Revisiting Self-Supervised Heterogeneous Graph Learning from Spectral Clustering Perspective

Yujie Mo[1,2] **Zhihe Lu**[2] **Runpeng Yu**[2] **Xiaofeng Zhu**[1*] **Xinchao Wang**[2*]
[1]School of Computer Science and Engineering,
University of Electronic Science and Technology of China
[2]National University of Singapore

Abstract

Self-supervised heterogeneous graph learning (SHGL) has shown promising potential in diverse scenarios. However, while existing SHGL methods share a similar essential with clustering approaches, they encounter two significant limitations: (i) noise in graph structures is often introduced during the message-passing process to weaken node representations, and (ii) cluster-level information may be inadequately captured and leveraged, diminishing the performance in downstream tasks. In this paper, we address these limitations by theoretically revisiting SHGL from the spectral clustering perspective and introducing a novel framework enhanced by rank and dual consistency constraints. Specifically, our framework incorporates a rank-constrained spectral clustering method that refines the affinity matrix to exclude noise effectively. Additionally, we integrate node-level and cluster-level consistency constraints that concurrently capture invariant and clustering information to facilitate learning in downstream tasks. We theoretically demonstrate that the learned representations are divided into distinct partitions based on the number of classes and exhibit enhanced generalization ability across tasks. Experimental results affirm the superiority of our method, showcasing remarkable improvements in several downstream tasks compared to existing methods.

1 Introduction

Self-supervised heterogeneous graph learning (SHGL) aims to effectively process diverse types of nodes and edges in the heterogeneous graph, producing low-dimensional representations without the need for human annotations [72, 71, 25]. Thanks to its remarkable capabilities, SHGL has attracted significant interest and has been utilized in a broad array of applications, including recommendation systems [44, 12], social network analysis [45, 9], and molecule design [68, 59].

Existing SHGL methods can be broadly classified into two groups, *i.e.,* meta-path-based methods and adaptive-graph-based methods. Meta-path-based methods typically utilize pre-defined meta-paths to explore relationships among nodes that may share the same label in the heterogeneous graph [18, 74]. However, building meta-paths requires extensive prior knowledge and incurs additional computation costs [69]. To address these drawbacks, adaptive-graph-based methods dynamically assign significant weights to node pairs likely to share the same label, using the adaptive graph structure rather than traditional meta-paths [30]. Both groups of SHGL methods facilitate message-passing among nodes within the same class, either through meta-path-based graphs or adaptive graph structures. As a result, this process minimizes intra-class differences and promotes a clustered pattern in the learned representations, aligning these methods closely with conventional clustering techniques.

*Corresponding authors

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

 https://doi.org/10.52202/079017-1366

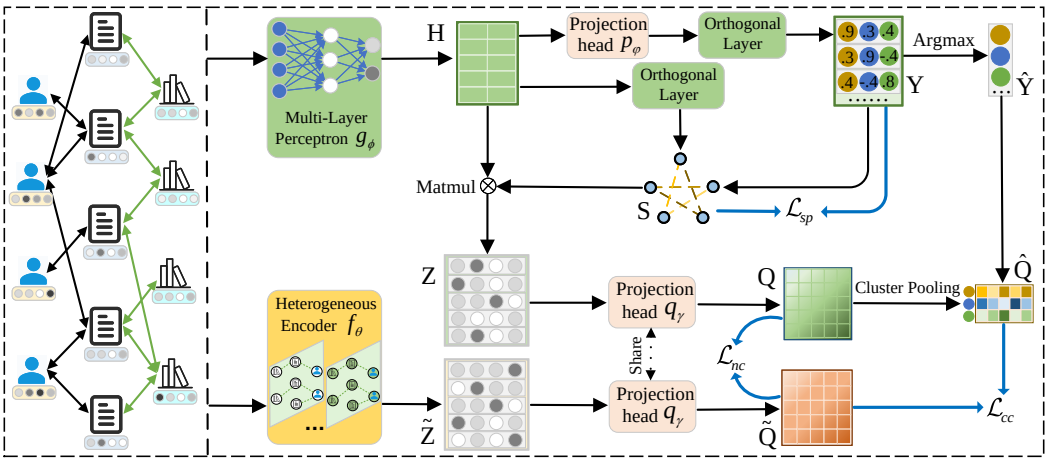


Figure 1: The flowchart of SCHOOL, which first employs the Multi-Layer Perception g_ϕ to derive semantic representations $\mathbf{H}$, followed by obtaining orthogonal cluster assignment matrix $\mathbf{Y}$ and orthogonal $\mathbf{H}$. Subsequently, SCHOOL filters noisy connections by deriving the rank-constrained affinity matrix $\mathbf{S}$, which is further used to multiply with $\mathbf{H}$ and then obtain node representations $\mathbf{Z}$. Meanwhile, SCHOOL employs a heterogeneous encoder f_θ to aggregate information across node types, yielding heterogeneous representations $\tilde{\mathbf{Z}}$. Finally, SCHOOL incorporates spectral loss $\mathcal{L}_{sp}$ to optimize $\mathbf{Y}$ to fit eigenvectors of the Laplacian matrix of $\mathbf{S}$. Moreover, SCHOOL designs node-level (*i.e.,* $\mathcal{L}_{nc}$) and cluster-level (*i.e.,* $\mathcal{L}_{cc}$) consistency constraints on projected representations (*i.e.,* $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$) and cluster representations $\hat{\mathbf{Q}}$ to capture the invariant and clustering information, respectively.

Despite the effectiveness of previous SHGL methods, they encounter two limitations. First, previous methods conduct message-passing relying on meta-path-based graphs and adaptive graph structures, which inevitably include noise, *i.e.,* connections among nodes from different classes [18, 58]. Consequently, such noise compromises the identifiability of node representations after the message-passing process. Second, while previous methods exhibit clustering characteristics, they typically emphasize the node-level consistency only, neglecting to capture and leverage the cluster-level information effectively [57, 30]. This may not fully exploit the potential benefits of clustering for representation learning, thereby diminishing the performance of downstream tasks.

Based on the above analysis, it is feasible to analyze previous SHGL methods from a clustering perspective thanks to their close connection to clustering techniques and further optimize the graph structures to mitigate noisy connections as well as harness the cluster-level information to enhance previous SHGL. To achieve this, there are three key challenges, *i.e.,* (i) How to formally understand previous SHGL methods from the clustering perspective? (ii) With insights from the clustering analysis, how to learn an adaptive graph structure that effectively captures intra-class connections while filtering out inter-class noise? (iii) How to enable the effective incorporation of cluster-level information in the heterogeneous graph to boost the performance of downstream tasks?

In this paper, we address the outlined challenges by first theoretically revisiting previous SHGL methods from a clustering perspective and then introducing a novel framework, termed Spectral Clustering-inspired HeterOgeneOus graph Learning (SCHOOL for short), that incorporates rank-constrained spectral clustering and dual consistency constraints, as depicted in Figure 1. Specifically, we start by proving that existing SHGL can be reformulated as spectral clustering with an additional regularization term under the assumption of orthogonality, thus addressing **challenge (i)** and laying the foundational theory for our approach. Next, to tackle **challenge (ii)**, we propose an efficient spectral clustering technique that includes a rank constraint on the affinity matrix, aiming to effectively mitigate noisy connections among different classes. Furthermore, to resolve **challenge (iii)**, we design dual consistency constraints at both node and cluster levels to capture invariant and clustering information, respectively, which reduces the intra-cluster differences and enhances the performance of downstream tasks. Finally, theoretical analysis indicates that the learned representations are divided into distinct partitions corresponding to the number of classes, and are demonstrated superior generalization ability compared to those derived from previous SHGL methods.

https://doi.org/10.52202/079017-1366

Compared to previous SHGL works, our contributions can be summarized as follows:

- To the best of our knowledge, we make the first attempt to theoretically revisit previous SHGL methods from the spectral clustering perspective in a unified manner.
- We adaptively learn a rank-constrained affinity matrix to mitigate noisy connections inherent in previous SHGL methods. Moreover, we introduce dual consistency constraints to capture both invariant and clustering information to enhance the effectiveness of our method.
- We theoretically demonstrate that the proposed method divides the learned representations into distinct partitions based on the number of classes, instead of dimensions in previous SHGL methods. Furthermore, the representations obtained by this method exhibit enhanced generalization ability compared to those derived from previous SHGL methods.
- We experimentally manifest the effectiveness of the proposed method across a variety of downstream tasks, using both heterogeneous and homogeneous graph datasets, compared to numerous state-of-the-art methods.

2 Method

Notations. Let $\mathbf{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{T}, \mathcal{R})$ represent a heterogeneous graph, where $\mathcal{V}$ and $\mathcal{E}$ indicate set of nodes and set of edges, respectively. $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ denotes the matrix of node features, where n indicates the number of nodes. Moreover, $\mathcal{T}$ and $\mathcal{R}$ indicate set of node types and set of edge types, respectively. Given the heterogeneous graph $\mathbf{G}$, most existing SHGL methods utilize meta-paths or adaptive graph structures to explore connections among nodes within the same class, thus exhibiting the characteristics of clustering and obtaining discriminative representations. To gain a deeper insight of previous SHGL methods, we first propose to revisit them from a clustering perspective as follows.

2.1 Revisiting Previous SHGL Methods from Spectral Clustering

As mentioned above, previous SHGL methods tend to conduct clustering implicitly, relying on meta-path-based graphs or adaptive graph structures. For example, given an academic heterogeneous graph with several node types (*i.e.,* paper, author, and subject), for the meta-path-based methods, if two papers belong to the same class, there may exist a meta-path "paper-subject-paper" to connect them (*i.e.,* two papers are grouped into the same subject). Similarly, for the adaptive-graph-based methods, when two papers belong to the same class, the adaptive graph structures likely assign large weights to connect them. Therefore, representations of nodes within the same class will be close to each other after the message-passing process, thus implicitly presenting a clustered pattern.

Based on the above observation, actually, we can further theoretically understand previous SHGL methods from the clustering perspective. To do this, we first give the following definition.

Definition 2.1. *(Spectral Clustering) Given the Laplacian matrix* $\mathbf{L}$*, the optimization problem of the spectral clustering can be described as follows:*

$$\min_{\mathbf{H}} \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \ \textit{s.t.,}\ \mathbf{H}^T\mathbf{H} = \mathbf{I}, \tag{1}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$*,* $\mathbf{W} \in \mathbb{R}^{n\times n}$ *is a data similarity matrix,* $\mathbf{D}$ *is a diagonal matrix whose entries are column sums of* $\mathbf{W}$*,* $\mathbf{H} \in \mathbb{R}^{n\times d}$ *is a representations matrix,* $\operatorname{Tr}(\cdot)$ *indicates the matrix trace, and* $\mathbf{I}$ *indicates the identity matrix.*

According to Definition 2.1, for both meta-path-based methods [31, 57, 58] and adaptive-graph-based SHGL methods [30], we then have Theorem 2.2, whose proof can be found in Appendix C.1.

Theorem 2.2. *Assume the learned representations* $\mathbf{H}$ *are orthogonal, optimizing previous meta-path-based and adaptive-graph-based SHGL methods is equivalent to performing spectral clustering with additional regularization, i.e.,*

$$\min_{\mathbf{H}} \mathcal{L}_{SHGL} \cong \min_{\mathbf{H}} \operatorname{Tr}(\mathbf{H}^T\hat{\mathbf{L}}\mathbf{H}) + R(\mathbf{H}) \ \textit{s.t.,}\ \mathbf{H}^T\mathbf{H} = \mathbf{I}, \tag{2}$$

where $R(\cdot)$ *indicates the regularization term,* $\hat{\mathbf{L}}$ *indicates the Laplacian matrix of the meta-path-based graph or the adaptive graph structure.*

https://doi.org/10.52202/079017-1366

Theorem 2.2 reveals the connection between previous SHGL and the spectral clustering as well as indicates that previous SHGL heavily relies on the Laplacian matrix of meta-path-based graph or adaptive graph structure. Moreover, based on Theorem 2.2, we can further bridge previous SHGL and the graph-cut algorithm [54] as follows, whose proof can be found in Appendix C.2.

Theorem 2.3. *Under the same assumption in Theorem 2.2, optimizing previous meta-path-based and adaptive-graph-based SHGL methods is approximate to performing the* $\mathrm{RatioCut}\left(V_1, \ldots, V_d\right)$ *algorithm that divides the learned representations into* d *partitions* $\{V_1, \ldots, V_d\}$*, i.e.,*

$$\min_{\mathbf{H}} \mathcal{L}_{SHGL} \cong \min_{\mathbf{H}} \mathrm{RatioCut}\left(V_1, \ldots, V_d\right), \tag{3}$$

where d *indicates the dimension of representations* $\mathbf{H}$*.*

Theorem 2.3 further indicates that previous SHGL methods divide the learned representations into d partitions, where d is generally much larger than the number of classes. Therefore, Theorem 2.3 connects the traditional graph-cut algorithm with existing SHGL methods, which requires custom analysis. As a result, we theoretically revisit previous SHGL methods from spectral clustering as well as graph-cut perspectives and build the connections between them, thus solving the challenge (i).

2.2 Rank-Constrained Spectral Clustering

Based on the connections between previous SHGL methods and the spectral clustering as well as the graph-cut algorithm, we have the observations as follows. First, according to Theorem 2.2, previous SHGL methods conduct spectral clustering based on the Laplacian matrix of meta-path-based graph or adaptive graph structure, which may not guarantee optimality and could potentially contain noisy connections, thus affecting the spectral clustering. Second, according to Theorem 2.3, previous SHGL methods conduct the graph-cut to divide the learned representations into d partitions, which are generally not equal to the number of classes c. As a result, optimizing previous SHGL methods becomes a hard or even error problem, and the learned representations can not be clustered well. Therefore, it is intuitive to mitigate noise in the adaptive graph structure as well as divide the learned representations into exactly c partitions to improve existing SHGL methods.

Specifically, in this paper, we propose to learn an adaptive affinity matrix with the rank constraint to mitigate noisy connections as much as possible. To do this, we first employ the Multi-Layer Perceptron (MLP) as the encoder $g_\phi \in \mathbb{R}^{f \times d_1}$ to obtain the semantic representations $\mathbf{H}$ by:

$$\mathbf{H} = \sigma(g_\phi(\mathbf{X})), \tag{4}$$

where f and d_1 are the dimensions of node features and representations, respectively, and σ is the activation function. After that, we propose to learn an adaptive affinity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ based on the semantic representations. Intuitively, in a uncorrelated representations subspace, a smaller distance $\|\mathbf{h}_i - \mathbf{h}_j\|_2^2$ between semantic representations should be assigned a larger probability $\mathbf{s}_{ij}$. Therefore, it is a natural approach to learn the affinity matrix $\mathbf{S}$ by:

$$\min_{\mathbf{S}} \textstyle\sum_{i,j=1}^{n} (\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 \mathbf{s}_{ij} + \alpha \mathbf{s}_{ij}^2) \ \text{ s.t.}, \forall i, \mathbf{s}_i^T \mathbf{1} = 1, 0 \leq \mathbf{s}_i \leq 1, \tag{5}$$

where α is a non-negative parameter. In Eq. (5), the first term encourages the affinity matrix to assign large weights to node pairs with small distances. Moreover, the second term avoids the trivial solution that only the nearest node can be the neighbor of v_i with probability 1. However, similar to previous SHGL methods, usually the affinity matrix learned by Eq. (5) can not reach the ideal case (*i.e.,* having no noisy connections among different classes and containing exactly c connected components). As a result, the noisy connections in the affinity matrix may induce a negative interference during the message-passing process. To solve this issue, we first introduce the following lemma in [32].

Lemma 2.4. *The multiplicity* c *of the eigenvalue 0 of the Laplacian matrix* $\mathbf{L_S}$ *is equal to the number of connected components in the affinity matrix* $\mathbf{S}$*.*

Lemma 2.4 indicates that if the rank of $\mathbf{L_S}$ equals to $n - c$, then the affinity matrix $\mathbf{S}$ contains exactly c connected components to achieve the ideal scenario, where $\mathbf{L_S} = \mathbf{D} - \frac{\mathbf{S}+\mathbf{S}^T}{2}$, and $\mathbf{D}$ is the degree matrix of $\frac{\mathbf{S}+\mathbf{S}^T}{2}$. Based on Lemma 2.4, we can solve the above issue by adding the rank constraint on the affinity matrix, *i.e.,* enforcing the smallest c eigenvalues of $\mathbf{L_S}$ to be 0:

$$\mathrm{rank}(\mathbf{L_S}) = n - c \Rightarrow \min \sum_{i=1}^{c} \tau_i \left(\mathbf{L_S}\right), \tag{6}$$

where $\tau_i(\mathbf{L_S})$ is the i-th smallest eigenvalue of $\mathbf{L_S}$ and $\tau_i(\mathbf{L_S}) \geq 0$ because $\mathbf{L_S}$ is positive semi-definite. Moreover, according to Ky Fan's Theorem [5], the constraint in Eq. (6) can be rewritten in the spectral clustering form as follows (derivation listed in Appendix C.5).

$$\sum_{i=1}^{c} \tau_i \left(\mathbf{L_S}\right) = \min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L_S}\mathbf{F}) = \min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \frac{1}{2} \sum_{i,j} \mathbf{s}_{ij} \left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2, \tag{7}$$

where $\mathbf{F} \in \mathbb{R}^{n\times c}$ is formed by part eigenvectors of $\mathbf{L_S}$ after the eigendecomposition. Therefore, based on Eq. (7), we can add the rank constraint on the affinity matrix and reformulate Eq. (5) as:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{F}} \textstyle\sum_{i,j=1}^{n} (\left\|\mathbf{h}_i - \mathbf{h}_j\right\|_2^2 \mathbf{s}_{ij} + \alpha \mathbf{s}_{ij}^2 + \beta \left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2 \mathbf{s}_{ij}) \\ &\text{s.t.}, \forall i, \mathbf{s}_i^T\mathbf{1} = 1, 0 \leq \mathbf{s}_i \leq 1, \mathbf{F}^T\mathbf{F} = \mathbf{I}, \end{aligned} \tag{8}$$

where β is a non-negative parameter. Eq. (8) can be solved by applying the alternating optimization approach. Specifically, when $\mathbf{S}$ is fixed, Eq. (8) becomes $\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \sum_{i,j=1}^{n} \left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2 \mathbf{s}_{ij} = 2\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L_S}\mathbf{F})$, whose optimal solution $\mathbf{F}$ is the eigenvectors of $\mathbf{L_S}$ corresponding to the c smallest eigenvalues. When $\mathbf{F}$ is fixed, denote $\mathbf{d}_{ij} = \left\|\mathbf{h}_i - \mathbf{h}_j\right\|_2^2 + \beta \left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2$, Eq. (8) can be written in the vector form, *i.e.,*

$$\min_{\mathbf{s}_i^T\mathbf{1}=1, 0\leq\mathbf{s}_i\leq 1,} \left\|\mathbf{s}_i + \frac{1}{2\alpha}\mathbf{d}_i\right\|_2^2. \tag{9}$$

According to the Lagrangian function of Eq. (9) and the KKT condition [3], we can obtain the closed-form solution of the element $\mathbf{s}_{ij}$ in the affinity matrix, *i.e.,*

$$\mathbf{s}_{ij} = (-\frac{1}{2\alpha}\mathbf{d}_{ij} + \lambda)_+, \tag{10}$$

where λ is a non-negative parameter, and $(\cdot)_+$ indicates $\max\{\cdot, 0\}$. In practice, a sparse affinity matrix usually obtains better performance and reduces computation costs. Therefore, we only calculate $\mathbf{s}_{ij}$ between node v_i and its k nearest neighbors. Then parameters α and λ can be further determined, details listed in Appendix C.6.

However, when fixing $\mathbf{S}$ and optimizing $\mathbf{F}$ in Eq. (8), computation costs of obtaining eigenvectors $\mathbf{F}$ remain prohibitive due to the cubic time complexity of the eigendecomposition. To address this issue, we propose to replace the eigendecomposition with a projection head and orthogonal layer [42]. Specifically, we first employ the projection head $p_\varphi \in \mathbb{R}^{d_1\times c}$ to map semantic representations to the cluster assignment space $\mathbf{P} \in \mathbb{R}^{n\times c}$, *i.e.,*

$$\mathbf{P} = \sigma(p_\varphi(\mathbf{H})), \tag{11}$$

where σ is the activation function. After that, we employ an orthogonal layer to derive the orthogonal cluster assignment matrix $\mathbf{Y} \in \mathbb{R}^{n\times c}$ (orthogonal derivation listed in Appendix C.7), *i.e.,*

$$\mathbf{Y} = \sqrt{n}\mathbf{P}\left(\mathbf{R}^{-1}\right), \tag{12}$$

where $\mathbf{R}$ is an upper triangular matrix obtained from the QR decomposition [6] (*i.e.,* $\mathbf{P} = \mathbf{E}\mathbf{R}$ and $\mathbf{E}^T\mathbf{E} = \mathbf{I}$) on the full rank $\mathbf{P}$. Similarly, we can also implement the orthogonal layer to achieve the uncorrelation (*i.e.,* $\mathbf{H}^T\mathbf{H} = \mathbf{I}$) on the representations subspace.

As a result, the projection head and the orthogonal layer act as a linear transformation to achieve the orthogonality constraint on $\mathbf{Y}$. To replace the eigendecomposition, the cluster assignment matrix $\mathbf{Y}$ should further fit the eigenvectors $\mathbf{F}$ in Eq. (8). To do this, we design a spectral loss $\mathcal{L}_{sp}$ to optimize the parameters in p_φ to simulate the spectral clustering of the third term in Eq. (8), *i.e.,*

$$\mathcal{L}_{sp} = \frac{1}{n^2} \textstyle\sum_{i,j=1}^{n} \mathbf{s}_{ij} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2 - \gamma H(\mathbf{Y}), \tag{13}$$

where γ is a non-negative parameter, $H(\mathbf{Y}) = -\sum_{i=1}^{c} P(\mathbf{y}^i)\log P(\mathbf{y}^i)$ is the entropy of cluster assignment probabilities $P(\mathbf{y}^i) = \frac{1}{n}\sum_{j=1}^{n}\mathbf{y}_j^i$, and $\mathbf{y}_j^i$ indicates the i-th column and j-th row of $\mathbf{y}$. According to Eq. (7), the first term in Eq. (13) simulates the spectral clustering to enforce the learned cluster assignment matrix $\mathbf{Y}$ to approximate eigenvectors $\mathbf{F}$, and the second term is a widely used regularization term [1, 14] to avoid the trivial solution that most nodes are assigned to the same cluster. Therefore, when $\mathbf{S}$ is fixed, we optimizing $\mathbf{Y}$ to approach the optimal $\mathbf{F}$ by achieving orthogonality

https://doi.org/10.52202/079017-1366

with Eq. (12), and fitting eigenvectors $\mathbf{F}$ with Eq. (13). As a result, this reduces the cubic time complexity of eigendecomposition to $\mathcal{O}(nd^2 + nc^2 + nkd + nkc + c^3 + d^3)$, where $d^2, c^2 < n$, thus is linear to the sample size, details are shown in Appendix B.2. Finally, the proposed method still optimizes the affinity matrix $\mathbf{S}$ and eigenvectors $\mathbf{F}$ in Eq. (8) in an alternating approach, *i.e.,* fix $\mathbf{F}$ and then obtain the closed-form solution of $\mathbf{S}$, and fix $\mathbf{S}$ and then optimize parameters (*i.e.,* g_ϕ and p_φ) to update $\mathbf{Y}$ to approach the optimal $\mathbf{F}$.

Therefore, the proposed method obtains the affinity matrix with exactly c connected components to mitigate noisy connections in an effective and efficient way. Then we can obtain the node representations $\mathbf{Z} = \mathbf{SH}$, which is expected to conduct message-passing among nodes within the same class. Moreover, we can bridge the connections between the proposed method and the spectral clustering as well as the graph-cut algorithm as follows, whose proof can be found in Appendix C.3.

Theorem 2.5. *Optimizing the spectral loss $\mathcal{L}_{sp}$ leads to performing the spectral clustering based on the affinity matrix $\mathbf{S}$ with c connected components and conducting RatioCut $(V_1, \ldots, V_c)$ algorithm to divide the learned representations into c partitions, i.e.,*

$$\min \mathcal{L}_{sp} \Rightarrow \min \mathrm{Tr}(\mathbf{Y}^T \mathbf{L}_\mathbf{S} \mathbf{Y}) \Rightarrow \min \mathrm{RatioCut}(V_1, \ldots, V_c). \tag{14}$$

Theorem 2.5 indicates that the proposed method conducts the spectral clustering as previous SHGL methods, but is performed on an affinity matrix with exactly c connected components (verified in Section 3.2.3), thus mitigating noisy connections from different classes and solving the challenge (ii). Moreover, the proposed method divides the learned representations into c partitions, which is a better optimization goal than previous SHGL methods to obtain discriminative representations.

2.3 Dual Consistency Constraints

The message-passing among nodes within the same class reduces intra-class differences and enhances node representations $\mathbf{Z}$. Meanwhile, the message-passing among nodes from different types also contributes to obtaining task-related contents and benefits downstream tasks [69]. To do this, we propose to aggregate the information of nodes from different types in the heterogeneous graph with a heterogeneous encoder $f_\theta \in \mathbb{R}^{f \times d_1}$.

Specifically, for the node v_i, we concatenate the information of itself and its relevant one-hop neighbors (*i.e.,* nodes of other types) based on edge types in $\mathcal{R}$, and then derive the heterogeneous representations $\tilde{\mathbf{Z}}$ by:

$$\tilde{\mathbf{z}}_i = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \sigma(f_\theta(\mathbf{x}_i) || \sum_{v_j \in \mathcal{N}_{i,r}} f_\theta(\mathbf{x}_j)), \tag{15}$$

where σ is the activation function, $|\mathcal{R}|$ indicates the number of edge types, $\mathcal{N}_{i,r}$ indicates the set of one-hop neighbors of node v_i based on the edge type $r \in \mathcal{R}$, $f_\theta(\cdot)$ indicates the linear transformation, and $\cdot||\cdot$ indicates the concatenation operation. Therefore, the heterogeneous representations $\tilde{\mathbf{Z}}$ aggregate the information of nodes from different types to introduce more task-related contents.

Given node representations $\mathbf{Z}$ and heterogeneous representations $\tilde{\mathbf{Z}}$, most previous SHGL methods utilize the node-level consistency constraint (*e.g.,* Info-NCE loss [36]) to capture the invariant information between them and enhance the effectiveness [57, 30]. In addition, according to Theorem 2.2, previous SHGL methods actually perform spectral clustering to learn node representations. However, previous SHGL methods fail to utilize the cluster-level information outputted by the spectral clustering, thus weakening the downstream task performance. To solve this issue, we design dual consistency constraints to capture the invariant information as well as the clustering information between $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$.

Specifically, we first employ a projection head $q_\gamma \in \mathbb{R}^{d_1 \times d_2}$ to map both $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$ into the same latent space, *i.e.,* $\mathbf{Q} = q_\gamma(\mathbf{Z})$ and $\tilde{\mathbf{Q}} = q_\gamma(\tilde{\mathbf{Z}})$, where d_2 is the projected dimension. Then we follow previous works [57, 58] to design a node-level consistency constraint to capture the invariant information between $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$, *i.e.,*

$$\mathcal{L}_{nc} = \|\mathbf{Q} - \tilde{\mathbf{Q}}\|_F^2 + \eta \log \textstyle\sum_{i,j=1}^{d} e^{\mathbf{c}_{ij}}, \tag{16}$$

where $\mathbf{C} = \mathbf{Q}^T\mathbf{Q} + \tilde{\mathbf{Q}}^T\tilde{\mathbf{Q}}$, and η is a non-negative parameter. Similar to previous works, the first term in Eq. (16) enforces representations in $\tilde{\mathbf{Q}}$ agree with the corresponding representations in $\mathbf{Q}$, thus capturing the invariant information between them. The second term enables different dimensions of $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$ to uniformly distribute over the latent space to avoid collapse.

In addition to the node-level consistency constraint, we further design the cluster-level consistency constraint to capture the clustering information from the cluster assignment matrix $\mathbf{Y}$. To do this, we first obtain the cluster indicator matrix $\hat{\mathbf{Y}}$ based on the cluster assignment matrix $\mathbf{Y}$, *i.e.,* $\hat{\mathbf{Y}} = \mathrm{argmax}(\mathbf{Y})$. After that, we conduct average pooling on node representations that possess the same cluster indicator to obtain the j-th cluster representation $\hat{\mathbf{q}}_j$, *i.e.,*

$$\hat{\mathbf{q}}_j = \frac{1}{|V_j|}\sum\nolimits_{v_i \in V_j} \mathbf{q}_i, \tag{17}$$

where V_j indicates the set of nodes whose cluster indicators equal to j, and $|V_j|$ indicates the number of nodes in V_j. Then we design a cluster-level consistency constraint on cluster representations $\hat{\mathbf{Q}}$ and projected representations $\tilde{\mathbf{Q}}$ to capture the clustering information, *i.e.,*

$$\mathcal{L}_{cc} = \sum\nolimits_{i=1}^{n} \|\tilde{\mathbf{q}}_i - \hat{\mathbf{q}}_{\mathbf{y}_i}\|_2^2, \tag{18}$$

where $\hat{\mathbf{q}}_{\mathbf{y}_i}$ indicates the cluster representation whose label equals to $\mathbf{y}_i$. Eq. (18) enables the projected representation $\tilde{\mathbf{q}}_i$ and the cluster representation $\hat{\mathbf{q}}_{\mathbf{y}_i}$ to align each other. As a result, representations capture the clustering information based on cluster indicators and reduce intra-cluster differences to improve the performance of downstream tasks, thus solving challenge (iii).

We integrate the spectral loss in Eq. (13), the node-level consistency constraint in Eq. (16), with the cluster-level consistency constraint in Eq. (18) to have the objective function as:

$$\mathcal{J} = \mathcal{L}_{sp} + \mu\mathcal{L}_{nc} + \delta\mathcal{L}_{cc}, \tag{19}$$

where μ and δ are non-negative parameters. Finally, we concatenate node representations $\mathbf{Z}$ with heterogeneous representations $\tilde{\mathbf{Z}}$ to obtain representations for downstream tasks. Actually, for the learned representations, we have the following Theorem, whose proof can be found in Appendix C.4.

Theorem 2.6. *The proposed method with dual consistency constraints achieves a lower boundary of the model complexity C and a higher generalization ability boundary G than previous SHGL with the node-level consistency constraint only, i.e.,*

$$\inf(C_{SCHOOL}) < \inf(C_{SHGL}), \quad \sup(G_{SCHOOL}) > \sup(G_{SHGL}), \tag{20}$$

where $\inf(\cdot)$ *and* $\sup(\cdot)$ *indicates lower bound and upper bound, respectively.*

Theorem 2.6 indicates that the representations learned by the dual consistency constraints can be theoretically proved to exhibit superior generalization ability than the representations learned by previous SHGL methods with the node-level consistency constraint only, thus are expected to perform better in different downstream tasks (verified in Section 3.2).

3 Experiments

In this section, we conduct experiments on both heterogeneous and homogeneous graph datasets to evaluate the proposed method in terms of different downstream tasks (*i.e.,* node classification and node clustering), compared to both heterogeneous and homogeneous graph methods. Detailed settings are shown in Appendix D, and additional results are shown in Appendix E.

3.1 Experimental Setup

3.1.1 Datasets

The used datasets include four heterogeneous graph datasets and two homogeneous graph datasets. Heterogeneous graph datasets include three academic datasets (*i.e.,* ACM [56], DBLP [56], and Aminer [11]), and one business dataset (*i.e.,* Yelp [27]). Homogeneous graph datasets include two sale datasets (*i.e.,* Photo and Computers [43]).

https://doi.org/10.52202/079017-1366

Table 1: Classification performance (*i.e.,* Macro-F1 and Micro-F1) on heterogeneous graph datasets.

Method	ACM		Yelp		DBLP		Aminer	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
DeepWalk	73.9±0.3	74.1±0.1	68.7±1.1	73.2±0.9	88.1±0.2	89.5±0.3	54.7±0.8	59.7±0.7
GCN	86.9±0.2	87.0±0.3	85.0±0.6	87.4±0.8	90.2±0.2	90.9±0.5	64.5±0.7	71.5±0.9
GAT	85.0±0.4	84.9±0.3	86.4±0.5	88.2±0.7	91.0±0.4	92.1±0.2	63.8±0.4	70.6±0.7
Mp2vec	87.6±0.5	88.1±0.3	78.2±0.8	83.6±0.9	85.7±0.3	87.6±0.6	58.7±0.5	65.3±0.6
HAN	89.4±0.2	89.2±0.2	90.5±1.2	90.7±1.4	91.2±0.4	92.0±0.5	65.3±0.7	72.8±0.4
HGT	91.5±0.7	91.6±0.6	89.9±0.5	90.2±0.6	90.9±0.6	91.7±0.8	64.5±0.5	71.0±0.7
DMGI	89.8±0.1	89.8±0.1	82.9±0.8	85.8±0.9	92.1±0.2	92.9±0.3	63.8±0.4	67.6±0.5
DMGIattn	88.7±0.3	88.7±0.5	82.8±0.7	85.4±0.5	90.9±0.2	91.8±0.3	62.4±0.9	66.8±0.8
HDMI	90.1±0.3	90.1±0.3	80.7±0.6	84.0±0.9	91.3±0.2	92.2±0.5	65.9±0.4	71.7±0.6
HeCo	88.3±0.3	88.2±0.2	85.3±0.7	87.9±0.6	91.0±0.3	91.6±0.2	71.8±0.9	78.6±0.7
HGCML	90.6±0.7	90.7±0.5	90.7±0.8	91.0±0.7	91.9±0.8	93.2±0.7	70.5±0.4	76.3±0.6
CPIM	91.4±0.3	91.3±0.2	90.2±0.5	90.3±0.4	93.2±0.6	93.8±0.8	70.1±0.9	75.8±1.1
HGMAE	90.5±0.5	90.6±0.7	90.5±0.7	90.7±0.5	92.9±0.5	93.4±0.6	72.3±0.9	80.3±1.2
HERO	92.2±0.5	92.1±0.7	92.4±0.7	92.3±0.6	93.8±0.6	94.4±0.4	75.1±0.7	84.5±0.9
SCHOOL	**92.7±0.6**	**92.6±0.5**	**93.0±0.7**	**92.8±0.4**	**94.0±0.3**	**94.7±0.4**	**77.5±0.9**	**86.8±0.7**

3.1.2 Comparison Methods

The comparison methods include eleven heterogeneous graph methods and twelve homogeneous graph methods. The former includes two semi-supervised methods (*i.e.,* HAN [56] and HGT [13]), one traditional unsupervised method (*i.e.,* Mp2vec [4]), and eight self-supervised methods (*i.e.,* DMGI [38], DMGIattn [38], HDMI [18], HeCo [57], HGCML [58], CPIM [31], HGMAE [48], and HERO [30]). The latter includes two semi-supervised methods (*i.e.,* GCN [20] and GAT [50]), one traditional unsupervised method (*i.e.,* DeepWalk [41]), and nine self-supervised methods, (*i.e.,* DGI [51], GMI [40], MVGRL [8], GRACE [75], GCA [76], G-BT [2], COSTA [70], DSSL [61], and LRD [63]).

For a fair comparison, we follow [4, 56, 27, 28] to select meta-paths for previous meta-path-based SHGL methods. Moreover, we follow [29] to implement homogeneous graph methods on heterogeneous graph datasets by separately learning the representations of each meta-path-based graph and further concatenating them for downstream tasks. In addition, we replace the heterogeneous encoder f_θ with GCN to implement the proposed method on homogeneous graph datasets because there is only one node type in the homogeneous graph. Moreover, we follow previous works [76] to generate two different views for the homogeneous graph by removing edges and masking features. The code of the proposed method is released at https://github.com/YujieMo/SCHOOL.

3.2 Results Analysis

3.2.1 Effectiveness on Heterogeneous and Homogeneous Graph

We first evaluate the effectiveness of the proposed method on the heterogeneous graph datasets and report the results of node classification and node clustering in Table 1 and Appendix E, respectively. Obviously, the proposed method obtains better performance on both node classification and node clustering tasks than comparison methods.

Specifically, first, for the node classification task, the proposed method consistently outperforms the comparison methods by large margins. For example, the proposed method on average, improves by 1.1%, compared to the best SHGL method (*i.e.,* HERO), on four heterogeneous graph datasets. The reason can be attributed to the fact that the proposed method adaptively learns a rank-constrained affinity matrix to mitigate noisy connections among different classes, thus reducing intra-class differences. Second, for the node clustering task, the proposed method also obtains promising improvements. For example, the proposed method on average, improves by 3.1%, compared to the best SHGL method (*i.e.,* HGMAE), on four heterogeneous graph datasets. This demonstrates the superiority of the proposed method, which simulates the spectral clustering with the spectral loss and conducts the cluster-level consistency constraint to further utilize the clustering information. As a result, the effectiveness of the proposed method is verified on different downstream tasks.

Table 2: Classification performance (*i.e.,* Macro-F1 and Micro-F1) of each component in the objective function $\mathcal{J}$ on all heterogeneous graph datasets.

$\mathcal{L}_{sp}$	$\mathcal{L}_{nc}$	$\mathcal{L}_{cc}$	ACM		Yelp		DBLP		Aminer	
			Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
–	–	✓	85.9±0.3	85.8±0.6	91.8±0.6	91.3±0.5	91.0±0.2	92.1±0.4	66.8±0.7	75.5±0.9
–	✓	–	88.8±0.6	88.6±0.7	92.5±0.7	92.1±0.4	91.7±0.4	92.7±0.5	72.4±0.5	80.3±0.7
✓	–	–	87.6±0.3	87.5±0.5	92.3±0.8	92.0±0.6	90.7±0.6	91.7±0.6	67.3±0.6	74.7±0.5
–	✓	✓	86.9±0.7	86.7±0.5	92.1±0.3	91.5±0.4	93.4±0.8	94.2±0.6	75.2±0.4	83.9±0.7
✓	–	✓	89.0±0.5	88.9±0.4	92.4±0.5	92.0±0.3	93.5±0.6	94.2±0.4	76.2±0.5	85.2±0.8
✓	✓	–	88.9±0.7	88.8±0.6	92.6±0.6	92.3±0.5	91.9±0.7	92.8±0.8	77.1±0.8	86.0±0.6
✓	✓	✓	**92.7±0.6**	**92.6±0.5**	**93.0±0.7**	**92.8±0.4**	**94.0±0.3**	**94.7±0.4**	**77.5±0.9**	**86.8±0.7**

We further evaluate the effectiveness of the proposed method on homogeneous graph datasets and report the results of node classification in Appendix E. We can observe that the proposed method also achieves competitive results on the homogeneous graph datasets compared to other homogeneous graph methods. For example, the proposed method outperforms the best self-supervised method (*i.e.,* LRD), on almost all homogeneous graph datasets. This indicates that the proposed method is also available to learn the noise-free affinity matrix for homogeneous graphs as well as capture invariant and clustering information to benefit downstream tasks. Therefore, the effectiveness of the proposed method is verified on both heterogeneous and homogeneous graph datasets.

3.2.2 Ablation Study

The proposed method investigates the objective function $\mathcal{J}$ to learn the rank-constrained affinity matrix, as well as capture invariant and clustering information. To verify the effectiveness of each component of $\mathcal{J}$ (*i.e.,* $\mathcal{L}_{sp}$, $\mathcal{L}_{nc}$, and $\mathcal{L}_{cc}$), we investigate the performance of all variants on the node classification task and report the results in Table 2.

From Table 2, we have the observations as follows. First, the proposed method with the complete objective function obtains the best performance. For example, the proposed method on average improves by 1.8%, compared to the best variant (*i.e.,* without $\mathcal{L}_{nc}$), indicating that all components in the objective function are necessary for the proposed method. This is consistent with our claims, *i.e.,* it is essential to optimize the adaptive graph structure to mitigate noisy connections as well as utilize the cluster-level information to benefit downstream tasks. Second, the variant without $\mathcal{L}_{sp}$ achieves inferior results to the other two variants (*i.e.,* without $\mathcal{L}_{nc}$ and without $\mathcal{L}_{cc}$, respectively). This can be attributed to the fact that the spectral loss $\mathcal{L}_{sp}$ enforces the cluster assignment matrix to fit the eigenvectors, which is necessary for the closed-form solution of the affinity matrix.

3.2.3 Visualization

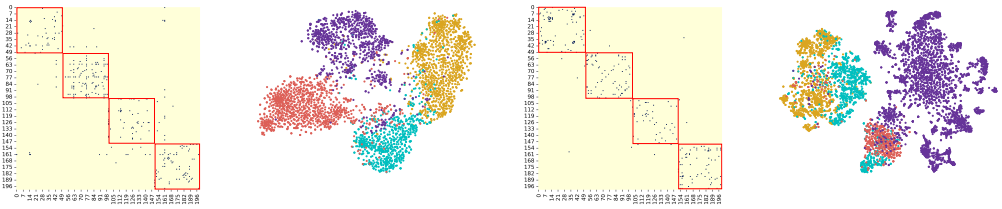
To verify the effectiveness of the learned affinity matrix and the representations for downstream tasks, we visualize the affinity matrix in the heatmap and visualize the representations with t-SNE [49] on DBLP and Aminer datasets and report the results in Figure 2.

Specifically, we randomly sample 50 nodes in each class and then visualize elements of the affinity matrix $\mathbf{S}$ among sampled nodes with the heatmap, where rows and columns are reordered by node labels. In the correlation map, the darker a pixel, the larger the value of the element of $\mathbf{S}$. In Figures 2(a) and 2(c), the heatmaps exhibit that there are nearly c (*i.e.,* the number of classes) components in the affinity matrix, and almost all elements with large values fall in the block diagonal structure. This indicates that the affinity matrix indeed contains c connected components to mitigate noisy connections among different classes. Moreover, the t-SNE visualization in Figures 2(b) and 2(d) further indicate that the learned representations can be well divided into c partitions. This is consistent with the observation in Theorem 2.5 and verifies the effectiveness of the learned representations.

4 Conclusion

In this paper, we revisited previous SHGL methods from the perspective of spectral clustering and then introduced a novel framework to alleviate existing issues. Specifically, we first proved that optimizing

https://doi.org/10.52202/079017-1366



(a) $\mathbf{S}$ of DBLP (b) t-SNE of DBLP (c) $\mathbf{S}$ of Aminer (d) t-SNE of Aminer

Figure 2: Visualization of the affinity matrix $\mathbf{S}$ and t-SNE on DBLP and Aminer datasets.

previous SHGL methods is equivalent to performing spectral clustering with additional regularization under the orthogonalization assumption. Then we proposed an efficient spectral clustering method with the rank constraint to learn an adaptive affinity matrix and mitigate noisy connections in previous methods. Moreover, we designed node-level and cluster-level consistency constraints to capture invariant and clustering information, thus benefiting the performance of downstream tasks. Theoretical analysis indicates that the learned representations are divided into distinct partitions based on the number of classes, and are expected to achieve better generalization ability than representations of previous SHGL methods. Comprehensive experiments verify the effectiveness of the proposed method on both homogeneous and heterogeneous graph datasets on different downstream tasks.

Potential limitations and broader impact. Our potential limitation is that this work is designed based on node features. However, in heterogeneous graphs, instances arise where nodes are devoid of features. While one-hot vectors or structural embeddings can be designated as node features to tackle this problem, we recognize the necessity of devising dedicated techniques tailored for heterogeneous graphs with missing node features. In addition, the proposed method can also be used to deal with the homophily problem, which aims to explore the connections within the same class. We consider these aspects as potential directions for future research. Despite the great development of SHGL, some theoretical foundations are still lacking. Our work theoretically connects existing SHGL methods and spectral clustering and may open a new path to understanding and designing SHGL. Besides that, we do not foresee any direct negative impacts on the society.

Acknowledgments

This project is supported by the National Key Research and Development Program of China under Grant No. 2022YFA1004100, the Natural Science Foundation of Guangdong Province of China under Grant No. 2024A1515011381, and the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-023).

References

[1] Liang Bai and Jiye Liang. Sparse subspace clustering with entropy-norm. In *ICML*, pages 561–568, 2020.

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

[3] S Boyd, L Vandenberghe, and L Faybusovich. Convex optimization. *IEEE Transactions on Automatic Control*, 51(11):1859–1859, 2006.

[4] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*, pages 135–144, 2017.

[5] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[6] Walter Gander. Algorithms for the qr decomposition. *Res. Rep*, 80(02):1251–1268, 1980.

[7] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *NeurIPS*, pages 5000–5011, 2021.

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

[9] Dongxiao He, Chundong Liang, Cuiying Huo, Zhiyong Feng, Di Jin, Liang Yang, and Weixiong Zhang. Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[10] Dongxiao He, Jitao Zhao, Rui Guo, Zhiyong Feng, Di Jin, Yuxiao Huang, Zhen Wang, and Weixiong Zhang. Contrastive learning meets homophily: two birds with one stone. In *ICML*, pages 12775–12789, 2023.

[11] Binbin Hu, Yuan Fang, and Chuan Shi. Adversarial learning on heterogeneous information networks. In *SIGKDD*, pages 120–129, 2019.

[12] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *SIGKDD*, pages 1531–1540, 2018.

[13] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *WWW*, pages 2704–2710, 2020.

[14] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *CVPR*, pages 8849–8858, 2020.

[15] Yudi Huang, Yujie Mo, Yujing Liu, Ci Nie, Guoqiu Wen, and Xiaofeng Zhu. Multiplex graph representation learning via bi-level optimization. In *IJCAI*, 2024.

[16] Yiding Jiang, Parth Natekar, Manik Sharma, Sumukh K Aithal, Dhruva Kashyap, Natarajan Subramanyam, Carlos Lassance, Daniel M Roy, Gintare Karolina Dziugaite, Suriya Gunasekar, et al. Methods and analysis of the first competition in predicting generalization of deep learning. In *NeurIPS*, pages 170–190, 2021.

[17] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. Dual intent enhanced graph neural network for session-based new item recommendation. In *WWW*, pages 684–693, 2023.

[18] Baoyu Jing, Chanyoung Park, and Hanghang Tong. Hdmi: High-order deep multiplex infomax. In *WWW*, pages 2414–2424, 2021.

[19] P. Diederik Kingma and Lei Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[20] N. Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, pages 1–14, 2017.

[21] Marc T Law, Raquel Urtasun, and Richard S Zemel. Deep spectral clustering learning. In *ICML*, pages 1985–1994, 2017.

[22] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multimodal. *arXiv preprint arXiv:2212.05767*, 2022.

[23] Meng Liu, Ke Liang, Yawei Zhao, Wenxuan Tu, Sihang Zhou, Xinbiao Gan, Xinwang Liu, and Kunlun He. Self-supervised temporal graph learning with temporal and structural intensity alignment. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[24] Meng Liu, Yue Liu, Ke Liang, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. Deep temporal graph clustering. In *ICLR*, 2024.

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

[26] Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Stan Z Li. Dink-net: Neural clustering on large graphs. In *ICML*, 2023.

[27] Yuanfu Lu, Chuan Shi, Linmei Hu, and Zhiyuan Liu. Relation structure-aware heterogeneous information network embedding. In *AAAI*, pages 4456–4463, 2019.

[28] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *SIGKDD*, pages 1150–1160, 2021.

 https://doi.org/10.52202/079017-1366

[29] Yujie Mo, Yajie Lei, Jialie Shen, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. Disentangled multiplex graph representation learning. In *ICML*, pages 24983–25005, 2023.

[30] Yujie Mo, Feiping Nie, Zheng Zhang, Ping Hu, Heng Tao Shen, Xinchao Wang, and Xiaofeng Zhu. Self-supervised heterogeneous graph learning: a homophily and heterogeneity view. In *ICLR*, 2024.

[31] Yujie Mo, Zongqian Wu, Yuhuan Chen, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. Multiplex graph representation learning via common and private information mining. In *AAAI*, pages 9217–9225, 2023.

[32] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

[33] Parth Natekar and Manik Sharma. Representation based complexity measures for predicting generalization in deep learning. *arXiv preprint arXiv:2012.02775*, 2020.

[34] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *SIGKDD*, pages 977–986, 2014.

[35] Feiping Nie, Danyang Wu, Rong Wang, and Xuelong Li. Self-weighted clustering with adaptive neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, pages 3428–3441, 2020.

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

[37] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. In *NeurIPS*, volume 34, pages 2148–2159, 2021.

[38] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. Unsupervised attributed multiplex network embedding. In *AAAI*, pages 5371–5378, 2020.

[39] Liang Peng, Yujie Mo, Jie Xu, Jialie Shen, Xiaoshuang Shi, Xiaoxiao Li, Heng Tao Shen, and Xiaofeng Zhu. Grlc: Graph representation learning with constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

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

[41] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, pages 701–710, 2014.

[42] Uri Shaham, Kelly Stanton, Henry Li, Ronen Basri, Boaz Nadler, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. In *ICLR*, 2018.

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

[44] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[45] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, pages 17–37, 2016.

[46] Zhiquan Tan, Yifan Zhang, Jingqin Yang, and Yang Yuan. Contrastive learning is spectral clustering on similarity graph. In *ICLR*, 2024.

[47] Chang Tang, Zhenglai Li, Jun Wang, Xinwang Liu, Wei Zhang, and En Zhu. Unified one-step multi-view spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):6449–6460, 2022.

[48] Yijun Tian, Kaiwen Dong, Chunhui Zhang, Chuxu Zhang, and Nitesh V. Chawla. Heterogeneous graph masked autoencoders. In *AAAI*, pages 9997–10005, 2023.

[49] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[50] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, pages 1–12, 2018.

[51] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *ICLR*, pages 1–17, 2019.

[52] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

[53] Qi Wang, Zequn Qin, Feiping Nie, and Xuelong Li. Spectral embedded adaptive neighbors clustering. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1265–1271, 2018.

[54] Song Wang and Jeffrey Mark Siskind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2003.

[55] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and S Yu Philip. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9(2):415–436, 2022.

[56] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019.

[57] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *SIGKDD*, pages 1726–1736, 2021.

[58] Zehong Wang, Qi Li, Donghua Yu, Xiaolong Han, Xiao-Zhi Gao, and Shigen Shen. Heterogeneous graph contrastive multi-view learning. In *SDM*, pages 136–144, 2023.

[59] Fang Wu, Dragomir Radev, and Stan Z Li. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs. In *AAAI*, volume 37, pages 5312–5320, 2023.

[60] 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*, pages 1–20, 2021.

[61] Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. Decoupled self-supervised learning for graphs. In *NeurIPS*, 2022.

[62] Liang Yang, Cheng Chen, Weixun Li, Bingxin Niu, Junhua Gu, Chuan Wang, Dongxiao He, Yuanfang Guo, and Xiaochun Cao. Self-supervised graph neural networks via diverse and interactive message passing. In *AAAI*, pages 4327–4336, 2022.

[63] Liang Yang, Runjie Shi, Qiuliang Zhang, Zhen Wang, Xiaochun Cao, Chuan Wang, et al. Self-supervised graph neural networks via low-rank decomposition. In *NeurIPS*, 2024.

[64] Xihong Yang, Jin Jiaqi, Siwei Wang, Ke Liang, Yue Liu, Yi Wen, Suyuan Liu, Sihang Zhou, Xinwang Liu, and En Zhu. Dealmvc: Dual contrastive calibration for multi-view clustering. In *ACM MM*, pages 337–346, 2023.

[65] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *AAAI*, pages 10834–10842, 2023.

[66] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *CVPR*, pages 4066–4075, 2019.

[67] Xingtong Yu, Yuan Fang, Zemin Liu, and Xinming Zhang. Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning. In *AAAI*, pages 16578–16586, 2024.

[68] Zhaoning Yu and Hongyang Gao. Molecular representation learning via heterogeneous motif graph neural networks. In *ICML*, pages 25581–25594, 2022.

[69] Rui Zhang, Arthur Zimek, and Peter Schneider-Kamp. A simple meta-path-free framework for heterogeneous network embedding. In *CIKM*, pages 2600–2609, 2022.

[70] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. Costa: Covariance-preserving feature augmentation for graph contrastive learning. In *SIGKDD*, pages 2524–2534, 2022.

[71] Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. Heterogeneous graph structure learning for graph neural networks. In *AAAI*, pages 4697–4705, 2021.

[72] Sheng Zhou, Kang Yu, Defang Chen, Bolang Li, Yan Feng, and Chun Chen. Collaborative knowledge distillation for heterogeneous information network embedding. In *WWW*, pages 1631–1639, 2022.

 https://doi.org/10.52202/079017-1366

[73] Sihang Zhou, Xinwang Liu, Jiyuan Liu, Xifeng Guo, Yawei Zhao, En Zhu, Yongping Zhai, Jianping Yin, and Wen Gao. Multi-view spectral clustering with optimal neighborhood laplacian matrix. In *AAAI*, pages 6965–6972, 2020.

[74] Yanqiao Zhu, Yichen Xu, Hejie Cui, Carl Yang, Qiang Liu, and Shu Wu. Structure-enhanced heterogeneous graph contrastive learning. In *SDM*, pages 82–90, 2022.

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

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

A Related Work

This section briefly reviews topics related to this work, including self-supervised heterogeneous graph learning in Section A.1, and spectral clustering in Section A.2.

A.1 Self-Supervised Heterogeneous Graph Learning

In recent years, self-supervised heterogeneous graph learning (SHGL) has emerged as a helpful technique to deal with the heterogeneous graph that consists of different types of entities without needing labeled data [56, 60, 57, 55, 39, 23, 67]. As a result, SHGL captures meaningful representations of nodes and edges, enabling better performance in downstream tasks like node classification and node clustering. Due to its powerful capability, SHGL has been applied to various real applications, such as social network analysis [22, 62, 10], and recommendation systems [44, 17, 15].

Existing SHGL methods can be broadly classified into two groups, *i.e.,* meta-path-based methods and adaptive-graph-based methods. In meta-path-based methods, several graphs are usually constructed based on different pre-defined meta-paths to examine diverse relationships among nodes that share similar labels [18, 74]. For example, STENCIL [74] and HDMI [18] construct meth-path-based graphs and then conduct node-level consistency constraints (*e.g.,* contrastive loss) between node representations in different graphs. In addition, HGCML [58] and CPIM [31] propose to maximize the mutual information between node representations from different meta-path-based graphs. However, pre-defined meta-paths in these methods generally require expert knowledge and prohibitive computation costs [69]. Therefore, adaptive-graph-based methods are proposed to learn the adaptive graph structures to capture the relationships among nodes that possess the same label, instead of using meta-paths. For example, recently, HERO [30] made the first attempt to learn an adaptive self-expressive matrix to capture the homophily in the heterogeneous graph, thus avoiding meta-paths.

Although existing SHGL methods (especially the adaptive-graph-based methods) have achieved impressive performance in several tasks, the learned graph structure cannot be guaranteed optimal. As a result, the learned graph structure may contain noisy connections from different classes to affect the message-passing process and weaken the discriminative information in node representations.

A.2 Spectral Clustering

Spectral clustering partitions data points into clusters based on a similarity matrix derived from the data [53, 47, 65]. Owing to its proficiency in identifying clusters with complex shapes and handling non-linearly separable data, spectral clustering is widely used in many scenarios [73, 24, 26, 64].

The spectral clustering methods can be broadly classified into two groups, *i.e.,* traditional spectral clustering and deep spectral clustering. Traditional spectral clustering methods aim to group data points that are similar to each other while being dissimilar to points in other clusters by eigendecomposition [34, 35]. For example, CAN [34] proposes to learn the data similarity matrix and clustering structure simultaneously with the eigendecomposition. SWCAN [35] further assigns weights for different features to learn the similarity graph and partition samples into clusters simultaneously. Despite its effectiveness, traditional spectral clustering generally requires expensive computation costs, especially for large datasets. To alleviate this issue, deep spectral clustering methods have been proposed in recent years. For example, DSC [66] employs an encoder and two decoders to train the network, thus obtaining discriminative representations for clustering and implementing the cluster assignment via the neural network. DSCL [21] introduces a novel metric learning framework that leverages spectral clustering principles, thus reducing complexity to linear levels. SpectralNet [42] proposes to learn a mapping function via the orthogonalization network to address the out-of-sample-extension and scalability problems.

The above methods conduct spectral clustering explicitly. Surprisingly, recent research shows that some popular self-supervised methods also implicitly conduct spectral clustering [7, 46]. For example, [7] demonstrates contrastive learning performs spectral clustering on the population augmentation graph by replacing the standard InfoNCE [36] with its proposed spectral contrastive loss. [46] demonstrates that contrastive learning with the standard InfoNCE loss is equivalent to spectral clustering on the similarity graph. Although these methods make efforts to connect previous self-supervised methods with spectral clustering, they cannot be easily transferred to SHGL. First, these methods are almost based on the augmentation graph, which assumes that different augmentations

https://doi.org/10.52202/079017-1366

of the same sample connect each other and thus form a graph. In contrast, in SHGL, there is no augmentation, and the graph is constructed by connecting different samples. Second, compared to the above methods, SHGL incorporates the message-passing process, which makes it more complex. Therefore, connecting SHGL methods with the spectral clustering remains challenging.

B Algorithm and Complexity Analysis

This section provides the pseudo-code of the proposed method in Section B.1, and the complexity analysis of our method in Section B.2.

B.1 Algorithm

Algorithm 1 The pseudo-code of the proposed method.

Input: Heterogeneous graph $\mathbf{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{T}, \mathcal{R})$, non-negative parameters β, γ, η, μ and δ;
Output: Encoders g_ϕ, f_θ;

```
Initialize parameters;
while not converge do
    Obtain semantic representations H with encoder g_φ;
    Obtain the closed-form solution of the affinity matrix S by Eq. (10);
    Obtain the orthogonal cluster assignment matrix Y by Eq. (11) and Eq. (12);
    Conduct the spectral loss based on Y and S by Eq. (13);
    Obtain node representations Z by Z = SH;
    Obtain heterogeneous representations Z̃ with encoder f_θ;
    Project node and heterogeneous representations into a latent space to obtain Q and Q̃;
    Conduct the node-level consistency constraint between Q and Q̃ by Eq. (16);
    Obtain cluster representations Q̂ by Eq. (17);
    Conduct the cluster-level consistency constraint between Q̃ and Q̂ by Eq. (18);
    Compute the objective function J by Eq. (19);
    Back-propagate J to update model weights;
end while
```

B.2 Complexity Analysis

Based on the Algorithm 1 above, we then analyze the time complexity of the proposed method. Recalling Eq. (10) in the main text:

$$\mathbf{s}_{ij} = (-\frac{1}{2\boldsymbol{\alpha}}\mathbf{d}_i + \lambda)_+, \tag{21}$$

where $\mathbf{d}_{ij} = \|\mathbf{h}_i - \mathbf{h}_j\|_2^2 + \beta \|\mathbf{f}_i - \mathbf{f}_j\|_2^2$, where $\mathbf{H} \in \mathbb{R}^{n\times d}$ and $\mathbf{F} \in \mathbb{R}^{n\times c}$ are semantic representations and eigenvector matrix, d and c indicate number of dimensions and classes, and n indicates the number of nodes. To reduce the computation costs, the proposed method proposes to only calculate $\mathbf{s}_{ij}$ between node v_i and its k nearest neighbors. Therefore, the time complexity of Eq. (21) is $\mathcal{O}(nk)$. Moreover, the proposed method proposes to replace the eigendecomposition with a projection head and orthogonalization layer to further reduce the time complexity. Specifically, the time complexity of the orthogonal process for $\mathbf{H}$ and $\mathbf{Y}$ with the QR decomposition is $\mathcal{O}(nd^2)$ and $\mathcal{O}(nc^2)$, respectively. The time complexity of the inversion process in Eq. (12) for $\mathbf{H}$ and $\mathbf{F}$ is $\mathcal{O}(d^3)$ and $\mathcal{O}(c^3)$. Moreover, the time complexity of the spectral loss is $\mathcal{O}(nkc)$ and the time complexity of $\mathbf{Z} = \mathbf{SH}$ is $\mathcal{O}(nkd)$. In addition, the time complexity of node-level and cluster-level consistency constraints are $\mathcal{O}(nd^2)$ and $\mathcal{O}(n)$, respectively. Therefore, the overall complexity of the proposed method is $\mathcal{O}(nd^2 + nc^2 + nkd + nkc + d^3 + c^3)$ in each epoch, where $d^2, c^2 < n$, thus is scaled linearly with the sample size.

C Proofs of Theorems

This section provides definition, detailed proofs of Theorems, and derivation process in Section 2, including the proofs of Theorem 2.2 in Section C.1, the proofs of Theorem 2.3 in Section C.2, the

proofs of Theorem 2.5 in Section C.3, the proofs of Theorem 2.6 in Section C.4, the derivation of Eq. (7) in Section C.5, the derivation of the closed-form solution and parameters in Section C.6, and the derivation of the orthogonalization in Section C.7.

C.1 Proof of Theorem 2.2

Theorem C.1. (Restating Theorem 2.2 in the main text). *Assume the learned representations* $\mathbf{H}$ *are orthogonal, optimizing previous meta-path-based and adaptive-graph-based SHGL methods is equivalent to performing spectral clustering with additional regularization, i.e.,*

$$\min_{\mathbf{H}} \mathcal{L}_{SHGL} \cong \min_{\mathbf{H}} \mathrm{Tr}(\mathbf{H}^T \hat{\mathbf{L}} \mathbf{H}) + R(\mathbf{H}) \ s.t., \ \mathbf{H}^T \mathbf{H} = \mathbf{I}, \tag{22}$$

where $R(\cdot)$ *indicates the regularization term,* $\hat{\mathbf{L}}$ *indicates the Laplacian matrix of the meta-path-based graph or the adaptive graph structure.*

Proof. First, we prove the connection between previous meta-path-based SHGL methods and spectral clustering. To do this, take a heterogeneous graph with two meta-paths as an example, we let $\mathcal{G} = \{\mathcal{G}^{(1)} \cup \mathcal{G}^{(2)}\}$ indicates the union of all meta-path-based graph views. Moreover, we denote the representations of previous methods before the message-passing as $\mathbf{H}$ (generally obtained by linear mapping from original node features). In addition, we denote the node representations of different graph views after the message-passing as $\mathbf{Z}^{(r)}$, respectively, where $r = 1, 2$, *i.e.,*

$$\mathbf{z}_i^{(r)} = \mathbf{h}_i + \{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(r)}\}, \tag{23}$$

where $\mathcal{N}(v_i)^{(r)}$ indicates the one-hop neighbors of node v_i in the r-th meta-path-based graph.

Based on the node representations $\mathbf{Z}^{(r)}$ of each graph, previous meta-path-based SHGL methods generally propose to extract the invariant information among node representations from different meta-path-based graphs. Here, we take the Mean Squared Error (MSE) loss as a simple example to extract the invariance and then conduct an analysis of previous meta-path-based SHGL methods. Therefore, the objective function of previous meta-path-based SHGL methods can be formulated as:

$$\min_{\theta} \sum_{i}^{n} ||\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(2)}||_2^2. \tag{24}$$

Based on Eq. (23), we can rewrite Eq. (24) as:

$$\begin{aligned}
&\min_{\theta} \sum_{i}^{n} ||\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(2)}||_2^2 \\
&= \min_{\theta} \sum_{i}^{n} ||\mathbf{h}_i + \{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}\}||_2^2 \\
&= \min_{\theta} \sum_{i}^{n} ||\mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}\} + \{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i||_2^2 \\
&= \min_{\theta} \sum_{i}^{n} ||\mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}\}||_2^2 + ||\{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i||_2^2 \\
&\quad + 2 \sum_{i}^{n} \langle (\mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}\}) \cdot (\{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i) \rangle \\
&= \min_{\theta} \sum_{i}^{n} \sum_{k}^{n} \mathcal{G}_{i,k}^{(2)} ||\mathbf{h}_i - \mathbf{h}_k||_2^2 + \sum_{i}^{n} \sum_{j}^{n} \mathcal{G}_{i,j}^{(1)} ||\mathbf{h}_i - \mathbf{h}_j||_2^2 \\
&\quad + 2 \sum_{i}^{n} \langle (\mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}\}) \cdot (\{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i) \rangle \\
&= \min_{\theta} \sum_{i}^{n} \sum_{l}^{n} \mathcal{G}_{i,l} ||\mathbf{h}_i - \mathbf{h}_l||_2^2 + 2 \sum_{i}^{n} \langle (\{\mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}) \\
&\quad \cdot (\{\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i) \rangle.
\end{aligned} \tag{25}$$

https://doi.org/10.52202/079017-1366

Denote $\mathbf{D}$ as the degree matrix of $\mathcal{G}$, denote $\mathbf{L} = \mathbf{D} - \mathcal{G}$ as the graph laplacian, and denote $\mathbf{h}^i \in \mathbb{R}^n$ $\mathbf{h}_j \in \mathbb{R}^d$ is the i-th column and j-th row of $\mathbf{H}$, according to the spectral graph analysis in [52], we further have

$$\begin{aligned}(\mathbf{h}^i)^T\mathbf{L}(\mathbf{h}^i) &= (\mathbf{h}^i)^T\mathbf{D}(\mathbf{h}^i)^T - (\mathbf{h}^i)^T\mathcal{G}(\mathbf{h}^i)^T \\ &= \sum_{j=1}^{n}\mathbf{d}_{jj}(\mathbf{h}^{ij})^2 - \sum_{j,k=1}^{n}\mathbf{h}^{ij}\mathbf{h}^{ik}\mathcal{G}_{jk} \\ &= \frac{1}{2}(\sum_{j=1}^{n}\mathbf{d}_{jj}(\mathbf{h}^{ij})^2 - 2\sum_{j,k=1}^{n}\mathbf{h}^{ij}\mathbf{h}^{ik}\mathcal{G}_{jk} + \sum_{k=1}^{n}\mathbf{d}_{kk}(\mathbf{h}^{ik})^2) \\ &= \frac{1}{2}\sum_{j,k=1}^{n}\mathcal{G}_{jk}(\mathbf{h}^{ij} - \mathbf{h}^{ik})^2.\end{aligned} \tag{26}$$

Therefore, we further have

$$\sum_{i=1}^{d}(\mathbf{h}^i)^T\mathbf{L}(\mathbf{h}^i) = \frac{1}{2}\sum_{j,k=1}^{n}\mathcal{G}_{jk}\sum_{i=1}^{d}(\mathbf{h}^{ij} - \mathbf{h}^{ik})^2. \tag{27}$$

That is

$$\text{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) = \frac{1}{2}\sum_{j,k=1}^{n}\mathcal{G}_{jk}||\mathbf{h}_j - \mathbf{h}_k||_2^2. \tag{28}$$

where $\text{Tr}(\cdot)$ indicates the matrix trace. Therefore, based on Eq. (25) and Eq. (28), we can obtain

$$\begin{aligned}&\min_{\theta}\sum_{i}^{n}||\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(2)}||_2^2 \\ &= \min_{\theta} 2\text{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) + 2\sum_{i}^{n}\langle(\mathbf{h}_i - \{\mathbf{h}_k, v_k \in \mathcal{N}(v_i)^{(2)}\}) \cdot (\mathbf{h}_j, v_j \in \mathcal{N}(v_i)^{(1)}\} - \mathbf{h}_i\})\rangle \\ &= \min_{\theta} 2\text{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) + 2\sum_{i,j,k}^{n}\mathcal{G}_{i,j}\mathcal{G}_{i,k}\langle(\mathbf{h}_i - \mathbf{h}_j) \cdot (\mathbf{h}_i - \mathbf{h}_k)\rangle.\end{aligned} \tag{29}$$

Based on the assumption that $\mathbf{H}^T\mathbf{H} = \mathbf{I}$, we can conclude that previous meta-path-based SHGL methods, which extract the invariance among different graphs, equals the known spectral clustering with additional regularization. Note that the MSE loss in the above example can be replaced by other contrastive or non-contrastive loss (*e.g.,* InfoNCE [36]), and we can easily obtain similar results.

After that, we further prove the connection between recent adaptive-graph-based SHGL methods [30] and the spectral clustering. Denote the self-expressive matrix in [30] as $\mathbf{S}$, and denote the representations after projection by linear transformation as $\mathbf{H}$. Moreover, denote $\mathbf{D_S}$ as the degree matrix of $\mathbf{S}$ and denote $\mathbf{L_S} = \mathbf{D_S} - \mathbf{S}$ as the graph Laplacian. Given that the self-expressive matrix is symmetrical and non-negative, the objective function of previous adaptive-graph-based SHGL methods can be formulated as:

$$\min_{\theta,\mathbf{S}}\|\mathbf{H} - \mathbf{S}\mathbf{H}\|_F^2 + \alpha\sum_{i,j=1}^{n}d_{ij}\mathbf{s}_{ij} + \beta\sum_{i,j=1}^{n}\mathbf{s}_{ij}^2, \tag{30}$$

where α and β are non-negative parameters, and d_{ij} indicates the distance among nodes based on $\mathbf{H}$ or original node features. Based on the self-expressive constraint in the first term of Eq. (30), we have

$$\mathbf{h}_i = \sum_{j=1,j\neq i}^{n}\mathbf{s}_{ij}\mathbf{h}_j, \forall 1 \leq i \leq n. \tag{31}$$

Therefore, for any $\mathbf{h}_i$ where $i \in [1, n]$, we further have

$$\mathbf{h}_i^T\mathbf{h}_i = \sum_{j=1,j\neq i}^{n}\mathbf{s}_{ij}\mathbf{h}_i^T\mathbf{h}_j. \tag{32}$$

Based on the constraint $\mathbf{s}_i^T\mathbf{1}=1$, we obtain

$$(\sum_{j=1}^{n}\mathbf{s}_{ij}+\sum_{j=1}^{n}\mathbf{s}_{ij})\mathbf{h}_i^T\mathbf{h}_i=2\sum_{j=1,i\neq j}\mathbf{s}_{ij}\mathbf{h}_i^T\mathbf{h}_j. \tag{33}$$

Therefore, the constraint in Eq. (31) can be transformed as:

$$(\sum_{j=1}^{n}\mathbf{s}_{ij}+\sum_{j=1}^{n}\mathbf{s}_{ij})\mathbf{h}_i^T\mathbf{h}_i-2\sum_{j=1,i\neq j}\mathbf{s}_{ij}\mathbf{h}_i^T\mathbf{h}_j=0. \tag{34}$$

In addition, we further have

$$\sum_{i=1}^{n}((\sum_{j=1}^{n}\mathbf{s}_{ij}+\sum_{j=1}^{n}\mathbf{s}_{ij})\mathbf{h}_i^T\mathbf{h}_i-2\sum_{j=1,i\neq j}\mathbf{s}_{ij}\mathbf{h}_i^T\mathbf{h}_j)=\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|\mathbf{h}_i-\mathbf{h}_j\right\|^2\mathbf{s}_{ij}. \tag{35}$$

Similar to the proof above, we can also rewrite Eq. (30) as:

$$\min_{\theta,\mathbf{S}}2\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})+\alpha\sum_{i,j=1}^{n}d_{ij}\mathbf{s}_{ij}+\beta\sum_{i,j=1}^{n}\mathbf{s}_{ij}^2. \tag{36}$$

Based on the assumption that $\mathbf{H}^T\mathbf{H}=\mathbf{I}$, we can also conclude that previous adaptive-graph-based SHGL methods are equal to the known spectral clustering with additional regularization. Therefore, we complete the proof.

□

C.2 Proof of Theorem 2.3

To prove Theorem 2.3, we first give the definition of the graph-cut algorithm as follows.

Definition C.2. *(Graph-Cut) For a given number k of subsets, the mincut approach simply consists in choosing a partition $V_1,...,V_d$ which minimizes*

$$\begin{aligned}\mathrm{Cut}\left(V_1,\ldots,V_d\right)&:=\frac{1}{2}\sum_{i=1}^{d}\mathbf{W}\left(V_i,\bar{V}_i\right),\\ \mathrm{RatioCut}\left(V_1,\ldots,V_d\right)&:=\frac{1}{2}\sum_{i=1}^{d}\frac{\mathbf{W}\left(V_i,\bar{V}_i\right)}{|V_i|}=\sum_{i=1}^{d}\frac{\mathrm{Cut}\left(V_i,\bar{V}_i\right)}{|V_i|},\end{aligned} \tag{37}$$

where $\mathbf{W}(V_a,V_b):=\sum_{i\in V_a,j\in V_b}w_{ij}$ indicates the weight between different subsets, and $\bar{V}$ is the complement of V.

Theorem C.3. (Restating Theorem 2.3 in the main text). *Under the same assumption in Theorem 2.2, optimizing previous meta-path-based and adaptive-graph-based SHGL methods is approximate to performing the* $\mathrm{RatioCut}\left(V_1,\ldots,V_d\right)$ *algorithm that divides the learned representations into d partitions $\{V_1,\ldots,V_d\}$, i.e.,*

$$\min_{\mathbf{H}}\mathcal{L}_{SHGL}\cong\min_{\mathbf{H}}\mathrm{RatioCut}\left(V_1,\ldots,V_d\right), \tag{38}$$

where d indicates the dimension of representations $\mathbf{H}$.

Proof. Given a partition of V with n samples into d sets $V_1,...,V_d$, we first define d indicator vectors $\mathbf{h}_j=(\mathbf{h}_{1,j},...,\mathbf{h}_{n,j})'$ by

$$\mathbf{h}_{i,j}=\begin{cases}1/\sqrt{|V_j|} & \text{if } v_i\in V_j\\ 0 & \text{otherwise}\end{cases}\qquad(i=1,\ldots,n;j=1,\ldots,d) \tag{39}$$

Then we set the matrix $\mathbf{H}\in\mathbb{R}^{n\times d}$ as the matrix containing those d indicator vectors as columns. Observe that the columns in $\mathbf{H}$ are orthonormal to each other, *i.e.,* $\mathbf{H}^T\mathbf{H}=\mathbf{I}$, where $\mathbf{I}$ is the identity

https://doi.org/10.52202/079017-1366

matrix. Denote $\mathbf{L}$ as the unnormalized graph Laplacian, according to [52], we can obtain

$$\begin{aligned}
\mathbf{h}_i^T \mathbf{L} \mathbf{h}_i &= \frac{1}{2} \sum_{j,k=1}^{|V_i \cup \bar{V}_i|} w_{jk} (\mathbf{h}_j - \mathbf{h}_k)^2 \\
&= \frac{1}{2} \sum_{j \in V_i, k \in \bar{V}_i} w_{jk} \left(\sqrt{\frac{|\bar{V}_i|}{|V_i|}} + \sqrt{\frac{|V_i|}{|\bar{V}_i|}} \right)^2 + \frac{1}{2} \sum_{j \in \bar{V}_i, k \in V_i} w_{jk} \left(-\sqrt{\frac{|\bar{V}_i|}{|V_i|}} - \sqrt{\frac{|V_i|}{|\bar{V}_i|}} \right)^2 \\
&= \mathrm{cut}(V_i, \bar{V}_i) \left(\frac{|\bar{V}_i|}{|V_i|} + \frac{|V_i|}{|\bar{V}_i|} + 2 \right) \\
&= \mathrm{cut}(V_i, \bar{V}_i) \left(\frac{|V_i| + |\bar{V}_i|}{|V_i|} + \frac{|V_i| + |\bar{V}_i|}{|\bar{V}_i|} \right) \\
&= |V_i| \cdot \mathrm{RatioCut}(V_i, \bar{V}_i).
\end{aligned} \tag{40}$$

Moreover, we have $\mathbf{h}_i^T \mathbf{L} \mathbf{h}_i = (\mathbf{H}^T \mathbf{L} \mathbf{H})_{ii}$. Therefore, we have

$$\mathrm{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}) = \sum_{i=1}^{d} (\mathbf{H}^T \mathbf{L} \mathbf{H})_{ii} = \sum_{i=1}^{d} \mathbf{h}_i^T \mathbf{L} \mathbf{h}_i = \mathrm{RatioCut}\left(V_1, \ldots, V_d\right), \tag{41}$$

where $\mathrm{Tr}(\cdot)$ indicates the trace of a matrix. Therefore, minimizing the $\mathrm{RatioCut}\left(V_1, \ldots, V_k\right)$ can be transferred to

$$\min_{V_1, \ldots, V_d} \mathrm{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}) \quad \text{s.t.}, \mathbf{H}^T \mathbf{H} = \mathbf{I}, \mathbf{H} \,\text{as}\, \text{ defined in Eq. (39)}. \tag{42}$$

Then we consider relaxing the constraints of the problem by allowing the entries of the matrix $\mathbf{H}$ to assume arbitrary real values. As a result, the problem is transformed into a relaxed version:

$$\min_{\mathbf{H} \in \mathbb{R}^{n \times d}} \mathrm{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}) \quad \text{s.t.}, \mathbf{H}^T \mathbf{H} = \mathbf{I}. \tag{43}$$

This is the standard spectral clustering, as we mentioned above. Therefore, we can obtain that conducting spectral clustering is approximating to conducting the RatioCut algorithm. which divide the learned representations into d partitions, where d indicates the dimension of representations. Thus, we complete the proof.

□

C.3 Proof of Theorem 2.5

Theorem C.4. (Restating Theorem 2.5 in the main text). *Optimizing the spectral loss $\mathcal{L}_{sp}$ leads to performing the spectral clustering based on the affinity matrix $\mathbf{S}$ with c connected components and conducting RatioCut $(V_1, \ldots, V_c)$ algorithm to divide the learned representations into c partitions, i.e.,*

$$\min \mathcal{L}_{sp} \Rightarrow \min \mathrm{Tr}(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}} \mathbf{Y}) \Rightarrow \min \mathrm{RatioCut}(V_1, \ldots, V_c). \tag{44}$$

Proof. According to Ky Fan's Theorem [5], the spectral loss $\mathcal{L}_{sp}$ can be written as:

$$\begin{aligned}
\mathcal{L}_{sp} &= \frac{1}{n^2} \sum_{i,j=1}^{n} \mathbf{s}_{ij} \left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2 - \gamma H(\mathbf{Y}) \\
&= \frac{2}{n^2} \mathrm{Tr}(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}} \mathbf{Y}) - \gamma H(\mathbf{Y}).
\end{aligned} \tag{45}$$

Therefore, optimizing the proposed method with $\mathcal{L}_{sp}$ is equivalent to performing the spectral clustering with additional regularization.

Note that, under the orthogonal constraint, the minimum of $\mathcal{L}_{sp}$ is attained when the column space of $\mathbf{Y}$ is the subspace of the c eigenvectors corresponding to the smallest c eigenvalues of $\mathbf{L}_{\mathbf{S}}$. In other

words, the learned $\mathbf{Y}$ can perfectly fit the eigenvectors when the minimum of $\mathcal{L}_{sp}$ is attained. Recall the objective function in the main text, *i.e.,*

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{F}} \sum_{i,j=1}^{n} (\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 \mathbf{s}_{ij} + \alpha \mathbf{s}_{ij}^2 + \beta \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \mathbf{s}_{ij}) \\ &\text{s.t.}, \forall i, \mathbf{s}_i^T \mathbf{1} = 1, 0 \leq \mathbf{s}_i \leq 1, \mathbf{F}^T \mathbf{F} = \mathbf{I}. \end{aligned} \tag{46}$$

Therefore, when the minimum of $\mathcal{L}_{sp}$ is attained, the constraints in the above function can be satisfied, *i.e.,* $\text{rank}(\mathbf{L_S}) = n - c$ holds. As a result, we can obtain the affinity matrix $\mathbf{S}$ with exactly c connected components.

Moreover, according to the Theorem C.3, we have

$$\text{Tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y}) = \sum_{i=1}^{c} (\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})_{ii} = \sum_{i=1}^{c} \mathbf{y}_i^T \mathbf{L_S} \mathbf{y}_i = \text{RatioCut}\left(V_1, \ldots, V_c\right). \tag{47}$$

That is, the proposed method divides the learned representations into c partitions, where c indicates the number of classes. Thus, we complete the proof.

□

C.4 Proof of Theorem 2.6

We first follow previous works [33] to define the Complexity Measure to evaluate the generalization ability of neural networks based on the Davies Bouldin Index.

Definition C.5. *(Complexity Measure) The complexity measure of neural networks can be defined as:*

$$C = \frac{1}{k} \sum_{i=0}^{k-1} \max_{i \neq j} \frac{S_i + S_j}{M_{i,j}}, \tag{48}$$

where

$$\begin{aligned} S_i &= \left(\frac{1}{n_i} \sum_{\tau}^{n_i} \left| O_\tau^i - \mu_i \right|^p \right)^{1/2} \quad \textit{for } i = 1 \cdots k \\ M_{i,j} &= \left\| \mu_i - \mu_j \right\|_2 \quad \textit{for } i, j = 1 \cdots k, \end{aligned} \tag{49}$$

i and j are indices of two different classes, $O_i^{(\tau)}$ is the output representation of the τ-th sample belonging to class i for the given model, μ_i is the cluster centroid of the representations of class i, S_i is a measure of scatter within representations of class i, and $M_{i,j}$ is a measure of separation between representations of classes i and j.

Moreover, we further follow previous works [33, 16] to define the generalization bound G of a model based on the model complexity, *i.e.,*

Definition C.6. *(Generalization Bound) For any $\delta \in [0, 1]$, with probability at least $1 - \delta$, the generalization bound G of a model follows the inequality,* i.e.,

$$G \leq \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i), \mathbf{y}_i) + \sqrt{\frac{C}{n}} + \mathcal{O}(\sqrt{\frac{\log(1/\delta)}{n}}), \tag{50}$$

where $(\mathbf{x}_i, \mathbf{y}_i)$ is a pair of labeled data, f is the model, l is the loss function, n is the number of labeled data, C is the model complexity measure.

Based on the Definitions above, we can derive the Theorem as follows.

Theorem C.7. (Restating Theorem 2.6 in the main text). *The proposed method with dual consistency constraints achieves a lower boundary of the model complexity C and a higher generalization ability boundary G than previous SHGL with the node-level consistency constraint only, i.e.,*

$$\inf(C_{SCHOOL}) < \inf(C_{SHGL}), \quad \sup(G_{SCHOOL}) > \sup(G_{SHGL}), \tag{51}$$

where $\inf(\cdot)$ and $\sup(\cdot)$ indicates lower bound and upper bound, respectively.

Proof. We take the binary classification as an example, Eq. (48) can be rewritten as $\frac{S_0+S_1}{M_{0,1}}$. Then, for the heterogeneous representations $\tilde{\mathbf{Z}}$ learned by the heterogeneous encoder, we can obtain its cluster

https://doi.org/10.52202/079017-1366

centroid μ_0:

$$\begin{aligned}\mu_0 = \mathbb{E}[\tilde{\mathbf{z}}_i^0] &= \mathbb{E}[\mathbf{W}(\mathbf{X}_i + \sum_{j \in \mathcal{N}(v_i)} \frac{1}{d}\mathbf{X}_j)] \\ &= \mathbf{W}\left(P_0 \cdot \mu_{\mathbf{X}_0} + (1-P_0)\cdot \mu_{\mathbf{X}_j}\right),\end{aligned} \tag{52}$$

where $\mathcal{N}(v_i)$ indicates the neighbors of v_i from other type of nodes, $\mathbf{W}$ indicates the parameters of the heterogeneous encoder, $\mu_{\mathbf{X}_0}$ indicates the cluster centroid of the node features of class i, $\mu_{\mathbf{X}_j}$ indicates the cluster centroid of the node features of other types of nodes. Similarly, we further have:

$$\mu_1 = \mathbf{W}\left(P_1 \cdot \mu_{\mathbf{X}_1} + (1-P_1)\cdot \mu_{\mathbf{X}_k}\right), \tag{53}$$

where $\mu_{\mathbf{X}_k}$ indicates the cluster centroid of the node features of other types of nodes.

Therefore, we can obtain

$$\begin{aligned}M_{0,1} &= \|\mu_0 - \mu_1\| \\ &= \left\|\mathbf{W}\left(P_0 \cdot \mu_{\mathbf{X}_0} + (1-P_0)\cdot \mu_{\mathbf{X}_j} - P_1 \cdot \mu_{\mathbf{X}_1} - (1-P_1)\cdot \mu_{\mathbf{X}_k}\right)\right\|.\end{aligned} \tag{54}$$

Moreover, we have

$$\begin{aligned}S_0^2 &= \mathbb{E}[\left\|O_i^0 - \mu_0\right\|^2] = \mathbb{E}\left[< O_\tau^0 - \mu_0, O_\tau^0 - \mu_0 >\right] \\ &= P_0^2 \mathbb{E}[\left\|\mathbf{W}\left(\mathbf{X}_i^0 - \mu_{\mathbf{X}_0}\right)\right\|^2] + (1-P_0)^2\,\mathbb{E}[\|\mathbf{W}(\mathbf{X}_i^j - \mu_{\mathbf{X}_j})\|^2],\end{aligned} \tag{55}$$

where $< \cdot, \cdot >$ is inner production. To rewrite the above function, we first derive the following inequality, *i.e.,*

$$a^2 b + (1-a)^2 c \geq \frac{bc}{b+c}, \tag{56}$$

where $0 \leq a \leq 1$, and $0 \leq b, c$. To prove the inequaility in Eq. (56), we construct function $f(a) = a^2 b + (1-a)^2 c - \frac{bc}{b+c}$. We then take the derivative of $f(a)$, i.e.,

$$f'(a) = 2ab - 2c + 2ac. \tag{57}$$

Then we let $f'(a) = 0$, and have $a = \frac{c}{b+c}$. We have

$$f(\frac{c}{b+c}) = \frac{c^2 b}{(b+c)^2} + \frac{b^2 c}{(b+c)^2} - \frac{bc}{b+c} = 0. \tag{58}$$

In addition, we take the second-order derivative of $f(a)$ and obtain

$$f''(a) = 2(b+c) \geq 0. \tag{59}$$

Therefore, $f(a)$ is decreasing when $a < \frac{c}{b+c}$ and increasing when $a > \frac{c}{b+c}$, and reaches its minimum 0 at $\frac{c}{b+c}$. As a result, $f(a) \geq 0$ always holds for $0 \leq a \leq 1$. Thus, we prove the inequality in Eq. (56).

Given the above inequality in Eq. (56), for Eq. (55), we let $\sigma_0^2 = \mathbb{E}[\|\mathbf{W}(\mathbf{X}_0^{(i)} - \mu_{\mathbf{X}_0})\|^2]$, $\sigma_1^2 = \mathbb{E}[\|\mathbf{W}(\mathbf{X}_1^{(i)} - \mu_{\mathbf{X}_1})\|^2]$, $\sigma_j^2 = \mathbb{E}[\|\mathbf{W}(\mathbf{X}_j^{(i)} - \mu_{\mathbf{X}_j})\|^2]$, and $\sigma_k^2 = \mathbb{E}[\|\mathbf{W}(\mathbf{X}_k^{(i)} - \mu_{\mathbf{X}_k})\|^2]$. Moreover, we replace a, b, and c in Eq. (56) with P_0, σ_0^2, and σ_j^2, respectively. Then Eq. (55) can be rewritten as:

$$S_0^2 = P_0^2\sigma_0^2 + (1-P_0)^2\,\sigma_j^2 \geq \frac{\sigma_0^2\sigma_j^2}{\sigma_0^2 + \sigma_j^2}. \tag{60}$$

Similarly, we can also reach the following inequality:

$$S_1^2 = P_1^2\sigma_1^2 + (1-P_1)^2\,\sigma_k^2 \geq \frac{\sigma_1^2\sigma_k^2}{\sigma_1^2 + \sigma_k^2}. \tag{61}$$

Therefore, the complexity measure C can calculated by:

$$C = \frac{\sqrt{S_0^2} + \sqrt{S_1^2}}{M_{0,1}} \geq \frac{\frac{\sigma_0\sigma_j}{\sqrt{\sigma_0^2+\sigma_j^2}} + \frac{\sigma_1\sigma_k}{\sqrt{\sigma_1^2+\sigma_k^2}}}{M_{0,1}}. \tag{62}$$

Note that the cluster-level consistency constraint minimizes the first term in the S_0^2 and S_1^2 (*i.e.,* σ_0^2 and σ_1^2). Moreover, we can observe that Eq. (60) and Eq. (61) are the increasing function with respect to σ_0 and σ_1. Therefore, minimizing the cluster-level consistency constraint is equivalent to minimizing the lower bound of the model complexity. Therefore, the lower bound of complexity measure C of the model with the dual consistent constraints is less than the model without it, *i.e.,* $\inf(C_{SCHOOL}) < \inf(C_{SHGL})$. As a result, according to [33], we can conclude that the representations learned by the dual consistent constraints have a higher bound of generalization ability than previous methods with instance-level constraint only, *i.e.,* $\sup(G_{SCHOOL}) > \sup(G_{SHGL})$ thus we complete the proof. □

C.5 Derivation of Eq. (7).

Recalling Eq. (7), *i.e.,*

$$\sum_{i=1}^{c} \tau_i \left(\mathbf{L_S}\right) = \min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L_S}\mathbf{F}) = \min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \frac{1}{2}\sum_{ij} \mathbf{s}_{ij} \left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2, \tag{63}$$

where $\mathbf{F} \in \mathbb{R}^{n\times c}$ is the eigenvector (*i.e.,* $\mathbf{F}^T\mathbf{F} = \mathbf{I}$) of $\mathbf{L_S}$ corresponding to the c eigenvalues. We first derive the first equation. The eigendecomposition of the symmetric $\mathbf{L_S}$ can be written as: $\mathbf{L_S} = \mathbf{B}\Lambda\mathbf{B}^T$, where $\mathbf{B}$ is the eigenvector matrix and Λ is the diagonal matrix whose diagonal elements are the eigenvalues of $\mathbf{L_S}$. We have:

$$\begin{aligned}
\mathrm{Tr}(\mathbf{F}^T\mathbf{L_S}\mathbf{F}) &= \mathrm{Tr}\Bigg(\begin{pmatrix} \mathbf{f}_1^T \\ \mathbf{f}_2^T \\ \vdots \\ \mathbf{f}_c^T \end{pmatrix} \mathbf{L_S} \begin{pmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \cdots & \mathbf{f}_c \end{pmatrix}\Bigg) \\
&= \mathrm{Tr}\begin{pmatrix} \mathbf{f}_1^T\mathbf{L_S}\mathbf{f}_1 & \mathbf{f}_1^T\mathbf{L_S}\mathbf{f}_2 & \cdots & \mathbf{f}_1^T\mathbf{L_S}\mathbf{f}_c \\ \mathbf{f}_2^T\mathbf{L_S}\mathbf{f}_1 & \mathbf{f}_2^T\mathbf{L_S}\mathbf{f}_2 & \cdots & \mathbf{f}_2^T\mathbf{L_S}\mathbf{f}_c \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{f}_c^T\mathbf{L_S}\mathbf{f}_1 & \mathbf{f}_c^T\mathbf{L_S}\mathbf{f}_2 & \cdots & \mathbf{f}_c^T\mathbf{L_S}\mathbf{f}_c \end{pmatrix} \\
&= \textstyle\sum_{i=1}^{c} \Lambda_i,
\end{aligned} \tag{64}$$

where $\sum_{i=1}^{c} \Lambda_i$ indicates the sum of any c eigenvalues of $\mathbf{L_S}$. Obviously, $\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L_S}\mathbf{F})$ achieves its minimization when $\mathbf{F}$ is the eigenvectors corresponding to c smallest eigenvalues. Therefore we have $\min_{\mathbf{S}} \sum_{i=1}^{c} \tau_i \left(\mathbf{L_S}\right) = \min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mathrm{Tr}(\mathbf{F}^T\mathbf{L_S}\mathbf{F})$ and the first equation in Eq. (63) is proved. Moreover, based on Eq. (26)-Eq. (28), we can further complete the proof the second equation in Eq. (63).

C.6 Derivation of the Closed-Form Solution and Parameters.

We first obtain the Lagrangian function of the objective function in Eq. (9) in the main text:

$$\mathcal{L}(\mathbf{s}_i, \lambda, \varepsilon) = \|\mathbf{s}_i + \frac{1}{2\alpha}\mathbf{d}_i\|_2^2 - \lambda(\mathbf{s}_i^T - 1) - \varepsilon_i^T\mathbf{s}_i, \tag{65}$$

where λ and $\varepsilon_i \geq \mathbf{0}$ are the Lagrangian multipliers. Based on the KKT condition [3], we can obtain the closed-form solution of the above Lagrangian function, *i.e.,*

$$\mathbf{s}_{ij} = (-\frac{1}{2\alpha_i}\mathbf{d}_{ij} + \lambda_i)_+, \tag{66}$$

where $(\cdot)_+$ indicates $\max\{\cdot, 0\}$. For the sparse affinity matrix $\mathbf{S}$, each vector $\mathbf{s}_i$ contains k nonzero elements only. Therefore, we have $\mathbf{s}_{ik} \geq 0$ and $\mathbf{s}_{i,k+1} = 0$. That is, $-\frac{1}{2\alpha_i}\mathbf{d}_{ik} + \lambda_i > 0$ and $-\frac{1}{2\alpha_i}\mathbf{d}_{i,k+1} + \lambda_i \leq 0$. Then based on Eq. (66) and the constraint $\mathbf{s}_i^T\mathbf{1} = 1$, we further have

$$\sum_{j=1}^{k}(-\frac{1}{2\alpha_i}\mathbf{d}_{ij} + \lambda_i) = 1. \tag{67}$$

https://doi.org/10.52202/079017-1366

Table 3: Statistics of all datasets.

Datasets	Type	#Nodes	#Node Types	#Edges	#Edge Types	Target Node	#Training	#Test
ACM	Heter	8,994	3	25,922	4	Paper	600	2,125
Yelp	Heter	3,913	4	72,132	6	Bussiness	300	2,014
DBLP	Heter	18,405	3	67,946	4	Author	800	2,857
Aminer	Heter	55,783	3	153,676	4	Paper	80	1,000
Photo	Homo	7,650	1	238,162	2	Photo	765	6,120
Computers	Homo	13,752	1	491,722	2	Computer	1,375	11,002

Therefore, we obtain $\lambda_i = \frac{1}{k} + \frac{1}{2k\alpha_i}\sum_{j=1}^{k}\mathbf{d}_{ij}$. Moreover, we have the following inequality for α_i, *i.e.,*

$$\frac{k}{2}\mathbf{d}_{ik} - \frac{1}{2}\sum_{j=1}^{k}\mathbf{d}_{ij} < \alpha_i \leq \frac{k}{2}\mathbf{d}_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k}\mathbf{d}_{ij}. \tag{68}$$

Hence, to achieve an optimal solution $\mathbf{s}_i$ contain precisely k non-zero values, we can set α_i as:

$$\alpha_i = \frac{k}{2}\mathbf{d}_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k}\mathbf{d}_{ij}. \tag{69}$$

Then the overall α could be set to the mean of $\alpha_1, \alpha_2, ..., \alpha_n$. Then α can be obtained by:

$$\alpha = \frac{1}{n}\sum_{i=1}^{n}(\frac{k}{2}\mathbf{d}_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k}\mathbf{d}_{ij}). \tag{70}$$

C.7 Derivation of the Orthogonalization

Recalling Eq. (12) in the main text:

$$\mathbf{Y} = \sqrt{n}\mathbf{P}\left(\mathbf{R}^{-1}\right), \tag{71}$$

where $\mathbf{P}$ is the cluster assignment matrix, and $\mathbf{R}$ is a upper triangular matrix obtained from the QR decomposition $\mathbf{P} = \mathbf{E}\mathbf{R}$ and $\mathbf{E}^T\mathbf{E} = \mathbf{I}$. Then we have

$$\mathbf{P}\left(\mathbf{R}^{-1}\right) = \mathbf{E}. \tag{72}$$

We further have

$$\begin{aligned}\mathbf{Y}^T\mathbf{Y} &= n\mathbf{E}^T\mathbf{E} \\ &= n\mathbf{I}.\end{aligned} \tag{73}$$

Therefore, we can obtain that $\mathbf{Y}$ is orthogonal.

D Experimental Settings

This section provides detailed experimental settings in Section Experiments, including the description of all datasets in Section D.1, summarization of all comparison methods in Section D.2, evaluation protocol in Section D.3, model architectures and settings in Section D.4, and computing resource details in Section D.5.

D.1 Datasets

We use four public heterogeneous graph datasets and two public homogeneous graph datasets from various domains. Heterogeneous graph datasets include three academic datasets (*i.e.,* ACM [56], DBLP [56], and Aminer [11]), and one business dataset (*i.e.,* Yelp [71]). Homogeneous graph datasets include two sale datasets (*i.e.,* Photo and Computers [43]). Table 3 summarizes the data statistics. We list the details of the datasets as follows.

- **ACM** is an academic heterogeneous graph dataset. It contains three types of nodes (paper (P), author (A), subject (S)), four types of edges (PA, AP, PS, SP), and categories of papers as labels.
- **Yelp** is a business heterogeneous graph dataset. It contains four types of nodes (business (B), user (U), service (S), level (L)), six types of edges (BU, UB, BS, SB, BL, LB), and categories of businesses as labels.
- **DBLP** is an academic heterogeneous graph dataset. It contains three types of nodes (paper (P), authors (A), conference (C)), four types of edges (PA, AP, PC, CP), and research areas of authors as labels.
- **Aminer** is an academic heterogeneous graph dataset. It contains three types of nodes (paper (P), author (A), reference (R)), four types of edges (PA, AP, PR, RP), and categories of papers as labels.
- **Photo** and **Computers** are two co-purchase homogeneous graph datasets. They are two networks extracted from Amazon's co-purchase data. Nodes are products, and edges denote that these products were often bought together. Products are categorized into several classes by the product category.

Table 4: The characteristics of all comparison methods.

Methods	Hetero	Homo	Semi-sup	Self-sup/unsup	Meta-path	Adaptive
DeepWalk (2014)		✓		✓		
GCN (2017)		✓	✓			
GAT (2018)		✓	✓			
DGI (2019)		✓		✓		
GMI (2020)		✓		✓		
MVGRL (2020)		✓		✓		
GRACE (2020)		✓		✓		
GCA (2021)		✓		✓		
G-BT (2022)		✓		✓		
COSTA (2022)		✓		✓		
DSSL (2022)		✓		✓		
LRD (2023)		✓		✓		
Mp2vec (2017)	✓			✓	✓	
HAN (2019)	✓		✓		✓	
HGT (2020)	✓		✓			
DMGI (2020)	✓			✓	✓	
DMGIattn (2020)	✓			✓	✓	
HDMI (2021)	✓			✓	✓	
HeCo (2021)	✓			✓	✓	
HGCML (2023)	✓			✓	✓	
CPIM (2023)	✓			✓	✓	
HGMAE (2023)	✓			✓	✓	
HERO (2024)	✓			✓		✓
SCHOOL (ours)	✓			✓		✓

D.2 Comparison Methods

The comparison methods include eleven heterogeneous graph methods and twelve homogeneous graph methods. Heterogeneous graph methods include Mp2vec [4], HAN [56], HGT [13], DMGI [38], DMGIattn [38], HDMI [18], HeCo [57], HGCML [58], CPIM [31], HGMAE [48], and HERO [30]. Homogeneous graph methods include GCN [20], GAT [50], DeepWalk [41], DGI [51], GMI [40], MVGRL [8], GRACE [75], GCA [76], G-BT [2], COSTA [70], DSSL [61], and LRD [63]. The characteristics of all methods are listed in Table 4, where "Hetero" and "Homo" indicate the methods designed for the heterogeneous graph and homogeneous graph, respectively. "Semi-sup", and "Self-sup/unsup" indicate that the method conducts semi-supervised learning, and self-supervised/unsupervised learning, respectively. "Meta-path" indicates that the method requires pre-defined meta-paths during the training process. "Adaptive" indicates that the method learns an adaptive graph structure instead of traditional meta-paths.

https://doi.org/10.52202/079017-1366

Table 5: Settings for the dimensions of encoders (i.e., $g_\phi \in \mathbb{R}^{f \times d_1}$ and $f_\theta \in \mathbb{R}^{f \times d_1}$) and projection heads (i.e., $p_\varphi \in \mathbb{R}^{d_1 \times c}$ and $q_\gamma \in \mathbb{R}^{d_1 \times d_2}$) on all datasets.

Settings	ACM	Yelp	DBLP	Aminer	Photo	Computers
f	1,902	82	334	128	745	767
d_1	512	256	128	256	1024	1024
d_2	64	256	256	256	256	256
c	3	3	4	4	8	10

Table 6: Clustering performance (*i.e.,* NMI and ARI) of all methods on heterogeneous graph datasets.

Method	**ACM**		**Yelp**		**DBLP**		**Aminer**	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
DeepWalk	41.6±0.5	35.3±0.6	35.1±0.8	37.7±1.1	69.0±0.2	73.3±0.3	26.2±0.3	22.4±0.4
Mp2vec	21.4±0.7	21.1±0.5	38.9±0.6	39.5±0.5	73.5±0.4	77.7±0.6	30.4±0.4	25.5±0.6
DMGI	67.8±0.9	70.2±1.0	36.8±0.6	34.4±0.7	72.2±0.8	72.8±0.9	27.3±0.9	23.1±0.8
DMGIattn	**70.2±0.3**	72.5±0.6	38.1±0.8	40.2±0.6	69.6±0.6	73.9±0.4	28.3±0.3	25.5±0.5
HDMI	69.5±0.5	72.3±0.7	38.9±0.6	40.7±0.8	73.1±0.3	74.4±0.4	33.5±0.4	28.9±0.5
HeCo	67.8±0.8	70.5±0.7	39.3±0.6	42.1±0.8	74.5±0.8	80.1±0.9	32.2±1.1	28.6±1.0
HGCML	69.1±0.7	71.6±0.8	37.4±0.6	39.5±0.8	74.5±0.9	75.1±1.1	35.9±0.6	31.1±0.5
CPIM	68.6±0.3	70.8±0.5	40.1±0.8	42.1±0.9	73.7±0.5	78.0±0.3	35.8±0.5	30.1±0.7
HGMAE	69.7±0.8	72.6±0.6	40.3±0.9	42.4±0.8	76.9±0.6	82.3±0.7	41.1±0.8	38.3±0.9
HERO	68.8±0.6	71.8±0.6	38.6±0.8	40.6±0.9	74.1±0.7	79.3±0.7	36.8±0.7	35.3±0.9
SCHOOL	69.6±0.7	**72.7±0.5**	**41.2±0.9**	**43.5±0.6**	**77.1±0.6**	**82.5±0.5**	**42.4±0.6**	**38.8±0.8**

D.3 Evaluation Protocol

We follow the evaluation in previous works [18, 37, 72] to conduct node classification and node clustering as semi-supervised and unsupervised downstream tasks, respectively. Specifically, we first train models with unlabeled data in a self-supervised manner and output learned node representations. After that, the resulting representations can be used for different downstream tasks. For the node classification task, we train a simple logistic regression classifier with a fixed iteration number, and then evaluate the effectiveness of all methods with Micro-F1 and Macro-F1 scores. For the node clustering task, we conduct clustering and split the learned representations into c clusters with the K-means algorithm, then calculate the normalized mutual information (NMI) and average rand index (ARI) to evaluate the performance of node clustering.

D.4 Model Architectures and Settings

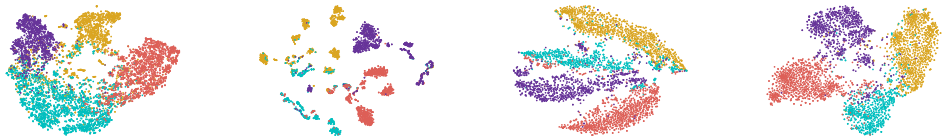
As described in Section 2, the proposed method employs the MLP (*i.e.,* g_ϕ) and the closed-form solution of the affinity matrix $\mathbf{S}$ to obtain node representations $\mathbf{Z}$. Moreover, the proposed method employs the heterogeneous encoder (*i.e.,* f_θ) to obtain heterogeneous representations $\widetilde{\mathbf{Z}}$. In addition, the proposed method employs the projection head p_φ to obtain the cluster assignment matrix $\mathbf{P}$. After that, the proposed method employs projection head q_γ to map the node representations and heterogeneous representations into latent spaces. In the proposed method, projection head p_φ and q_γ are simply implemented by the linear layer, followed by the ReLU activation. We report the settings for the dimensions of encoders in Table 5. Finally, In the proposed method, all parameters were optimized by the Adam optimizer [19] with an initial learning rate. Moreover, We use early stopping with a patience of 30 to train the proposed SHGL model. In all experiments, we repeat the experiments five times for all methods and report the average results.

D.5 Computing Resource Details

All experiments were implemented in PyTorch and conducted on a server with 8 NVIDIA GeForce 3090 (24GB memory each). Almost every experiment can be done on an individual 3090, and the training time of all comparison methods as well as our method, is less than 1 hour.

Table 7: Classification performance (*i.e.,* Macro-F1 and Micro-F1) on homogeneous graph datasets.

Method	Photo		Computers	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
DeepWalk	87.4±0.5	89.7±0.3	84.0±0.3	85.6±0.4
GCN	90.5±0.3	92.5±0.2	84.0±0.4	86.4±0.3
GAT	90.2±0.5	91.8±0.4	83.2±0.2	85.7±0.4
DGI	89.3±0.2	91.6±0.3	79.3±0.3	83.9±0.5
GMI	89.3±0.4	90.6±0.2	80.1±0.4	82.2±0.4
MVGRL	90.1±0.3	91.7±0.4	84.6±0.6	86.9±0.5
GRACE	90.3±0.5	91.9±0.3	84.2±0.3	86.8±0.5
GCA	91.1±0.4	92.4±0.4	85.9±0.5	87.7±0.3
COSTA	91.3±0.4	92.5±0.3	86.4±0.3	88.3±0.4
DSSL	90.6±0.2	92.1±0.3	85.6±0.3	87.3±0.4
LRD	91.1±0.5	92.8±0.7	**86.6±0.3**	88.6±0.6
SCHOOL	**91.9±0.4**	**93.1±0.3**	85.9±0.6	**88.7±0.5**



(a) HDMI (SIL: 0.36) (b) HeCo (SIL: 0.33) (c) HERO (SIL: 0.34) (d) SCHOOL (SIL: 0.41)

Figure 3: Visualization plotted by t-SNE and the corresponding silhouette scores (SIL) of node representations of the proposed SCHOOL and other SHGL comparison methods on the DBLP dataset.

E Additional Experiments

This section provides some additional experimental results to support the proposed method, including experiments on the effectiveness of the affinity matrix in Section E.1, visualization of the learned representations in Section E.4, parameter analysis in Section E.5, experimental results on the node clustering task in Table 6, and experimental results on homogeneous graph datasets in Table 7.

E.1 Effectiveness of the Rank-Constrained Affinity Matrix

The proposed method proposes to learn a rank-constrained affinity matrix with exact c components to capture the connections within the same class while mitigating the connections from different classes. This actually shares part of a similar idea with the self-attention mechanism, which aims to assign weights for all sample pairs. To further verify the effectiveness of the rank-constrained affinity matrix, we investigate the performance of the variants methods with the cosine similarity, the affinity matrix, the self-attention mechanism, and report the results in Table 8.

Obviously, the proposed method with the affinity matrix obtains superior performance than the cosine similarity and the self-attention mechanism on all datasets. The reason can be attributed to the fact that the affinity matrix in the proposed method is constrained to contain exactly c components to mitigate noisy connections from different classes. In contrast, although either the cosine similarity or self-attention mechanisms may assign small weights for node pairs from different classes, it inevitably introduces noise during the message-passing process to affect the quality of node representations. As a result, the effectiveness of the rank-constrained affinity matrix is verified.

E.2 Effectiveness of the Node-level Consistency Constraint

To verify the effectiveness of the node-level consistency constraint, we conducted experiments to replace the proposed node-level consistency constraint with the InfoNCE loss and reported the

https://doi.org/10.52202/079017-1366

Table 8: Classification performance (*i.e.,* Macro-F1 and Micro-F1) of variant methods with the affinity matrix, cosine similarity and, self-attention mechanisms on heterogeneous graph datasets.

Method	ACM		Yelp		DBLP		Aminer	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
cosine similarity	85.3±0.9	85.1±1.1	88.2±0.4	87.7±0.7	89.3±0.7	90.5±0.8	67.6±0.5	75.4±0.6
self-attention	88.7±0.8	88.4±0.7	92.0±0.5	91.7±0.6	91.2±0.4	92.1±0.6	73.2±0.7	82.1±0.6
affinity matrix	**92.7±0.6**	**92.6±0.5**	**93.0±0.7**	**92.8±0.4**	**94.0±0.3**	**94.7±0.4**	**77.5±0.9**	**86.8±0.7**

Table 9: Classification performance (*i.e.,* Macro-F1 and Micro-F1) of the node-level consistency constraint and InfoNCE loss on heterogeneous graph datasets.

Method	ACM		Yelp		DBLP		Aminer	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
InfoNCE	91.3±1.1	91.2±0.8	92.4±0.7	92.0±0.8	94.1±0.6	94.6±0.5	76.8±0.4	85.4±0.3
Proposed	**92.7±0.6**	**92.6±0.5**	**93.0±0.7**	**92.8±0.4**	**94.0±0.3**	**94.7±0.4**	**77.5±0.9**	**86.8±0.7**

results in Table 9. From Table 9, we can find that the variant method with InfoNCE loss obtains a similar performance to the proposed method. However, the InfoNCE loss generally requires the time complexity of $\mathcal{O}(n^2)$, where n is the number of nodes. This may introduce large computation costs during the training process. In contrast, the proposed method simply designs the node-level consistency constraint in Eq. (15) to capture the invariant information with the time complexity of $\mathcal{O}(nd^2)$, where d is the representation dimension and generally $d^2 < n$.

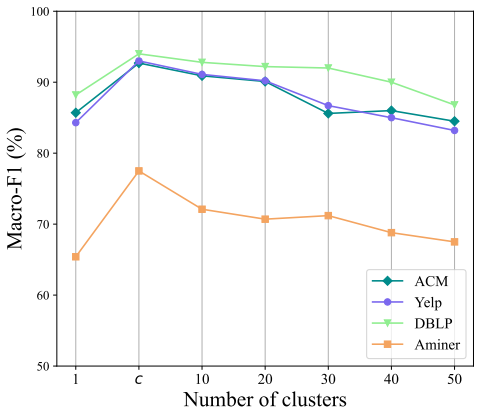


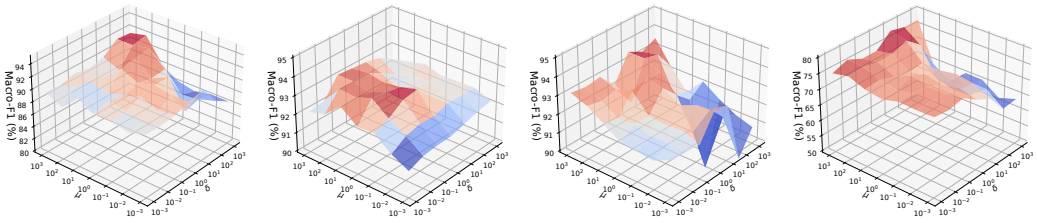
Figure 4: Classification performance (i.e., Macro-F1) of the proposed method under different clusters.

E.3 Effectiveness of Different Cluster Numbers

The proposed method divides the learned representations into several clusters. Generally, the number of clusters equals to c obtains better results because, in downstream tasks, it is easier to distinguish c clusters than a larger number of clusters. To verify it, we changed the number of clusters and reported the results in Figure 4. Obviously, the proposed method obtains the best results when the number of clusters equals to c and decreases as the number of classes increases. This is reasonable because when the number of classes increases, the nodes within the same class may be assigned to different clusters, thus making it difficult to classify them correctly.

E.4 Visualization of the Learned Representations

To further verify the effectiveness of the learned representations, we visualize node representations of the proposed SCHOOL and other SHGL comparison methods on the DBLP dataset and report the results and corresponding silhouette scores (SIL) in Figure 3. Obviously, in the visualization,



(a) ACM (b) Yelp (c) DBLP (d) Aminer

Figure 5: The classification performance of the proposed method at different parameter settings (*i.e.,* μ, and δ) on all heterogeneous graph datasets.

the node representations learned by the proposed method exhibit better clustering status, *i.e.,* nodes with different class labels are more widely separated. Moreover, the representations learned by the proposed method obtain the best silhouette score, compared to other SHGL comparison methods (*i.e.,* HDMI, HeCo, and HERO). The reason can be attributed to the fact that the proposed method conducts spectral clustering explicitly, and cuts the learned graph into c components as well as further utilizes the clustering information to facilitate downstream tasks.

E.5 Parameter Analysis

In the proposed method, we employ non-negative parameters (*i.e.,* μ, and δ) to achieve a trade-off between different terms of the final objective function $\mathcal{J}$. To investigate the impact of μ, and δ with different settings, we conduct the node classification on all heterogeneous graph datasets by varying the value of parameters in the range of $[10^{-3}, 10^{3}]$ and reporting the results in Figure 5. From Figure 5, we can find that if the values of parameters are too small (*e.g.,* 10^{-3}), the proposed method cannot achieve satisfactory performance. This verifies that both node-level and cluster-level consistency constraints are significant for the proposed method.

https://doi.org/10.52202/079017-1366

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]

2. **Limitations**

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

   Answer: [Yes] We discuss the limitations of the work in Section 4.

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] We provide the assumptions and complete proof in Appendix C.

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] See Section 3 and Appendix D.

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] We released codes and data at https://github.com/YujieMo/SCHOOL.

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] We specify all the training and test details in Appendix D.

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]

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] We list the details of experiments compute resources in Appendix D.5.

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]

10. **Broader Impacts**

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

    Answer: [Yes] We discuss broder impacts in Section 4.

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]

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]

13. **New Assets**

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

    Answer: [Yes]

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]

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]

https://doi.org/10.52202/079017-1366