Supplementary Material - Trustworthy Neighborhoods Mining: Homophily-Aware Neutral Contrastive Learning for Graph Clustering

1. Introduction

This supplementary material aims to offer additional technical details and supporting evidence to facilitate a deeper understanding of our proposed NeuCGC. Specifically, this supplementary document elaborates on the notations adopted in the main paper, provides comprehensive information on the datasets and baseline methods used in the experiments, and offers further insights into the design of the model architecture and the proposed neutral contrastive learning mechanism. These elaborations contribute to a clearer understanding of the proposed approach.

2. Notations

Here, to ensure clarity and convenience of reading, we define the key mathematical notations used in the NeuCGC framework. These symbols represent the data, variables, and operations central to our methodology. Table I summarizes the notations, along with their descriptions and their dimensions.

TABLE I
Summary of notation in the main paper

Notation	Description
$\mathcal{G}$	Attributed graph.
$\mathcal{V}$	Node set.
$\mathcal{E}$	Edge set.
N	Number of nodes, $N = \lvert\mathcal{V}\rvert$.
v_i	The i'th node. $v_i \in \mathcal{V}$.
(i, j)	The edge connecting v_i and v_j. $(i, j) \in \mathcal{E}$.
$\mathbf{X} \in \mathbb{R}^{N\times D}$	Node attribute matrix. The original features.
$\mathbf{x}_i \in \mathbb{R}^{D}$	Node attribute vector of v_i.
D	Dimensionality of the original features $\mathbf{X}$.
d	Dimension of the latent embeddings.
y_i	Ground truth label of v_i.
$\mathbf{A} \in \mathbb{R}^{N\times N}$	Adjacency matrix of graph $\mathcal{G}$.
$\mathbf{Z} \in \mathbb{R}^{N\times d}$	Latent representations.
$\mathbf{z}_i \in \mathbb{R}^{d}$	Latent representation of v_i.
$f_{\mathbf{\Theta}}$	Neural networks parameterized by $\mathbf{\Theta}$.
$\mathcal{F}_{\mathbf{\Theta}^{(l)}}$	The l'th encoder of the pseudo-Siamese networks.
r_h	Homophily ratio.
r_{nh}	Neighborhood homophily ratio.
$\mathcal{N}_i$	Neighborhood (neighbor set) of v_i.
δ	Graph neighborhood congener ratio.
$p_{\mathbf{\Theta}^{(l)}}(\mathbf{Z}^{(l)} \mid \mathbf{X})$	Global probability distribution of the l'th view.
$p_{\mathbf{\Theta}^{(l)}}(\mathbf{z}_i^{(l)} \mid \mathbf{x}_i)$	Probability distribution of v_i in the l'th view.
$D_{KL}(P \Vert Q)$	Kullback-Leibler (KL) divergence from probability distribution Q to P.
$D_{SKL}(P, Q)$	Symmetric Kullback-Leibler (SKL) divergence between probability distributions Q and P.
$\mathcal{L}_{GDA}$	Global feature distribution alignment (GDA) loss.
$\mathcal{L}_{NCA}$	Neighborhood distribution neutral contrastive alignment (NCA) loss.
$\mathcal{L}_{AFC}$	Adaptive feature consistency neutral contrastive (AFC) loss.
$\mathbf{K} \in \mathbb{R}^{N\times N}$	Cross-view SKL divergence matrix.
η	Neutral contrastive factor.
$\mathbf{S} \in \mathbb{R}^{N\times N}$	Cross-view cosine similarity matrix.
$\mathbf{S}^{\mathcal{N}} \in \mathbb{R}^{N\times N}$	Cross-view similarity matrix of original neighbors.
$\mathrm{norm}(\cdot)$	Min-max normalization.
$\mathbb{1}(\cdot)$	Indicator function.
$\mathrm{tr}(\cdot)$	Trace operator.
ξ	Neighbor threshold.
k	High-confidence signal.
$\mathbf{c} \in \mathbb{R}^{N}$	K-means clustering labels.
$\mathbf{c}^h \in \mathbb{R}^{k\cdot N}$	High-confidence pseudo labels.
$\mathbf{H} \in \mathbb{R}^{N\times N}$	High-confidence graph.
λ_1, λ_2	Hyper-parameters for balancing the losses.
$\mathcal{O}(\cdot)$	Computational complexity.

3. Dataset Descriptions

In this section, we provide a detailed description of the datasets used in our experiment.

1) **Cora, Citeseer, and Pubmed** [1]: They are citation networks among the most widely used node classification benchmarks. Each dataset is a citation and high-homophily graph, where nodes are documents, and edges are citation relationships from one to another. The class label of each node is based on the research field. A bag of words of its abstracts is used as the features of nodes.
2) **ACM** [2]: A paper network from the ACM dataset. There is an edge between two papers if they are written by same author. Paper features are the bag-of-words of the keywords. We select papers published in KDD, SIGMOD, SIGCOMM, MobiCOMM and divide the papers into three classes (database, wireless communication, data mining) by their research area.
3) **DBLP** [2]: An author network from the DBLP dataset. There is an edge between two authors if they are the coauthor relationship. The authors are divided into four areas: database, data mining, machine learning and information retrieval. We label each author's research area according to the conferences they submitted. Author features are the elements of a bag-of-words represented of keywords.
4) **Photo** [3]: A co-purchase graph from Amazon, where nodes represent products and frequently bought products are connected by edges. Node features represent product reviews, and class labels indicate the product category.
5) **Texas, Wisconsin, and Cornell** [4]: They are webpage networks collected from computer science departments of different universities by Carnegie Mellon University. For each network, nodes are web pages and edges

indicate hyperlinks between web pages. Node features are bag-of-words representations of web pages. The task is to classify nodes into five categories: student, project, course, staff, and faculty.

6) **Chameleon and Crocodile** [5]: They are Wikipedia networks, where nodes represent web pages and edges represent hyperlinks between them. Features of nodes are several informative nouns on Wikipedia pages. Labels of nodes are based on the average daily traffic of the web page.

4. Baselines

In this section, we present a detailed description of the baselines used in our comparison experiment.

1) **SDCN** [6]: A deep graph clustering method that combines the strengths of both autoencoder and GCN with a delivery operator and a dual self-supervised module. This make it the first DGC method that applies structural information into deep clustering explicitly.
2) **DFCN** [7]: A deep fusion graph clustering method that adopts a structure and attribute information fusion module for better information interaction between AE and GAE. Additionally, it develops a symmetric graph autoencoder to further improve the generalization capability.
3) **AutoSSL** [8]: A graph self-supervised learning method that propose two strategies to efficiently search SSL tasks based on pseudo-homophily. One employs evolution algorithm and the other performs differentiable search via meta-gradient descent. This enable it to adjust the task weights during search.
4) **GraphMAE** [9]: A masked graph autoencoders method for generative self-supervised graph representation learning. By adopting critical designs, namely, masked feature reconstruction, scaled cosine error, and re-mask decoding, it unleashes the power of autoencoders for graph learning.
5) **DyFSS** [10]: A a multi-task self-supervised graph clustering framework that dynamically fuses the features extracted from multiple SSL tasks for each node using distinct weights derived from a gating network.
6) **DGCluster** [11]: A deep graph clustering framework that uses pairwise (soft) memberships between nodes to solve the graph clustering problem via modularity maximization.
7) **DCRN** [12]: A siamese network-based deep graph clustering method to solve the problem of representation collapse. It adopts a dual correlation reduction strategy to improve the discriminative capability of the sample representation, and thus is free from the complicated negative sample generation operation.
8) **AGC-DRR** [13]: An attributed graph clustering method that adaptively learns the adjacent matrix with an adversarial learning mechanism. It adopts a dual redundancy reduction strategy that decreases the information redundancy in both the input space and latent feature space to improve clustering performance.
9) **CONVERT** [14]: A traditional contrastive graph clustering method with reliable augmentation by designing a reversible perturb-recover network to generate the augmented view with reliable semantics.
10) **SCGC** [15]: A simple and neighbor-based contrastive graph clustering method. It adopts a data augmentation technique to conduct data perturbation in the enhanced attribute space, and a neighbor-oriented contrastive loss to keep the structural consistency across views.
11) **NCLA** [16]: A homophily assumption-based neighbor contrastive learning method for self-supervised graph learning. It adopts a multi-head graph attention mechanism as the learnable graph augmentation to avoid improper modification of the original topology.
12) **SCAGC** [17]: A contrastive graph clustering method that treats the representations of intra-cluster nodes as positive pairs and the representations of inter-cluster nodes as negative pairs with the prompt of pseudo-label for node representation learning.
13) **CCGC** [18]: A cluster-guided contrastive deep graph clustering network to improve the quality of positive and negative samples by mining the high-confidence clustering information. It adopts special un-shared parameters Siamese encoders to avoid semantic drift caused by inappropriate graph data augmentations.
14) **HSAN** [19]: A contrastive deep graph clustering method that focuses on both hard positive and negative sample pairs. It designs a comprehensive similarity measure criterion by considering both attribute and structure information to assist the hard sample mining.
15) **GraphACL** [20]: A graph contrastive learning approach which captures one-hop neighborhood context and two-hop monophily similarities in a simple asymmetric learning framework.
16) **HeterGCL** [21]: A self-supervised graph contrastive learning method that improves the "augmentation-encoding-contrast" pattern by incorporating structure and semantic learning to obtain effective node representations for different homophilic-level graphs.

5. Additional Discussions

In order to enhance the comprehensibility of our method, we present more in-depth analysis concerning the architectural design of the pseudo-Siamese encoders and the influence of the neutral contrastive factor η in this section.

5.1 Why Do We Choose the MLP-based Pseudo-Siamese Encoders?

As stated in Section IV-A of the main paper, we adopt pseudo-Siamese encoders built with dual, parameter-unshared multilayer perceptrons (MLPs) for feature extraction. This design is motivated by the observation that traditional Graph Neural Networks (GNNs) primarily rely on neighborhood aggregation, which has been shown to degrade performance on heterophilic graphs due to their label-inconsistent neighborhoods [22], [23], [24]. MLPs, by contrast, can avoid topology-induced biases by focusing solely on node attributes.

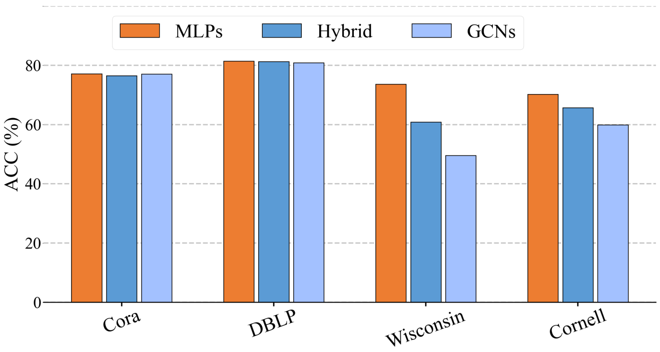


Fig. 1. Performance comparison with respect to clustering accuracy of different network structures (MLPs, Hybrid, and GCNs) adapted in the pseudo-Siamese encoders on four homophilic and heterophilic graph datasets.

Due to space limitations in the main paper, we conduct comparison experiment in this section. Specifically, as shown in Fig. 1, we compare the following encoder variants:

1) MLPs (ours): Both encoders are unshared single-layer MLP.
2) Hybrid: One encoder is a single-layer MLP, and the other is a single-layer GCN.
3) GCNs: Both encoders are unshared single-layer graph convolutional network (GCN).

The experiment is conducted on two homophilic datasets (Cora and DBLP) and two heterophilic datasets (Wisconsin and Cornell). All settings remain consistent across methods. After obtaining the dual embeddings, we concatenate them and apply K-means clustering to evaluate performance using clustering accuracy (ACC). The results in Fig. 1 show:

1) On homophilic graphs, all three encoders achieve comparable performance. While on heterophilic graphs, our MLP-based encoders significantly outperform both GCN-based and hybrid variants.
2) The GCN-based encoder yields the lowest performance on heterophilic datasets, likely due to the interference from structurally misleading neighbors.
3) The hybrid encoder offers intermediate performance, slightly alleviating the GCN's limitations through partial reliance on attribute-based learning from MLPs.

These findings align with observations from prior studies that highlight the structural bias of GNNs in heterophilic settings [22], [23], [24]. Overall, the results confirm that MLPs offer more consistent and robust performance across graphs with varying levels of homophily. Therefore, this comparison empirically supports our choice of using MLP-based pseudo-Siamese encoders to ensure a topology-agnostic representation learning mechanism that generalizes well to both homophilic and heterophilic graph datasets.

5.2 *Additional Analysis of the Neutral Contrastive Factor*

5.2.1 *Impact of the Neutral Contrastive Factor η on Clustering Performance*

To investigate the role of η in our neutral contrastive learning framework, we conduct a case study by manually setting η as a hyper-parameter, varying it from 0 to 1 in increments of 0.1. We evaluate the clustering accuracy (ACC) on a homophilic graph dataset, Cora ($r_h = 0.81$), and a heterophilic graph dataset, Wisconsin ($r_h = 0.18$). The results are presented in Fig. 2 (left). From the figure, we have some key observations include:

1) On Cora (homophilic): Higher η values (e.g., [0.6, 1.0]) improve clustering accuracy, as homophilic graphs have more same-class neighbors with shared semantic information. A larger η assigns greater weight to neutral pairs, enhancing their contribution to contrastive learning and enabling the model to learn more discriminative representations. Conversely, lower η values (e.g., [0.0, 0.3]) reduce accuracy due to insufficient utilization of these neighbors.
2) On Wisconsin (heterophilic): Lower η values (e.g., [0.0, 0.2]) yield better performance, as heterophilic graphs contain more neighbors from different classes. A smaller η minimizes their interference in contrastive learning, preserving feature quality. Higher values degrade performance by amplifying the impact of dissimilar neighbors.

These findings confirm that η significantly affects model performance and that optimal η values vary with graph homophily, validating the necessity of our neutral contrastive learning mechanism. Notably, as described in Equation (10) of the main text, we compute η adaptively to avoid tedious hyper-parameter tuning, which is further analyzed below.

5.2.2 *Impact of the Threshold ξ on Clustering Performance*

In Equation (10) of the main paper, we use the threshold ξ to filter reliable neighbors by comparing their similarity to ξ, with η calculated as the proportion of reliable neighbors in the neighborhood. To study ξ's impact, we treat it as a hyper-parameter, vary it from 0 to 1 in increments of 0.1, and evaluate clustering accuracy on Cora and Wisconsin. The results are shown in Fig, 2 (right), from which we make some key observations as followed:

1) A larger ξ filters out more neighbors, resulting in a smaller η. This benefits heterophilic graphs like Wisconsin, where fewer neighbors share the same class, but harms homophilic graphs like Cora, where more neighbors are semantically consistent.

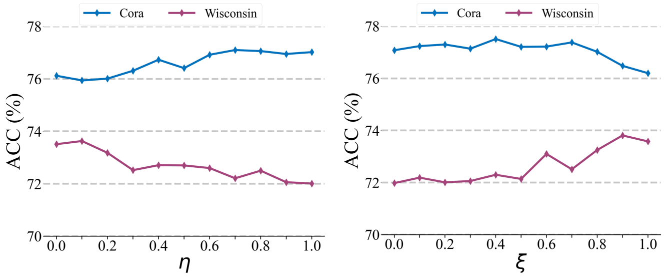
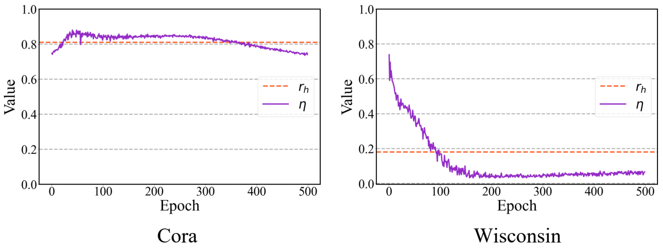


Fig. 2. Sensitive analysis of the neutral contrastive factor η (left) and the threshold ξ (right) on homophilic graph dataset Cora and heterophilic graph dataset Wisconsin. Here, we manually set η and ξ as two hyper-parameters, varying them from 0 to 1 in increments of 0.1. In practice, they both are adaptively calculated in our proposed method.



Cora

Wisconsin

Fig. 3. The learning curves of the adaptive neutral contrastive factor η during training and the graph homophily ratio r_h on homophilic graph dataset Cora (left) and heterophilic graph dataset Wisconsin (right).

2) A smaller ξ includes more neighbors as reliable, yielding a larger η. This enhances performance on Cora by leveraging same-class neighbors but degrades performance on Wisconsin due to interference from dissimilar neighbors.

These results demonstrate that ξ effectively modulates η to adapt the contribution of neutral pairs in contrastive learning for graphs of varying homophily. As described in Equation (11), we compute ξ adaptively using the average cross-view similarity of the same node, eliminating manual tuning and tailoring ξ and η to each graph's characteristics. The effectiveness of this adaptive approach is analyzed next.

5.2.3 *Variation of Adaptively Computed η Across Homophilic and Heterophilic Graphs*

Our ablation study in Section V-E of the main paper demonstrates the effectiveness of η. To further illustrate how our adaptively computed η functions in the neutral contrastive learning mechanism, we visualize its variation during training in Fig. 3. The results in this figure reveal:

1) On Cora (homophilic): The learned η stabilizes near the graph's homophily ratio ($r_h = 0.81$). A larger η increases the contribution of neutral pairs, enabling the model to capture richer semantic consistency and produce more discriminative clustering assignments.
2) On Wisconsin (heterophilic): The learned η is closer to 0, reducing the contribution of neutral pairs to avoid interference from dissimilar neighbors, thus maintaining high-quality clustering results.

These observations validate the effectiveness of our adaptive computation of η and ξ, explaining why our neutral contrastive learning mechanism achieves superior performance across graphs with different homophily levels.

5.2.4 *Role of Coarse- vs. Fine-Grained Neutral Contrastive Factors*

In this work, we introduce the novel concept of neutral pair, with their definition elaborated in Section III-C of the main manuscript. Neutral pairs are formed with each node and its neighbors, weighted by the neutral contrastive factor. The weighting can be applied in two ways:

1) Coarse-Grained manner: A single weight coefficient is applied to all neutral pairs, resulting in uniform contributions in contrastive learning. In our Neutral Contrastive Distribution Alignment module, we use a single coefficient η estimated based on graph homophily, which qualifies as a coarse-grained neutral contrastive factor to uniformly weight all original neighbors in $\mathbf{A}$.
2) Fine-Grained manner: Each neutral pair is assigned a unique weight, allowing varied contributions in contrastive learning. In the Adaptive Feature Consistency Neutral Contrastive Learning module, we use edge weights from the learned high-confidence graph $\mathbf{H}$ to weight each neutral pair, constituting fine-grained neutral contrastive factors.

These coarse- and fine-grained approaches work synergistically within our neutral contrastive learning framework, enabling the model to learn representative and discriminative consistent features, leading to remarkable clustering performance across graphs with varying homophily levels.

5.3 *Additional Analysis with LLM-Empowered Approaches*

5.3.1 *Overview of LLM-empowered Graph Learning*

Large language models (LLMs) [25], [26] are a series of deep neural networks pretrained on massive text corpora to capture the statistical and semantic structures of natural language [27], [28]. Their emergence represents a paradigm shift from task-specific deep learning models to general-purpose language understanding systems. Recent study has demonstrated that LLMs exhibit preliminary graph reasoning capabilities, achieving remarkable performance in simple graph reasoning tasks such as connectivity, cycle, and shortest path, outperforming random baselines by 37.33%–57.82% [29]. According to a recent survey [30], LLMs can enhance graph learning by enriching semantic content, aligning cross-domain knowledge, and generating structural hypotheses. These abilities make LLMs well-suited for semantic reasoning, cross-domain or cross-modal alignment, and generative augmentation in graph-based applications, leading to a surge of research in LLM-for-graph learning.

For instance, LLM4NG [31] and LLMGNN [32] leverage the strong contextual understanding of LLMs to infer latent or missing information from incomplete inputs such as nodes or labels. Given their extensive knowledge base, methods such as TAPE [33], MAGB [34], and UniGraph2 [35] exploit LLMs for cross-domain and cross-modal alignment (e.g., textual attribute alignment or multimodal alignment). Moreover, owing to their generative ability, some studies (e.g., LLM4NG [31], LLMGNN [32], GraphEdit [36], TAGrader [37]) employ textual or structural prompts to synthesize new graph data—such as addressing class imbalance or structural imbalance—for improved data robustness.

Although numerous LLM-empowered graph learning methods have been proposed [32], [38], [39], most of them focus on supervised node classification or text classification tasks. In contrast, leveraging LLMs for unsupervised graph node clustering remains relatively underexplored. Among the few existing works, Grenade [40] integrates a pre-trained language model (PLM) encoder and a GNN encoder to jointly capture rich textual semantics and graph structural information, aligning their learned representations via contrastive learning on text-attributed graphs. Similarly, GCLR [41] employs carefully designed prompting strategies to elicit more reliable and

TABLE II

Graph node clustering performance comparison of our NeuCGC against GCLR-refined baselines on Cora and Citeseer datasets. Four metrics (in %) are used to evaluate the clustering results. “$-\mathrm{starting}$” means the starting performance of each baseline, “$-\mathrm{GCLR}-\mathrm{Mixtral}$” indicates the baseline model is refined by GCLR with $\mathrm{Mixtral}-8\mathrm{x}-7\mathrm{b}$, while “$-\mathrm{GCLR}-\mathrm{ChatGPT}$” represents it is refined by GCLR with $\mathrm{ChatGPT}$.

Method	**Cora**				**Citeseer**			
	ACC	**NMI**	**ARI**	**F1**	**ACC**	**NMI**	**ARI**	**F1**
DiffPool-starting	60.0	43.5	36.6	56.8	47.1	25.6	23.1	43.1
DiffPool-GCLR-Mixtral	51.4	22.2	6.5	29.1	59.0	26.8	19.7	41.4
DiffPool-GCLR-ChatGPT	59.3	39.6	29.9	51.3	55.7	36.2	30.9	49.1
DinkNet-starting	68.3	52.0	44.2	62.1	66.5	43.1	42.4	60.4
DinkNet-GCLR-Mixtral	65.2	23.4	9.3	27.4	67.4	37.0	27.2	47.9
DinkNet-GCLR-ChatGPT	64.7	50.4	36.4	55.9	69.7	45.7	45.3	65.6
DMoN-starting	57.6	41.6	33.8	50.9	47.9	28.5	24.3	43.7
DMoN-GCLR-Mixtral	56.7	30.1	13.7	29.4	49.9	27.1	14.5	29.9
DMoN-GCLR-ChatGPT	61.4	42.6	34.0	53.9	49.0	30.6	26.7	44.2
MinCutPool-starting	64.2	48.9	40.4	58.3	64.2	44.4	42.0	61.7
MinCutPool-GCLR-Mixtral	61.6	41.6	30.5	54.0	67.5	39.6	35.8	59.8
MinCutPool-GCLR-ChatGPT	71.5	53.8	50.0	65.0	68.0	47.0	46.0	65.4
NeuCGC	77.1	59.0	56.3	75.8	72.5	46.7	48.1	64.0

informative feedback from LLMs for unsupervised graph node clustering.

5.3.2 Distinctions Between LLM-Empowered and Traditional Graph Learning Methods

Traditional graph learning methods primarily rely on GNNs to encode structural information through message-passing mechanisms, while LLM-empowered methods integrate LLMs to incorporate semantic reasoning, cross-domain and cross-modal alignment, and generative augmentation capabilities [30]. This hybridization addresses limitations in handling text-rich or reasoning-intensive graph tasks. Below are detailed distinctions across core aspects [42]:

1) **Input Semantics**. *Traditional*: primarily works on numerical features and topology. These can be sparse, high-dimensional, and lack deep contextual meaning. *LLM-empowered*: works on contextual semantic embeddings from LLMs. These are dense, low-dimensional, and capture nuanced meaning, synonyms, and external knowledge that go beyond structural adjacency [43], [44].
2) **Knowledge Utilization**. *Traditional*: relies solely on information contained within the given graph (attributes, topology, or handcrafted features). *LLM-empowered*: can inject external world knowledge encoded in the pre-trained LLMs, thereby enriching node semantics and enabling inference on sparse or noisy graphs. This external knowledge transfer improves performance on tasks where structural information alone is insufficient [45], [46].
3) **Learning Mechanism**. *Traditional*: typically task-specific and trained from scratch or with fine-tuning on labeled graphs. *LLM-empowered*: introduces prompt-based or in-context learning mechanisms, enabling flexible adaptation to new tasks in a zero-shot or few-shot manner. This reduces dependency on large-scale annotated data and enhances generalization [47], [48].
4) **Interpretability and Reasoning**. *Traditional*: mainly captures low-level structural patterns, they lack explicit reasoning capability. *LLM-empowered*: can provide interpretable, human-readable rationales for predictions, fostering explainability in graph-based decision processes, due to the text-based reasoning and explanation abilities of LLMs [49].
5) **Robustness and Noise Tolerance**. *Traditional*: the performance depends on graph quality, such as node attributes, links. *LLM-empowered*: can correct ambiguous or noisy text via reasoning, but are sensitive to LLM hallucination and prompt quality–often need filtering/alignment [50].
6) **Computational Characteristics**. *Traditional*: computationally efficient and scalable for large graphs but often limited in semantic understanding. *LLM-empowered*: may be more expensive (API calls, larger models) and can face latency/budget constraints; many works design selective prompting or hybrid schemes to control cost [41].
7) **Applications**. *Traditional*: focuses on structure-centric domains like social networks, bioinformatics, and citation analysis. *LLM-empowered*: extends to text-rich scenarios such as knowledge graph, molecular discovery, personalized recommendation, and interactive reasoning [30].

5.3.3 Performance Comparison with LLM-Based Graph Clustering Methods

In this part, we conduct a performance comparison between the proposed NeuCGC and the latest LLM-empowered graph clustering methods. As previously discussed, the majority of existing LLM-empowered graph learning techniques are designed for supervised tasks, such as node classification and link prediction, or other clustering tasks like text clustering. Consequently, our comparative analysis is primarily focused on a recent unsupervised graph node clustering method, GCLR [41].

GCLR operates as an active learning framework, whose core mechanism involves using selective prompts to obtain feedback from an LLM oracle, which is then incorporated into a GNN-based clustering model via a fine-tuning process to enhance its performance [41]. Since GCLR explores various combinations of oracle LLMs, feedback mechanisms, and fine-tuning losses, we select the configuration demonstrated to be most effective in its original paper—specifically, the one utilizing LLM feedback with a cross-entropy loss—for a fair and meaningful comparison.

For this experiment, we directly report the results as presented in the GCLR paper. The comparative results, which utilize Mixtral-8x-7b [51] and ChatGPT-3.5-Turbo [52] as the oracle LLMs, and four graph clustering baselines–DiffPool [53], DinkNet [54], DMoN [55], and MinCutPool [56]–as finetuning models, are detailed in Table II, respectively.

From these results, several key observations can be drawn:

1) LLM-empowered methods indeed bring certain performance improvements on some datasets or baselines, but the gains are generally limited, and in some cases, even show performance degradation. For instance, both DiffPool and DinkNet exhibit decreased accuracy on the Cora dataset after being refined by GCLR. The magnitude and direction of performance changes vary across baselines, datasets, and LLM oracles. For example, DMoN and MinCutPool experience a drop in ACC when refined with Mixtral-8x-7b on Cora, but a slight improvement when refined with ChatGPT; in contrast, such improvements are not observed on Citeseer with respect to ACC. These mixed results suggest that the field of LLM-empowered graph clustering still holds substantial room for exploration and optimization.
2) Our proposed NeuCGC consistently outperforms other compared methods across most metrics, demonstrating its superior clustering capability. Although DinkNet-GCLR-ChatGPT and MinCutPool-GCLR-ChatGPT achieve slightly higher F1-scores than NeuCGC, this further confirms that LLMs possess the potential to provide valuable guidance and enhancement for graph clustering.

Based on the above analysis, we conclude that exploring how to better leverage LLMs to unlock their full potential for unsupervised graph clustering remains a highly promising and impactful research direction for the future. In particular, designing a framework that enables LLM-based refinement to yield consistent and substantial clustering improvements regardless of dataset characteristics or baseline choice represents a realistic and highly meaningful challenge.

References

[1] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[2] Y. Liu, J. Xia, S. Zhou, X. Yang, K. Liang, C. Fan, Y. Zhuang, S. Z. Li, X. Liu, and K. He, "A survey of deep graph clustering: Taxonomy, challenge, application, and open resource," *arXiv preprint arXiv:2211.12875*, 2022.

[3] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[4] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 807–816.

[5] B. Rozemberczki, C. Allen, R. Sarkar, and x. Thilo Gross, "Multi-scale attributed node embedding," *Journal of Complex Networks*, vol. 9, no. 1, pp. 1–22, 2021.

[6] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proceedings of the Web Conference 2020*, 2020, pp. 1400–1410.

[7] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, and J. Cheng, "Deep fusion clustering network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9978–9987.

[8] W. Jin, X. Liu, X. Zhao, Y. Ma, N. Shah, and J. Tang, "Automated self-supervised learning for graphs," in *International Conference on Learning Representations*, 2022.

[9] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "Graphmae: Self-supervised masked graph autoencoders," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 594–604.

[10] P. Zhu, Q. Wang, Y. Wang, J. Li, and Q. Hu, "Every node is different: Dynamically fusing self-supervised tasks for attributed graph clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, 2024, pp. 17 184–17 192.

[11] A. Bhowmick, M. Kosan, Z. Huang, A. Singh, and S. Medya, "Dgcluster: A neural framework for attributed graph clustering via modularity maximization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 10, 2024, pp. 11 069–11 077.

[12] Y. Liu, W. Tu, S. Zhou, X. Liu, L. Song, X. Yang, and E. Zhu, "Deep graph clustering via dual correlation reduction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7603–7611.

[13] L. Gong, S. Zhou, W. Tu, and X. Liu, "Attributed graph clustering with dual redundancy reduction," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 2022, pp. 3015–3021.

[14] X. Yang, C. Tan, Y. Liu, K. Liang, S. Wang, S. Zhou, J. Xia, S. Z. Li, X. Liu, and E. Zhu, "Convert: Contrastive graph clustering with reliable augmentation," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 319–327.

[15] Y. Liu, X. Yang, S. Zhou, X. Liu, S. Wang, K. Liang, W. Tu, and L. Li, "Simple contrastive graph clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13 789–13 800, 2024.

[16] X. Shen, D. Sun, S. Pan, X. Zhou, and L. T. Yang, "Neighbor contrastive learning on learnable graph augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 9782–9791.

[17] W. Xia, Q. Wang, Q. Gao, M. Yang, and X. Gao, "Self-consistent contrastive attributed graph clustering with pseudo-label prompt," *IEEE Transactions on Multimedia*, vol. 25, pp. 6665–6677, 2023.

[18] X. Yang, Y. Liu, S. Zhou, S. Wang, W. Tu, Q. Zheng, X. Liu, L. Fang, and E. Zhu, "Cluster-guided contrastive graph clustering network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 10 834–10 842.

[19] Y. Liu, X. Yang, S. Zhou, X. Liu, Z. Wang, K. Liang, W. Tu, L. Li, J. Duan, and C. Chen, "Hard sample aware network for contrastive deep graph clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8914–8922.

[20] T. Xiao, H. Zhu, Z. Chen, and S. Wang, "Simple and asymmetric graph contrastive learning without augmentations," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 16 129–16 152.

[21] C. Wang, Y. Liu, Y. Yang, and W. Li, "Hetergcl: Graph contrastive learning framework on heterophilic graph," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 2024, pp. 2397–2405.

[22] J. Zhu, R. A. Rossi, A. Rao, T. Mai, N. Lipka, N. K. Ahmed, and D. Koutra, "Graph neural networks with heterophily," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 168–11 176.

[23] X. Zheng, Y. Wang, Y. Liu, M. Li, M. Zhang, D. Jin, P. S. Yu, and S. Pan, "Graph neural networks for graphs with heterophily: A survey," *arXiv preprint arXiv:2202.07082*, 2024.

[24] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, "Permutation equivariant graph framelets for heterophilous graph learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11 634–11 648, 2024.

[25] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[26] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.

[27] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.

[28] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 5, pp. 1–72, 2025.

[29] H. Wang, S. Feng, T. He, Z. Tan, X. Han, and Y. Tsvetkov, "Can language models solve graph problems in natural language?" in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 30 840–30 861.

[30] M. Li, P. Zhang, W. Xing, Y. Zheng, K. Zaporojets, J. Chen, R. Zhang, Y. Zhang, S. Gong, J. Hu *et al.*, "A survey of large language models for data challenges in graphs," *Expert Systems with Applications*, p. 129643, 2025.

[31] J. Yu, Y. Ren, C. Gong, J. Tan, X. Li, and X. Zhang, "Leveraging large language models for node generation in few-shot learning on text-attributed graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 13 087–13 095.

[32] Z. Chen, H. Mao, H. Wen, H. Han, W. Jin, H. Zhang, H. Liu, and J. Tang, "Label-free node classification on graphs with large language models (llms)," *arXiv preprint arXiv:2310.04668*, 2023.

[33] X. He, X. Bresson, T. Laurent, A. Perold, Y. LeCun, and B. Hooi, "Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning," *arXiv preprint arXiv:2305.19523*, 2023.

[34] H. Yan, C. Li, J. Yin, Z. Yu, W. Han, M. Li, Z. Zeng, H. Sun, and S. Wang, "When graph meets multimodal: Benchmarking and meditating on multimodal attributed graph learning," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 5842–5853.

[35] Y. He, Y. Sui, X. He, Y. Liu, Y. Sun, and B. Hooi, "Unigraph2: Learning a unified embedding space to bind multimodal graphs," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 1759–1770.

[36] Z. Guo, L. Xia, Y. Yu, Y. Wang, K. Lu, Z. Huang, and C. Huang, "Graphedit: Large language models for graph structure learning," *arXiv preprint arXiv:2402.15183*, 2024.

[37] B. Pan, Z. Zhang, Y. Zhang, Y. Hu, and L. Zhao, "Distilling large language models for text-attributed graph learning," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 1836–1845.

[38] A. Zolnai-Lucas, J. Boylan, C. Hokamp, and P. Ghaffari, "Stage: Simplified text-attributed graph embeddings using pre-trained llms," *arXiv preprint arXiv:2407.12860*, 2024.

[39] Z. Hu, Y. Li, Z. Chen, J. Wang, H. Liu, K. Lee, and K. Ding, "Let's ask gnn: Empowering large language model for graph in-context learning," *arXiv preprint arXiv:2410.07074*, 2024.

[40] Y. Li, K. Ding, and K. Lee, "Grenade: Graph-centric language model for self-supervised representation learning on text-attributed graphs," *arXiv preprint arXiv:2310.15109*, 2023.

[41] P. Trivedi, N. Choudhary, E. W. Huang, V. N. Ioannidis, K. Subbian, and D. Koutra, "Large language model guided graph clustering," in *The Third Learning on Graphs Conference*, 2024.

[42] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, "Large language models on graphs: A comprehensive survey," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[43] D. Wang, Y. Zuo, F. Li, and J. Wu, "Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 5950–5973.

[44] X. Zhu, H. Xue, Z. Zhao, W. Xu, J. Huang, M. Guo, Q. Wang, K. Zhou, and Y. Zhang, "Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models," *arXiv preprint arXiv:2503.03313*, 2025.

[45] Y.-H. Lin, Q.-H. Chen, Y.-J. Cheng, J.-R. Zhang, Y.-H. Liu, L.-Y. Hsia, and Y.-N. Chen, "Llm inference enhanced by external knowledge: A survey," *arXiv preprint arXiv:2505.24377*, 2025.

[46] Z. Song, B. Yan, Y. Liu, M. Fang, M. Li, R. Yan, and X. Chen, "Injecting domain-specific knowledge into large language models: a comprehensive survey," *arXiv preprint arXiv:2502.10708*, 2025.

[47] Y. Wu, G. Lu, Y. Zuo, H. Zhang, and J. Wu, "Graph-r1: Incentivizing the zero-shot graph learning capability in llms via explicit reasoning," *arXiv preprint arXiv:2508.17387*, 2025.

[48] Y. Li, Y. Yang, J. Zhu, H. Chen, and H. Wang, "Llm-empowered few-shot node classification on incomplete graphs with real node degrees," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 1306–1315.

[49] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," *arXiv preprint arXiv:2310.01061*, 2023.

[50] Z. Wang, Z. Gao, A. Kharel, and I.-Y. Ko, "Are llms better gnn helpers? rethinking robust graph learning under deficiencies with iterative refinement," *arXiv preprint arXiv:2510.01910*, 2025.

[51] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[52] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," in *Advances in neural information processing systems*, vol. 35, 2022, pp. 27 730–27 744.

[53] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in neural information processing systems*, vol. 31, 2018.

[54] Y. Liu, K. Liang, J. Xia, S. Zhou, X. Yang, X. Liu, and S. Z. Li, "Dink-net: Neural clustering on large graphs," in *International conference on machine learning*. PMLR, 2023, pp. 21 794–21 812.

[55] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller, "Graph clustering with graph neural networks," *Journal of Machine Learning Research*, vol. 24, no. 127, pp. 1–21, 2023.

[56] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International conference on machine learning*. PMLR, 2020, pp. 874–883.