative. The hard negatives set and the corresponding subspace coefficients C are computed within the middle pathway. The green line in contrastive loss indicates hard negatives while the red line indicates true negatives, which are vary across epochs.

i with a probability of S_{ji}. The negatives set in this case turns out to be $\mathcal{N}_i = \tilde{\mathcal{N}}_i \setminus \mathcal{F}_i$. Therefore, the objective for u_i in GRAPE$_{mask}$ is defined as:

$$
\ell_{mask}(u_i) = \\
- \log \frac{e^{\theta(u_i, v_i)/\tau}}{e^{\theta(u_i, v_i)/\tau} + \sum_{j \in \mathcal{N}_i} \left(e^{\theta(u_i, v_j)/\tau} + e^{\theta(u_i, u_j)/\tau} \right)}, \quad (12)
$$

GRAPE$_{pos}$: GRACE in Eq. (2) exclusively treats itself as positives, whose positives set for anchor i is $\tilde{\mathcal{P}}_i = \{i\}$. For anchor i, GRAPE$_{pos}$ incorporates j into the positives set with a probability of S_{ji} each epoch. The expanded positives set is denoted as $\mathcal{P}_i$. Therefore, the objective for u_i in GRAPE$_{pos}$ is defined as:

$$
\ell_{pos}(u_i) = \\
- \log \frac{e^{\theta(u_i, v_i)/\tau} + \sum_{k \in \mathcal{P}_i} \left(e^{\theta(u_i, v_k)/\tau} + e^{\theta(u_i, u_k)/\tau} \right)}{e^{\theta(u_i, v_i)/\tau} + \sum_{j \neq i} \left(e^{\theta(u_i, v_j)/\tau} + e^{\theta(u_i, u_j)/\tau} \right)}, \quad (13)
$$

It is noteworthy that loss (13) is a variant of MIL-NCE [46]. Optimizing loss (13) enhances the overall similarity of positive pairs relative to negative pairs, rather than focusing on instance-specific distances.

Similar to GRACE, the overall contrastive loss is given as:

$$
\mathcal{L}_{mask/pos} = \frac{1}{2n} \sum_{i=1}^{n} \left(\ell_{mask/pos}(u_i) + \ell_{mask/pos}(v_i) \right). \quad (14)
$$

Reviewing the results in Table 1, we can empirically summarize that Grape$_{pos}$ are suitable for high-confidence false negatives, while Grape$_{mask}$ tolerates low-confidence false negatives. Therefore, Grape$_{mask}$ is deemed as a more robust scheme. The model architecture is presented in Figure 4 and the procedure for GRAPE is detailed in Appendix A.

3.4 Theoretical Analysis

Why GRAPE works? Reflecting on our motivation: we aim to identifies expansive and adaptive hard negatives as false negatives, which appears to be empirically derived. The essence behind this

is the message-passing in GNNs: neighbors that encompass a substantial proportion of shared connections are not unduly distanced from each other. Therefore, local hard negative mining yield limited benefits. Recall the results in Table 1 that 1-hop* (even 2-hop*) does not significantly boost the baseline, while 3-hop* shows a leap, which interprets the pursuit of expansive hard negatives. Besides, the self-expression loss can be expanded as follows

$$
\min_c \sum_{l=1}^{L} \frac{1}{2d_l} \| f_l(z) - \sum_{j \in \Phi} f_l(x_j) c_j \|_2^2 \iff \\
\min_c -2 \sum_{j \in \Phi} \left(\sum_{l=1}^{L} \frac{1}{2d_l} f_l^T(z) f_l(x_j) \right) c_j + \sum_{i,j \in \Phi} \left(\sum_{l=1}^{L} \frac{1}{2d_l} f_l^T(x_i) f_l(x_j) \right) c_i c_j
$$
(15)

The first term endows larger self-expression coefficients for negatives similar to the anchor, while the second term endows smaller coefficients for those highly similar to the other negatives. In GCL, the second term implies that the contributions of those involved in message passing with other hard negatives are diminished in self-expression, which is consistent with the intent in Figure 3a. This is rooted in its capacity to capture global long-range interactions, as discussed in [66]. In fact, the self-expression loss can also be interpreted as self-supervised (i.e., reconstructing oneself using others), which fuels contrastive learning in another self-supervised way. Moreover, the regularizer in Eq. (8) exhibits sparsity as μ approaches 1 and group effect as μ approaches 0. The analysis of subspace-preserving property is detailedly discussed in [83]. It is worth noting that λ and μ directly impact the tightness of hard negative selection. GRAPE with large λ and μ results in a small hard negatives set.

By iteratively updating self-expression coefficients during training, the efficacy of GRAPE loss is qualitatively described as follows:

PROPOSITION 1. *In cases where GRAPE captures hard negatives $\{\Phi_i\}_{i=1}^{n}$ within each individual subspace, both $\mathcal{L}_{mask}$ and $\mathcal{L}_{pos}$ contribute to the inter-subspace separation and intra-subspace cohesion.*

Furthermore, GRAPE is associated with various methods, such as graph attention [64], nonlinear latent subspace clustering [50], and uniformity-tolerance dilemma [68].

Table 2: Node classification accuracy in percentage with standard deviation on eight real-world graph datasets. The bold and underlined values indicate the best and the runner-up results respectively.

Methods	Input	Cora	CiteSeer	PubMed	Wiki CS	Am-Photo	Am-Computer	Co-CS	Co-Physics
GCN	X, A, Y	81.32 ± 0.5	70.84 ± 0.7	77.69 ± 0.3	76.85 ± 0.1	92.16 ± 0.2	87.06 ± 0.5	92.54 ± 0.3	95.65 ± 0.2
GAT	X, A, Y	82.57 ± 1.0	71.96 ± 1.0	77.51 ± 0.3	78.35 ± 0.1	91.45 ± 0.4	86.80 ± 0.3	91.98 ± 0.3	95.47 ± 0.2
GAE	X, A	70.49 ± 1.8	63.56 ± 2.1	70.73 ± 1.0	72.08 ± 0.3	88.40 ± 0.3	82.93 ± 0.4	86.83 ± 0.6	92.50 ± 0.3
VGAE	X, A	74.18 ± 1.1	64.85 ± 1.0	71.71 ± 0.5	73.49 ± 0.3	92.20 ± 0.1	86.37 ± 0.2	92.11 ± 0.1	94.52 ± 0.0
DGI	X, A	82.90 ± 0.8	70.14 ± 0.8	76.80 ± 0.6	75.35 ± 0.1	91.61 ± 0.2	83.95 ± 0.5	92.15 ± 0.6	95.38 ± 0.1
GMI	X, A	82.43 ± 0.9	69.85 ± 1.3	79.90 ± 0.2	74.85 ± 0.1	90.68 ± 0.2	82.21 ± 0.3	OOM	OOM
MVGRL	X, A	83.20 ± 0.7	69.85 ± 1.5	78.28 ± 0.2	77.52 ± 0.1	91.74 ± 0.1	87.52 ± 0.1	92.11 ± 0.1	95.13 ± 0.0
GRACE	X, A	81.05 ± 0.3	71.27 ± 0.4	79.57 ± 0.9	78.19 ± 0.0	92.15 ± 0.2	86.25 ± 0.3	92.26 ± 0.0	94.46 ± 0.6
CCA-SSG	X, A	84.20 ± 0.4	72.57 ± 0.3	<u>81.10 ± 0.2</u>	78.42 ± 0.1	92.05 ± 0.3	87.95 ± 0.3	92.03 ± 0.1	<u>95.40 ± 0.1</u>
BGRL	X, A	82.47 ± 0.2	71.13 ± 0.5	80.05 ± 0.2	78.06 ± 0.0	<u>92.95 ± 0.3</u>	<u>88.19 ± 0.3</u>	**93.34 ± 0.1**	**95.54 ± 0.1**
ProGCL$_W$	X, A	81.79 ± 0.6	68.63 ± 0.6	78.16 ± 0.2	78.30 ± 0.2	92.47 ± 0.2	87.23 ± 0.2	92.57 ± 0.1	OOM
COSTA$_{MV}$	X, A	81.66 ± 0.2	72.42 ± 0.4	78.39 ± 0.6	<u>78.67 ± 0.1</u>	92.20 ± 0.3	88.09 ± 0.0	<u>92.96 ± 0.1</u>	95.24 ± 0.0
GRAPE$_{mask}$	X, A	**85.18 ± 0.0**	<u>72.59 ± 0.0</u>	81.50 ± 0.2	**79.11 ± 0.1**	**93.32 ± 0.0**	**88.42 ± 0.1**	92.78 ± 0.0	95.37 ± 0.0
GRAPE$_{pos}$	X, A	<u>85.07 ± 0.0</u>	**73.54 ± 0.1**	79.84 ± 0.2	78.13 ± 0.1	<u>92.95 ± 0.0</u>	87.46 ± 0.1	92.29 ± 0.1	95.08 ± 0.0

Maximizing mutual information The improvement of GRAPE over the baseline can also be elucidated from the perspective of maximizing Mutual Information (MI):

THEOREM 2. *The contrastive loss in Eq. (14) gives a stricter lower bound of MI between input features X and embeddings in two views U and V, compared with the contrastive loss $\mathcal{L}$ in Eq. (3) proposed by GRACE. This can be written formally as*

$$- \mathcal{L} < -\mathcal{L}_{mask/pos} \leqslant \mathcal{I}(U; V) \qquad (16)$$

Therefore, maximizing GRAPE loss corresponds to optimizing a more rigorous lower bound for the mutual information between node features and the acquired node representations, thereby furnishing a theoretical justification for the performance enhancement. **Complexity Analysis** Compared to our baseline, GRACE, extra complexity arises from the periodic updating of hard negatives set $\{\Phi_i\}_{i=1}^n$ and the computation of self-expression coefficients $\{c_i\}_{i=1}^n$ every $Intvl$ epochs. Each of these n independent subproblems can be solved concurrently in parallel. The additional time overhead is $O(Md)$, where M represents the largest cardinality within $\{\Phi_i\}_{i=1}^n$. Since the hard negatives sets are restricted within the K-hop, there is $M \ll n$. Therefore, the additional time overhead is manageable.

4 EXPERIMENTS

4.1 Experimental Protocol

We conducted comparisons between GRAPE and ten advanced methods on eight node prediction datasets. The benchmark graph datasets include: **Cora**, **CiteSeer**, **PubMed**, **Wiki CS**, **Amazon Photo**, **Amazon Computers**, **Coauthor CS**, **Coauthor Physics**. They are all hosted by DGL package [2], where dataset information is detailed. The comparative methods include: two supervised baselines (**GCN** [27], **GAT** [64]), two autoencoder-based baselines (**GAE** [28], **VGAE** [28]), eight state-of-the-art GCL methods (**DGI** [65], **GMI** [51], **MVGRL** [18], **GRACE** [96], **CCA-SSG** [87], **BGRL** [61], **ProGCL$_W$** [76], **COSTA$_{MV}$** [89]).

For all augmentation-based methods, we adopt the most commonly used strategies for the graph augmentation: "edge removing" and "feature masking" [95]. At each epoch, "edge removal" randomly removes a certain proportion of edges from the original

graph, while "feature masking" randomly masks a certain proportion of features. To be consistent with the comparison method, we configure the GNN encoder as a two-layer GCN. Self-supervised training is conducted on the entire graph and on the features of all samples. The embeddings obtained are fed into a ℓ_2-regularized logistic regression to get the final result. For Cora, CiteSeer, and PubMed datasets, we employ the standard split settings: 20 nodes per class are available for training, 500 nodes for validation and 1000 for testing. For the other datasets, we randomly assign 10% of the nodes for training, another 10% for validation, and allocate the remaining 80% for testing. The overall model is trained using the Adam optimizer.

We implement our GRAPE based on GRACE. The max training epoch is set to 100. The dimensions in the two-layer GNN encoder are set to 512 and 256, respectively. The learning rate for GRAPE is set to 1×10^{-3}, while that for linear classifiers is set to 1×10^{-2}. The interval for updating C $Intvl$ is fixed to 5 and the truncated parameter ρ is fixed to 1. Our graph augmentation is achieved through a combination of 40% edge removal and 10% feature masking. The trade-off parameter λ is selected within $\{10^{-1}, 10^0, 10^1, 10^2\}$ and μ is selected within $\{0, 0.1, \cdots, 0.9, 1.0\}$. The temperature parameter τ is selected within $\{0.1, 0.2, \cdots, 1.0\}$ and the range of hard negatives K is selected within $\{1, \cdots, 5\}$. For all comparative methods, we adhere to the authors' default parameter settings and, where necessary, conduct parameter grid searches to achieve fair comparisons. Their implementations are all open-sourced. All experiments are conducted on NVIDIA RTX A6000 GPU with 48GB memory.

4.2 Main Results

The node classification results are presented in Table 2. The reported results are averaged over 10 runs with random seeds. Accuracy is rounded to two decimal places, while standard deviation is rounded to one decimal place. The "Input" refers to data for training, where X, A and Y denotes feature matrix, adjacency matrix and label matrix respectively. OOM denotes out of memory. It can be observed that GRAPE achieves the state-of-the-art self-supervised performance on the first six datasets and surpasses the performance of supervised baselines (GCN, GAT) on the first seven datasets. Compared to its baseline GRACE, GRAPE achieves a comprehensive improvement. The hyperparameters involved in the experiment are

---

[2]https://github.com/dmlc/dgl

listed in Appendix C.1. We perform node clustering performance evaluations in the completely unsupervised case in Appendix C.2. These results corroborate GRAPE's capacity for precise identification of false negatives.

4.3 How GRAPE Affects Training?

Wang et al. [68] introduced the concept of uniformity-tolerance dilemma in contrastive representation. We employ the two metrics to showcase the difference between GRAPE and its baseline GRACE. Specifically, the cohesion (CO) and uniformity (UN) of the learned embeddings can be defined as follows:

$$\text{CO} = \sum_{y_i = y_j} \left(f^T(\boldsymbol{x_i}) f(\boldsymbol{x_j}) \right), \ \text{UN} = \sum_{i,j} \exp \left(-f^T(\boldsymbol{x_i}) f(\boldsymbol{x_j}) \right) \ (17)$$

f denotes our GNN encoder $f_\Theta(\boldsymbol{A}, \boldsymbol{X})$. A higher CO implies higher intra-class cohesion, while a higher UN implies a more uniform embedding distribution. The comparison of the two metrics for GRAPE and GRACE during training is depicted in Figure 5.
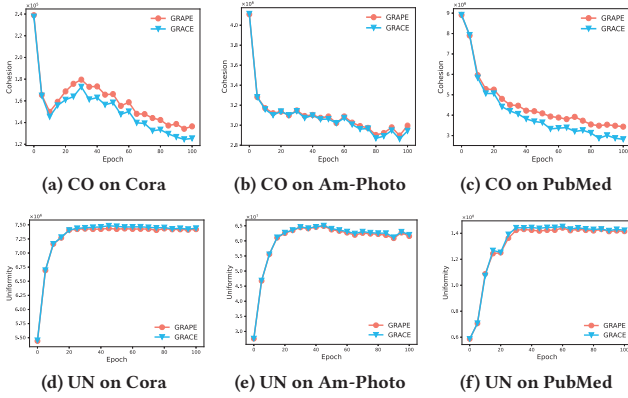

(a) CO on Cora   **(b) CO on Am-Photo**   **(c) CO on PubMed**

(d) UN on Cora   **(e) UN on Am-Photo**   **(f) UN on PubMed**

Figure 5: Variation of cohesion and uniformity.

At the beginning of training, CO and UN for both GRAPE and GRACE are nearly identical due to the similar initialization. As discussed ahead, GRAPE explicitly or implicitly brings the representations inside the same subspace closer, which strengthens the intra-class cohesion and reduces the global uniformity. With the expansion of the hard negatives set, the margin of cohesion between GRAPE and GRACE is enlarged, which is in line with our original intention. Since the mask mechanism of GRAPE$_{mask}$ is presented in a probabilistic form, uniformity doesn't exhibit significant decreases compared to GRACE. Additionally, Figure 6 shows how the test accuracy steadily improves as the GRAPE loss is optimized.
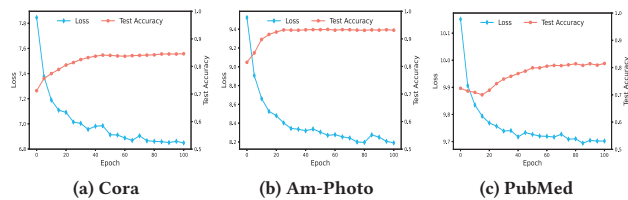

(a) Cora   **(b) Am-Photo**   **(c) PubMed**

Figure 6: Variation of loss and test accuracy with training.

Furthermore, the efficiency of GRAPE can be enhanced through aforementioned the scalable parameterization. The corresponding results are provided in Appendix C.3.

We present intuitive results to illustrate the effectiveness of GRAPE. Figure 7 shows the distributions of the true/false negatives of the same anchor in different phases on Cora. The horizontal axis denotes the cosine similarity between negatives and anchor, which is non-negative due to the ReLU before output. The variation with training is discernible, especially from (b) to (c), validating the efficacy of adaptive hard negative selection.
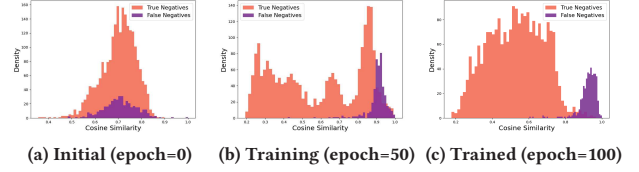

(a) Initial (epoch=0)   **(b) Training (epoch=50)**   **(c) Trained (epoch=100)**

Figure 7: Negatives distributions in different phases.

We present t-SNE visualization of GRAPE's running results without labels (i.e., before classification). As depicted in Figure 8, nodes are partitioned into multiple distinct clusters.
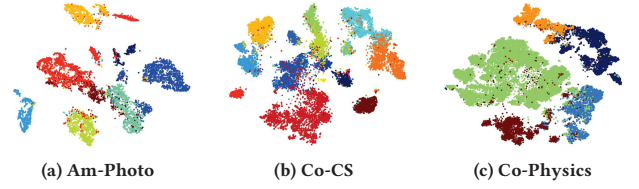

(a) Am-Photo   **(b) Co-CS**   **(c) Co-Physics**

Figure 8: Visualization of node embedding without labels.

The influence of the hyperparameters in GRAPE is examined to validate the feasibility in Appendix C.4. Furthermore, in contrast to methods such as BGRL, we observe that GRAPE displays a relatively low sensitivity to graph augmentation. All results are under the setting with 40% edge removal and 10% feature masking, which provides a substantial advantage compared to other GCL methods.

5 CONCLUSION

In this paper we put forth a novel method for estimating negatives' hardness in GCL. Our method emphasizes the potential in exploring expansive and adaptive negatives. These two goals are coupled in our subspace preserving scheme. We shed light on the motivation, provide empirical and theoretical underpinnings and conduct comprehensive experiments to dissect the effectiveness of GRAPE. The proposed method has two limitations: it is only measured on homogeneous graphs and is only applicable in transductive learning settings. Drawing from the contributions of this paper, we hopefully point out two interesting and promising avenues for further research. First, since subspace theory is not directly reliant on existing connections, it shows potential in addressing the impact of noisy, incomplete, or vulnerable graph structures on GNNs (a branch called graph structure learning). Second, self-expression contribute to preserving local structures and may serve as a form of constraint to slow down message passing for deeper GNNs.

REFERENCES

[1] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* 2, 1 (2009), 183–202.

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *ICML*. 531–540.

[3] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 7 (1997), 711–720.

[4] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. 2015. Diversity-induced multi-view subspace clustering. In *CVPR*. 586–594.

[5] Shuo Chen, Gang Niu, Chen Gong, Jun Li, Jian Yang, and Masashi Sugiyama. 2021. Large-margin contrastive learning with distance polarization regularizer. In *ICML*. 1673–1683.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.

[7] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *CVPR*. 15750–15758.

[8] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. In *NeurIPS*, Vol. 33. 8765–8775.

[9] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2021. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *CVPR*. 9588–9597.

[10] Ehsan Elhamifar and René Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 11 (2013), 2765–2781.

[11] Yuqiang Fang, Ruili Wang, Bin Dai, and Xindong Wu. 2014. Graph-based learning via auto-grouped sparse regularization and kernelized extension. *IEEE Trans. Knowl. Data Eng.* 27, 1 (2014), 142–154.

[12] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*. 6894–6910.

[13] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. In *NeurIPS*, Vol. 32.

[14] 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. 2020. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, Vol. 33. 21271–21284.

[15] Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *J. Mach. Learn. Res.* 13, 2 (2012).

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, Vol. 2. 1735–1742.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*, Vol. 30.

[18] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*. 4116–4126.

[19] Xiaofei He and Partha Niyogi. 2003. Locality preserving projections. In *NeurIPS*, Vol. 16.

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

[21] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. 2014. Smooth representation clustering. In *CVPR*. 3834–3841.

[22] Anil Jain and Douglas Zongker. 1997. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 2 (1997), 153–158.

[23] Yizhu Jiao, Yun Xiong, Jiawei Zhang, Yao Zhang, Tianqi Zhang, and Yangyong Zhu. 2020. Sub-graph contrast for scalable self-supervised graph representation learning. In *ICDM*. 222–231.

[24] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. In *NeurIPS*, Vol. 33. 21798–21809.

[25] Mohsen Kheirandishfard, Fariba Zohrizadeh, and Farhad Kamangar. 2020. Deep low-rank subspace clustering. In *CVPR*. 864–865.

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

[27] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

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

[29] Simin Kou, Xuesong Yin, Yigang Wang, Songcan Chen, Tieming Chen, and Zizhao Wu. 2023. Structure-Aware Subspace Clustering. *IEEE Trans. Knowl. Data Eng.* (2023).

[30] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. 2006. Efficient sparse coding algorithms. In *NeurIPS*, Vol. 19.

[31] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. 2020. *i*-Mix: A Domain-Agnostic Strategy for Contrastive Representation Learning. In *ICLR*.

[32] Namkyeong Lee, Junseok Lee, and Chanyoung Park. 2022. Augmentation-free self-supervised learning on graphs. In *AAAI*, Vol. 36. 7372–7380.

[33] Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. 2023. HomoGCL: Rethinking Homophily in Graph Contrastive Learning. *arXiv preprint arXiv:2306.09614* (2023).

[34] Zechao Li, Jing Liu, Jinhui Tang, and Hanqing Lu. 2015. Robust structured subspace learning for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 10 (2015), 2085–2098.

[35] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. 2012. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2012), 171–184.

[36] Guangcan Liu and Shuicheng Yan. 2011. Latent low-rank representation for subspace segmentation and feature extraction. In *ICCV*. 1615–1622.

[37] Nian Liu, Xiao Wang, Deyu Bo, Chuan Shi, and Jian Pei. 2022. Revisiting graph contrastive learning from the perspective of graph spectrum. In *NeurIPS*, Vol. 35. 2972–2983.

[38] Weijie Liu, Jiahao Xie, Chao Zhang, Makoto Yamada, Nenggan Zheng, and Hui Qian. 2022. Robust Graph Dictionary Learning. In *ICLR*.

[39] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. 2022. Graph self-supervised learning: A survey. *IEEE Trans. Knowl. Data Eng.* 35, 6 (2022), 5879–5900.

[40] Yue Liu, Xihong Yang, Sihang Zhou, and Xinwang Liu. 2023. Simple contrastive graph clustering. *IEEE TNNLS* (2023).

[41] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2023. Hard sample aware network for contrastive deep graph clustering. In *AAAI*, Vol. 37. 8914–8922.

[42] Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, and Shuicheng Yan. 2018. Subspace clustering by block diagonal representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 2 (2018), 487–501.

[43] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. 2012. Robust and efficient subspace segmentation via least squares regression. In *ECCV*. 347–360.

[44] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis Bach. 2008. Supervised dictionary learning. In *NeurIPS*, Vol. 21.

[45] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[46] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*. 9879–9889.

[47] Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, Vol. 14.

[48] Zhiyuan Ning, Pengfei Wang, Pengyang Wang, Ziyue Qiao, Wei Fan, Denghui Zhang, Yi Du, and Yuanchun Zhou. [n.d.]. Graph soft-contrastive learning via neighborhood ranking. ([n. d.]).

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

[50] Vishal M Patel, Hien Van Nguyen, and René Vidal. 2013. Latent space sparse subspace clustering. In *ICCV*. 225–232.

[51] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*. 259–270.

[52] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. 2013. Personalized recommendation combining user interest and social circle. *IEEE Trans. Knowl. Data Eng.* 26, 7 (2013), 1763–1777.

[53] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*. 1150–1160.

[54] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive Learning with Hard Negative Samples. In *ICLR*.

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

[56] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, Vol. 29.

[57] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000* (2019).

[58] Qingqiang Sun, Wenjie Zhang, and Xuemin Lin. 2023. Progressive Hard Negative Masking: From Global Uniformity to Local Tolerance. *IEEE Trans. Knowl. Data Eng.* (2023).

[59] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. 2021. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*, Vol. 34. 15920–15933.

[60] Joshua B Tenenbaum, Vin de Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500 (2000), 2319–2323.

[61] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. 2021. Large-Scale Representation Learning on Graphs via Bootstrapping. In *ICLR*.

[62] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning?. In *NeurIPS*, Vol. 33. 6827–6839.

[63] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. 2019. On Mutual Information Maximization for Representation Learning. In *ICLR*.

[64] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[65] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. In *ICLR*.

[66] Rene Vidal. 2021. Attention: Self-Expression Is All You Need. (2021).

[67] Cédric Vincent-Cuaz, Titouan Vayer, Rémi Flamary, Marco Corneli, and Nicolas Courty. 2021. Online graph dictionary learning. In *ICML*. 10564–10574.

[68] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *CVPR*. 2495–2504.

[69] Yu-Xiang Wang and Huan Xu. 2013. Noisy sparse subspace clustering. In *ICML*. 89–97.

[70] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. 2003. Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.* 3 (2003), 1439–1461.

[71] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. 2008. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 2 (2008), 210–227.

[72] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: neural news recommendation with personalized attention. In *KDD*. 2576–2584.

[73] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *SIGIR*. 235–244.

[74] Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. 2020. Conditional Negative Sampling for Contrastive Learning of Visual Representations. In *ICLR*.

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

[76] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z Li. 2022. ProGCL: Rethinking Hard Negative Mining in Graph Contrastive Learning. In *ICML*. 24332–24346.

[77] Jun Xia, Yanqiao Zhu, Yuanqi Du, and Stan Z Li. 2022. A survey of pretraining on graphs: Taxonomy, methods, and applications. *arXiv preprint arXiv:2202.07893* (2022).

[78] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*. 478–487.

[79] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 2 (2022), 2412–2429.

[80] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. 2019. Gromov-wasserstein learning for graph matching and node embedding. In *ICML*. 6932–6941.

[81] Jun Xu, Mengyang Yu, Ling Shao, Wangmeng Zuo, Deyu Meng, Lei Zhang, and David Zhang. 2019. Scaled simplex representation for subspace clustering. *IEEE Trans. Cybern.* 51, 3 (2019), 1493–1505.

[82] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. 2006. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1 (2006), 40–51.

[83] Chong You, Chun-Guang Li, Daniel P Robinson, and René Vidal. 2016. Oracle based active set algorithm for scalable elastic net subspace clustering. In *CVPR*. 3928–3937.

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

[85] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *NeurIPS*, Vol. 33. 5812–5823.

[86] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*. 1294–1303.

[87] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. 2021. From canonical correlation analysis to self-supervised graph neural networks. In *NeurIPS*, Vol. 34. 76–89.

[88] Shangzhi Zhang, Chong You, René Vidal, and Chun-Guang Li. 2021. Learning a self-expressive network for subspace clustering. In *CVPR*. 12393–12403.

[89] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. 2022. COSTA: covariance-preserving feature augmentation for graph contrastive learning. In *KDD*. 2524–2534.

[90] Zheng Zhang, Zhihui Lai, Yong Xu, Ling Shao, Jian Wu, and Guo-Sen Xie. 2017. Discriminative elastic-net regularized linear regression. *IEEE Trans Image Process*

[91] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph Debiased Contrastive Learning with Joint Representation Clustering.. In *IJCAI*. 3434–3440.

[92] Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. 2022. Graph data augmentation for graph machine learning: A survey. *arXiv preprint arXiv:2202.08871* (2022).

[93] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2021. Data augmentation for graph neural networks. In *AAAI*, Vol. 35. 11015–11023.

[94] Jiangbin Zheng, Yile Wang, Ge Wang, Jun Xia, Yufei Huang, Guojiang Zhao, Yue Zhang, and Stan Li. 2022. Using Context-to-Vector with Graph Retrofitting to Improve Word Embeddings. In *ACL*. 8154–8163.

[95] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116* (2021).

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

[97] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*. 2069–2080.

[98] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67, 2 (2005), 301–320.

26, 3 (2017), 1466–1481.

A PROCEDURE FOR GRAPE

Algorithm 1 Procedure for GRAPE.

Input: Initial graph $G = (X, A)$, temperature parameter: τ, trade-off parameters: λ, μ, range of hard negatives: K, interval for updating C: $Intvl$, maximum epochs: Epo.

1: **Initialization:** Randomly initialize the GNN parameters. Determine candidate set of hard negatives $\{\Phi_i\}_{i=1}^n$ and set the self-expression coefficients $\{c_i\}_{i=1}^n$ to zero.

2: **for** $epoch = 1$ to Epo **do**

3:     Generate two augmented graphs $G_1 = (X_1, A_1)$ and $G_2 = (X_2, A_2)$. Feed G, G_1, G_2 into GNN encoder to obtain embeddings $\{z_i\}_{i=1}^n$, $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^n$.

4:     **if** $epoch \% intvl == 0$ **then**

5:         Compute the self-expression coefficients $\{c_i\}_{i=1}^n$ on former hard negatives by solving Eq. (8).

6:         Update $\{\Phi_i\}_{i=1}^n$ within K-hop with solution $\{c_i\}_{i=1}^n$.

7:         Re-compute the self-expression coefficients $\{c_i\}_{i=1}^n$ on Φ and obtain S by Eq. (11)

8:     **end if**

9:     Compute contrastive loss $\mathcal{L}_{mask/pos}$ in Eq. (12) or (13).

10:     Update $f_\Theta(A, X)$ with Adam by minimizing the overall loss in Eq. (14);

11: **end for**

Output: The trained $f_\Theta(A, X)$ and the node embeddings $\{z_i\}_{i=1}^n$.

B DETAILED PROOF

THEOREM 1. *Assume $\tilde{c}(H)$ is the optimal solution of problem (8). The auxiliary function is defined as*

$$g(\boldsymbol{h}) = \sum_{l=1}^{L} \frac{1}{d_l} \boldsymbol{h}_l^T \left(z_l - H_l \tilde{c}(H) \right). \tag{1}$$

Then hard negatives set can be computed by $\Phi = \{\boldsymbol{h} \mid |g(\boldsymbol{h})| > \lambda\mu\}$.

PROOF. Problem (8) can be reformulated as

$$\min_{\boldsymbol{c}} \frac{1}{2} \|z - H\boldsymbol{c}\|_2^2 + \lambda \left(\mu \|\boldsymbol{c}\|_1 + \frac{1-\mu}{2} \|\boldsymbol{c}\|_2^2 \right) \tag{2}$$

where

$$z = [\frac{1}{d^{(1)}} z^{(1)T}, \cdots, \frac{1}{d^{(L)}} z^{(L)T}]^T$$
$$H = [\frac{1}{d^{(1)}} H^{(1)T}, \cdots, \frac{1}{d^{(L)}} H^{(L)T}]^T \quad (3)$$

By taking derivatives, the optimal solution $\tilde{c}(H)$ to problem (2) satisfies:

$$\lambda(1-\mu)\tilde{c}(H) = \Gamma_{\lambda\mu}\left(H^T\left(z - H\tilde{c}(H)\right)\right). \quad (4)$$

Let $\left[q^T(H), g(h)\right]^T$ be the optimal solution for problem

$$\min_c \|z - [H, h] c\|_2^2 + \lambda\left(\mu\|c\|_1 + \frac{1-\mu}{2}\|c\|_2^2\right). \quad (5)$$

Then there exist

$$\lambda(1-\mu)\left[q^T(H), g(h)\right]^T = \Gamma_{\lambda\mu}\left([H, h]^T\left(z - [H, h]\left[q^T(H), g(h)\right]^T\right)\right). \quad (6)$$

By splitting the counterpart terms, the following two equations hold:

$$\lambda(1-\mu)q(H) = \Gamma_{\lambda\mu}\left(H^T\left(z - Hq(H) - hg(h)\right)\right) \quad (7)$$

$$\lambda(1-\mu)g(h) = \Gamma_{\lambda\mu}\left(h^T\left(z - Hq(H) - hg(h)\right)\right) \quad (8)$$

If $h \notin \Phi$, then $\left[c^T(H), 0\right]^T$ is an optimal solution because it meets Eq. (7) and (8). Since the optimal solution to problem (5) is unique, condition (a) is thus satisfied. Since the optimal solution to problem (5) is unique, term (a) stipulated in the definition of Φ holds.

In the case where $h \in \Phi$, we show that $g(h)$ is not equal to 0. If $g(h) = 0$, due to the uniqueness of the optimal solution in problem (2), Eq. (7) deduces $q(H) = \tilde{c}(H)$. However, the obtained solution $\left[c^T(H), 0\right]^T$ does not satisfy Eq. (8). Therefore $g(h) \neq 0$ holds. Combining the above discussion, Φ is the adaptive hard negatives set by Definition 2, which completes the proof. □

PROPOSITION 2. *If GRAPE captures hard negatives $\{\Phi_i\}_{i=1}^n$ within each individual subspace, both $\mathcal{L}_{mask}$ and $\mathcal{L}_{pos}$ contribute to the inter-subspace separation and intra-subspace cohesion.*

PROOF. Compared to GRACE, GRAPE$_{pos}$ explicitly brings hard negative samples within the same subspace closer while repelling negatives outside the subspace. Our focus then turns to GRAPE$_{mask}$. From gradient analysis, the ratio of the gradients of negatives to that of positives can be defined following [68]:

$$r(u_i, v_j) = \left|\frac{\partial \ell(u_i)}{\partial \theta(u_i, v_j)}\right| / \left|\frac{\partial \ell(u_i)}{\partial \theta(u_i, v_i)}\right|, \quad (9)$$

representing the relative penalty on negatives. The ratio in GRACE and GRAPE can be derived as follows:

$$\text{GRACE:} \quad r_1(u_i, v_j) = \frac{e^{\theta(u_i, v_j)/\tau}}{\sum_{k \neq i}\left(e^{\theta(u_i, v_j)/\tau} + e^{\theta(u_i, u_j)/\tau}\right)}, \quad (10)$$

$$\text{GRAPE:} \quad r_2(u_i, v_j) = \frac{e^{\theta(u_i, v_j)/\tau}}{\sum_{k \in \mathcal{N}_i}\left(e^{\theta(u_i, v_j)/\tau} + e^{\theta(u_i, u_j)/\tau}\right)}. \quad (11)$$

Clearly there is $r_2(u_i, v_j) \geqslant r_1(u_i, v_j)$, which implies that GRAPE imposes a greater penalty on negative pairs that are not within the same subspace. Moreover, posit that $S_{ij} = S_{ji} = 1$, i.e., v_i is not in the denominator of $\ell(u_j)$ and vice versa. In this case, $\theta(u_i, v_j)$ is not penalized explicitly. Apart from self-alignment, the subproblem involving u_i in the process of minimizing $\mathcal{L}_{mask}$ is equivalent to:

$$\min_{u_i} \sum_{i \notin \mathcal{N}_k \vee k \notin \mathcal{N}_i}\left(e^{\theta(u_i, u_k)} + e^{\theta(u_i, v_k)}\right) \quad (12)$$

If we consider the first-order Taylor expansion of the problem and omit the second or higher-order infinitesimal terms, problem (12) simplifies to

$$\min_{u_i} \sum_{i \notin \mathcal{N}_k \vee k \notin \mathcal{N}_i}\left(\theta(u_i, u_k) + \theta(u_i, v_k)\right). \quad (13)$$

It is clear that there is a unique solution to the above problem. If i and j belong to the same subspace, the overlap between set $\{k | i \notin \mathcal{N}_k \vee k \notin \mathcal{N}_i\}$ and set $\{k | j \notin \mathcal{N}_k \vee k \notin \mathcal{N}_j\}$ appears to be high. Hence, the optimal solutions of the subproblems for u_i and v_j tends to exhibit high similarity. As a result, u_i and v_j are implicitly drawn closer by updating the network parameters. Thereby we prove the Proposition 2 qualitatively. □

THEOREM 2. *The contrastive loss in Eq. (14) gives a stricter lower bound of MI between input features X and embeddings in two views U and V, compared with the contrastive loss $\mathcal{L}$ in Eq. (3) proposed by GRACE. This can be written formally as*

$$-\mathcal{L} < -\mathcal{L}_{mask/pos} \leqslant \mathcal{I}(X; U, V) \quad (14)$$

PROOF. The proof can be analogized to [58]. □

Table 4: Node clustering results in percentage on three graph datasets.

Datasets	PubMed		Am-Photo		Am-Computer	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI
GAE	24.41	24.35	57.30	49.45	42.80	24.68
VGAE	21.44	18.54	54.18	40.25	42.88	23.74
DGI	27.96	29.50	44.77	35.11	37.35	20.25
GMI	24.96	25.04	50.47	42.22	45.88	30.50
MVGRL	31.96	30.79	56.48	44.06	29.18	19.57
GRACE	26.01	28.44	61.93	50.41	48.76	33.85
CCA-SSG	25.04	28.22	62.30	53.87	49.62	36.64
ProGCL$_W$	27.26	29.50	60.54	48.32	43.29	28.44
COSTA$_{MV}$	27.91	28.59	58.69	48.87	45.54	36.90
GRAPE$_{mask}$	32.13	31.80	65.33	57.72	53.06	38.49
GRAPE$_{pos}$	**33.98**	**32.91**	**66.32**	**59.65**	**55.74**	**41.82**

C EXPERIMENTAL SETUP

C.1 Hyperparameter Setting

Table 3 lists the hyperparameters for our main performance experiments. The two-layer GCN maintains its dimensions at 512 and 256 consistently. Notably, the parameters do not require meticulous tuning, implying that GRAPE obviates the need for parameter search. This indicates the sound scalability of our method.
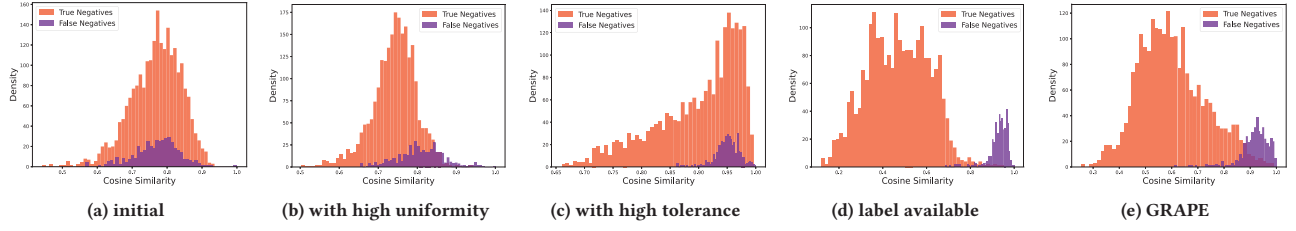
| (a) initial | (b) with high uniformity | (c) with high tolerance | (d) label available | (e) GRAPE |

Figure 9: Similarity histograms of negatives

Table 3: Statistics of datasets and corresponding hyperparameter settings.

Datasets	Domain	Nodes	Edges	Features	Classes	λ	μ	τ	K	ρ	$Intvl$	Epo
Cora	Citation Network	2708	10556	1433	7	1	0.5	0.5	5	1	5	100
CiteSeer	Citation Network	3327	9228	3703	6	1	0.5	1.0	5	1	5	100
PubMed	Citation Network	19717	88651	500	3	1	0.5	1.0	4	1	5	100
Wiki CS	Knowledge Base	11701	216213	300	10	1	0.1	0.2	2	1	5	100
Am-Photo	Social Network	7650	119081	745	8	1	0.1	0.5	2	1	5	100
Am-Computer	Social Network	13752	245861	767	10	1	0.5	0.5	3	1	5	100
Co-CS	Citation Network	18333	81894	6805	15	1	0.5	0.5	3	1	5	100
Co-Physics	Citation Network	34493	247962	8415	5	100	0.5	0.5	2	1	5	100

C.2 Node Clustering

The evaluation of node clustering is similar to classification, except that k-means is employed for clustering. Clustering performance is assessed using NMI and ARI, where higher values of these metrics indicate superior clustering results. The average results of the 5 runs are presented in Table 4.

C.3 On Scalable Parameterization

Taking GRAPE$_{mask}$ as the example, the test accuracy and training time are exhibited in Table 5.

Table 5: Test accuracy (%) and training time (s) on different parameterization.

Methods	PubMed	Am-Photo	Am-Computer
Full	81.50% (116.59)	93.32% (22.57)	88.42% (66.32)
MLP	79.26% (93.08)	92.10% (18.61)	87.59% (51.19)
Attentive	79.43% (87.49)	91.92% (18.03)	87.50% (50.80)

C.4 Supplementary Parameter Analysis

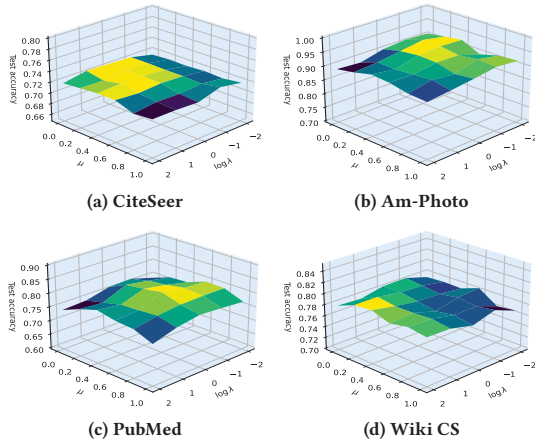| (a) CiteSeer | (b) Am-Photo |
| (c) PubMed | (d) Wiki CS |

Figure 10: Sensitivity of trade-off parameters.

The sensitivity analysis of the two trade-off parameters in Eq. (8) is depicted in Figure 10. The test accuracy of GRAPE remains stable across a wide range of μ and λ, indicating its independence from meticulous parameter settings. Simultaneously, both parameters indeed exert an influence on the model. Besides, GRAPE framework is applicable to any family of GNN and the performance gap is not remarkably pronounced.
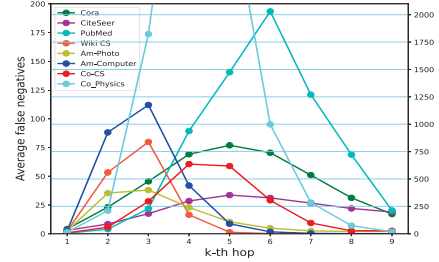
C.5 Illustrative Experiment

Figure 11: The number of average false negatives at each hops on eight graph datasets.

We plot the average distributions of false negatives on eight most commonly used network datasets in Figure 11. Cora and Citeseer correspond to the left coordinate axis, while the rest correspond to the right coordinate axis. It can be observed that false negatives are prevalent over an expansive range. This gives rise to the following concern: on the one hand, capturing more expansive false negatives approximates the performance under "*all-hop**" setting; on the other hand, it is essential to prevent the capture of true negatives and thus avert the occurrence of '*x-hop*' scenario.

An intuitive comparison, as depicted in Figure 9, showcases the average distribution of negatives on Cora. Obviously, solely adjusting the temperature parameter τ does not suffice to achieve local tolerance. The setting with high uniformity (small τ) pushes false negatives further away, whereas the setting with high tolerance (large τ) makes it challenging to discriminate true negatives. Compared to temperature-based methods, GRAPE alleviates class collisions effectively.