MARK: Multi-agent Collaboration with Ranking Guidance for Text-attributed Graph Clustering

Yiwei Fu[1*], Yuxing Zhang[2*], Chunchun Chen[3*], Jianwen Ma[1], Quan Yuan[2], Rong-Cheng Tu[4], Xinli Huang[2], Wei Ye[3,5], Xiao Luo[6†], Minghua Deng[1†]

[1] Peking University [2] East China Normal University [3] Tongji University
[4] Nanyang Technological University [5] Shanghai Innovation Institute [6] UCLA

{fuyw, jianwen_ma}@stu.pku.edu.cn, {zhangyuxing, quanyuan}@stu.ecnu.edu.cn
{c2chen, yew}@tongji.edu.cn, rongcheng.tu@ntu.edu.sg
xlhuang@cs.ecnu.edu.cn, xiaoluo@cs.ucla.edu, dengmh@math.pku.edu.cn

Abstract

This paper studies the problem of text-attributed graph clustering, which aims to cluster each node into different groups using both textual attributes and structural information. Although graph neural networks (GNNs) have been proposed to solve this problem, their performance is usually limited when uncertain nodes are near the cluster boundaries due to label scarcity. In this paper, we introduce a new perspective of leveraging large language models (LLMs) to enhance text-attributed graph clustering and develop a novel approach named Multi-agent Collaboration with Ranking Guidance (MARK). The core of our MARK is to generate reliable guidance using the collaboration of three LLM-based agents as ranking-based supervision signals. In particular, we first conduct the coarse graph clustering, and utilize a concept agent to induce the semantics of each cluster. Then, we infer the robustness under perturbations to identify uncertain nodes and use a generation agent to produce synthetic text that closely aligns with their topology. An inference agent is adopted to provide the ranking semantics for each uncertain node in comparison to its synthetic counterpart. The consistent feedback between uncertain and synthetic texts is identified as reliable guidance for fine-tuning the clustering model within a ranking-based supervision objective. Experimental results on various benchmark datasets validate the effectiveness of the proposed MARK compared with competing baselines.

1 Introduction

Graph Neural Networks (GNNs) (Hamilton et al., 2017; Veličković et al., 2017), a popular method to handle graph, can learn the overall structural information within graphs but are unable to directly process the informative text associated with nodes. With the advent of the LLMs era (Guo et al., 2025), research on integrating LLMs with GNNs for TAGs has garnered increasing attention in recent years (Jin et al., 2024). However, most studies (Chen et al., 2024c; He et al., 2024; Huang et al., 2024; Bi et al., 2024) have focused on supervised tasks, such as node classification and link prediction, leaving text-attributed graph clustering—an unsupervised task—largely underexplored.

Text-attributed graph clustering utilizes both structural and textual information to assign nodes to distinct clusters, ensuring that similar nodes are grouped together while dissimilar nodes are dispersed across separate clusters. Since cluster assignments can intuitively reflect communities within a graph, many real-world problems can be formulated as graph clustering tasks, such as identifying friend groups in social networks (Hartup, 2022) or recommending papers to researchers within the same field (Wu et al., 2022).

Existing methods (Trivedi et al., 2024), after obtaining cluster assignments from the clustering model, utilize an LLM-based agent to decompose the task of querying the clustering categories of nodes into several sub-tasks, which are then processed sequentially to derive the final feedback. After filtering out potential incorrect feedback based on the LLM's prediction confidence scores, the clustering model is then fine-tuned using cross-entropy loss with the feedback-based pseudo-labels. However, three challenges still remain: **(1)** *Can we use multi-agent cooperation to query clustering categories?* Queries based on a single agent lack breadth of knowledge, resulting in an overly narrow perspective (Guo et al., 2024). **(2)** *Can we develop filtering strategies that consider both text semantics and graph topology?* The noisy feedback filtering strategy that relies solely on prediction confidence utilizes the prior knowledge of LLMs but fails to leverage the topological perspective from the graph. **(3)** *Can we choose a more robust fine-tuning loss*

[*] Equal contribution, co-first authors.
[†] Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2025, pages 6057–6072
July 27 - August 1, 2025

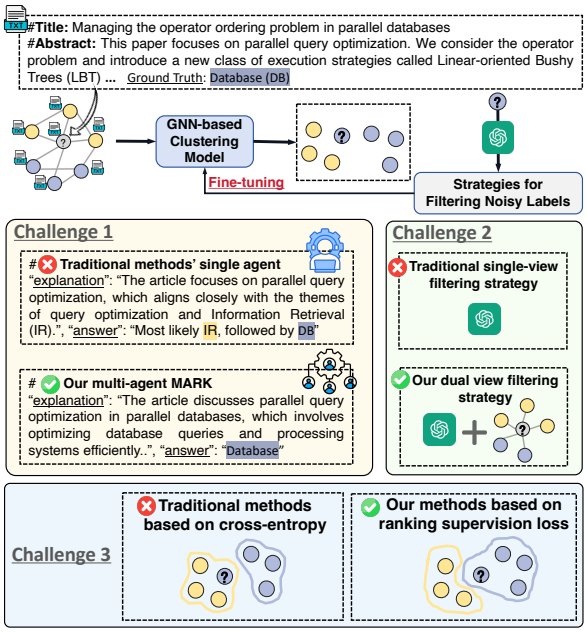


Figure 1: Potential challenges exist before and advantages of MARK compared with traditional methods.

function? When the pseudo-labels contain noisy feedbacks that may be missed during filtering, the cross-entropy loss is not robust, leading to overfitting on incorrect clusters (Feng et al., 2021).

In this paper, we introduce a new perspective of leveraging LLMs to enhance text-attributed graph clustering and develop a novel approach named Multi-agent Collaboration with Ranking Guidance (MARK). The core of our MARK is to generate reliable guidance using the collaboration of three LLM-based agents as ranking-based supervision signals. Specifically, through empirical analysis, we find that fine-tuning the graph clustering model on uncertain nodes yields better gains compared to doing so on certain or random nodes. Inspired by this, we perform a graph clustering model on two perturbation graphs in the ***Graph Clustering Engine***. We identify the nodes with inconsistent cluster assignments between the two views as uncertain nodes and employ three agents to query their cluster categories. In the ***Concept Agent***, we first induce the concept of each cluster using the nodes close to the center of each cluster. We calculate the similarity between uncertain nodes and their neighbors within the corresponding ego-graph to select a few representative neighbors for each uncertain node. To reduce the cost of querying neighbor representatives from the LLM, we use a ***Generation Agent*** to summarize the neighbor representatives and generate synthetic text with equivalent semantics. In the ***Inference Agent***, we construct the inference prompt by appending the cluster concepts obtained from the ***Concept Agent*** to both the uncertain text and the synthetic text. Since the synthetic text captures both the topological and semantic information of the uncertain node, we use the consistency of the agent's feedback on the uncertain text and the synthetic text as a filtering criterion. The feedback may still be inconsistent with the true labels of uncertain nodes, even after filtering. Therefore, we adopt a ranking-based supervision objective during fine-tuning to enhance the robustness of node representations, thereby mitigating the adverse effects of noise. Overall, the coarse clustering assignments generated by the graph clustering engine provide topological information to the multi-agent framework. In turn, the collaborative decisions from the agents enhance the performance of the graph clustering engine.

To summarize, the contributions are as follows: (1) *New Perspective*. We are the first to connect text-attribute graph clustering with a multi-agent framework for reliable semantics. (2) *Novel Methodology*. Our MARK leverage the collaboration of a concept agent, a generation agent and an inference agent to provide ranking signals, which are utilized to guide the graph clustering. (3) *Extensive Experiments*. Experimental results on various benchmark datasets validate the effectiveness of the proposed MARK compared with competing baselines.

2 Related Work

Recently, research on the integration of Large Language Models (LLMs) and Graph Neural Networks (GNNs) in text-attribute graphs has garnered increasing attention (Chen et al., 2023; Zhang et al., 2025; Zhao et al., 2025). Researches can be divided into three categories: (1) LLM-as-encoder. (Wen and Fang, 2023; Zhao et al., 2023a; Jin et al., 2023) uses pre-trained language models (PLMs) (Reimers, 2019) to encode texts, achieving a hybrid architecture of PLM and GNN. To obtain expressive representations, (Zhu et al., 2024) utilizes the hidden embeddings of LLMs to construct node embeddings. (2) LLM-as-predictor. (Chen et al., 2024a; Zhao et al., 2023b; Tang et al., 2024) serializes the graph into various sequences, which are then fed into the LLM for prediction. (Yang et al., 2021; Yasunaga et al., 2022) adjust traditional LLM architecture to conduct joint text and graph encoding. (3) LLM-as-reasoner. Under the guidance of carefully designed prompts, (Chen

et al., 2024b; Bi et al., 2024; Wu et al., 2025) enables LLMs to perform inference tasks for node class or link existence.

The above works almost focus on node classification and link prediction, while text-attributed graph clustering remains under-explored. To the best of our knowledge, GCLR (Trivedi et al., 2024) is currently the only study dedicated to text-attributed graph clustering. GCLR employs a contrastive (Liu et al., 2023a) and pooling-based (Tsitsulin et al., 2023; Bianchi et al., 2020; Ying et al., 2018) attributed graph clustering model as its backbone. GCLR adopts an LLM-as-reasoner architecture described above, where the LLM solely offers feedback for the fine-tuning of the GNN, thereby eliminating the costly pretraining and fine-tuning of LLMs. Despite the success, the three challenges discussed in the previous section require further exploration in this paper.

3 Method

3.1 Problem Formulation

$\mathcal{G} = (\mathbf{A}, \mathcal{D}, \mathbf{X})$ is a text-attributed graph (TAG) with the adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, the raw texts $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^N\}$ and PLM-encoded textual features $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{N\times D}$ (Reimers, 2019). N denotes the number of nodes. The node embeddings encoded by GNN are represented by $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\} \in \mathbb{R}^{N\times F}$. D and F are the dimensions of node features and embeddings, respectively. Graph clustering aims to partition the nodes in the text-attributed graph $\mathcal{G}$ into K disjoint clusters $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$. The embedding of the cluster centers is represented by $\{\mu_1, \mu_2, \ldots, \mu_K\} \in \mathbb{R}^{K\times F}$.

3.2 Overview of MARK

Following the two-stage learning style (Zhang et al., 2023; Trivedi et al., 2024), our aim is to leverage the collaboration of multi-agent to provide ranking signals for the graph clustering.

As described in Figure 2, the proposed MARK includes a graph clustering engine and three supporting agents: **(1)** Graph clustering engines build upon the shoulders of existing advanced graph clustering models, leveraging their sophisticated representation capabilities to perform contrastive learning on two perturbed views. By seeking robustness in cluster assignments across these two views, uncertain nodes are identified. **(2)** The concept agent $\mathcal{M}_{con}$ selects high-confidence nodes from each cluster to induce cluster names. The induced concepts will serve as foundational knowledge for the other two agents, enhancing their capability to execute specific tasks. **(3)** The generation agent $\mathcal{M}_{gen}$ synthesizes virtual text by aggregating the neighboring texts of each uncertain node, thereby enhancing data diversity while considering the neighborhood topology. **(4)** The inference agent $\mathcal{M}_{inf}$ filters out low-confidence uncertain nodes by evaluating the consistency of feedback from LLMs between uncertain texts and synthetic texts. Finally, we adopt a ranking-based supervised objective to utilize the filtered nodes for fine-tuning the graph clustering engine. In the following, we present them step by step.

3.3 Graph Clustering Engine

After obtaining the coarse clustering assignments, it is typically necessary to obtain feedback on informative nodes from the LLM to fine-tune the clustering model. Existing methods (Trivedi et al., 2024) typically select nodes with high entropy as candidate query nodes. However, nodes with low entropy may also have incorrect clustering assignments; thus, selecting only high-entropy nodes rather than all nodes as candidates for querying can introduce bias. To address this challenge, we have developed a robustness-based mechanism for identifying potentially valuable candidate query nodes. We employ a dual-view contrastive learning framework to detect clustering discrepancies under perturbations, treating these non-robust nodes as uncertain nodes for querying. The fundamental principle behind this mechanism is to find nodes with ambiguous category perceptions, which are prone to inconsistent judgments under minor perturbations.

Specifically, we apply perturbations at both feature level and edge level to obtain two augmented views of the graph, and then feed them separately into a shared GNN encoder to get node representations $\mathbf{H}', \mathbf{H}'' \in \mathbb{R}^{N\times F}$, where F means the dimension of the node representations. We align the two views by alignment loss $\mathcal{L}_{ali}$, as shown below:

$$\mathcal{L}_{ali} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathrm{sim}(\mathbf{h}'_i, \mathbf{h}''_i)/\tau)}{\sum_{k=1}^{N}\exp(\mathrm{sim}(\mathbf{h}'_i, \mathbf{h}''_k)/\tau)}, \tag{1}$$

where $\mathbf{h}'_i$ and $\mathbf{h}''_i$ are the representations of two augmentations with regard to the i-th node, $\mathrm{sim}(*)$ indicates the cosine similarity function, τ controls the temperature to adjust the distribution. The clustering loss $\mathcal{L}_{clu}$ serves as a flexible plugin, enabling

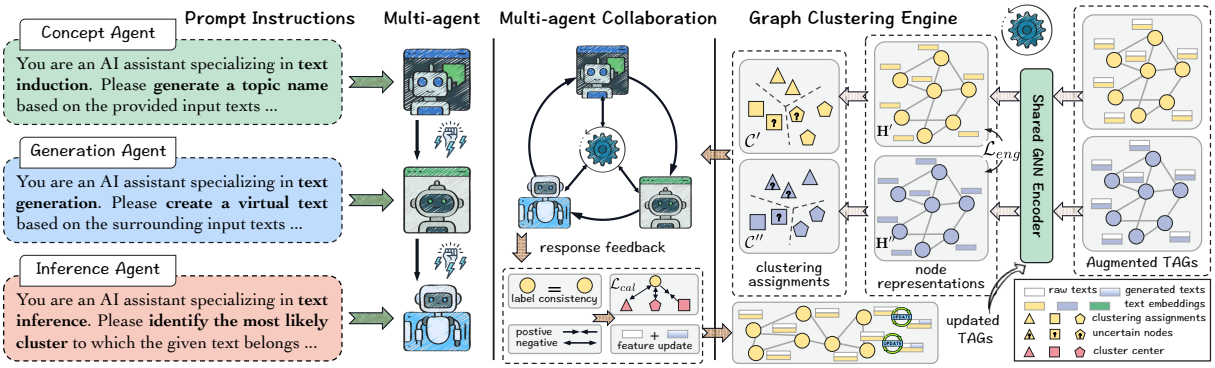
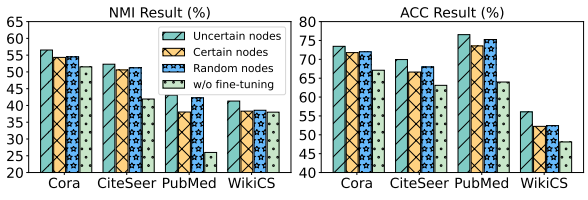


Figure 2: Overview of MARK. A graph clustering engine is introduced to identify uncertain nodes. Three agents (the concept agent, generation agent, and inference agent) collaborate to provide ranking signals for these uncertain nodes. Finally, consistency feedback serves as a reliable guide for fine-tuning the clustering engine.



(a) NMI metric (b) ACC metric

Figure 3: Performance comparison of multi-agent collaboration using different node sets.

MARK to utilize the clustering capabilities of any state-of-the-art graph clustering model. The pre-training loss for the engine is defined as follows:

$$\mathcal{L}_{eng} = \mathcal{L}_{ali} + \mathcal{L}_{clu}. \tag{2}$$

Subsequently, embeddings $\mathbf{H}'$ and $\mathbf{H}''$ are used to derive the clustering assignments $\mathcal{C}'$ and $\mathcal{C}''$, respectively. We construct GNN-uncertain node set $\mathcal{S}$ for querying LLMs, as shown below:

$$\mathcal{S} = \{i \in \mathcal{G} \mid \mathcal{C}'_i \neq \mathcal{C}''_i\}. \tag{3}$$

An empirical analysis has been performed to explain our motivation. As shown in Figure 3, the results show that, given the ground-truth labels, fine-tuning the clustering model using the set of uncertain nodes $\mathcal{S}$ yields the most significant performance improvement compared to using certain nodes or randomly selected nodes. This highlights the necessity of prioritizing uncertain nodes (Zhang et al., 2025; Chen et al., 2023).

3.4 Multi-agent Collaboration

Concept Agent for Cluster Induction After obtaining cluster assignments $\widehat{\mathcal{C}} = \{\widehat{\mathcal{C}}_1, \widehat{\mathcal{C}}_2, \cdots, \widehat{\mathcal{C}}_K\}$ generated by the graph clustering engine, we select the top-n samples that are closest to the center of each cluster to construct set $\mathcal{D}_{clo} = \{\mathcal{D}^1_{clo}, \mathcal{D}^2_{clo}, \cdots, \mathcal{D}^K_{clo}\}$. Through querying $\mathcal{M}_{con}$, we acquire the concept $\mathcal{F}_{con}$ for each cluster along with its corresponding explanation,

$$\mathcal{F}_{con} = \mathcal{M}_{con}(\mathcal{D}^1_{clo}, \mathcal{D}^2_{clo}, \cdots, \mathcal{D}^K_{clo}). \tag{4}$$

The acquired cluster concepts $\mathcal{F}_{con}$ will be leveraged to craft the following two agents with more precisely defined identities, thereby enhancing their capabilities for specific tasks. The detailed agent design can be found in Figure 6.

Generation Agent for Neighbor Summary In order to filter out the LLM responses in $\mathcal{S}$ that do not take topology into account, it is a natural approach to consider the neighborhood information of these nodes as an aid in the decision-making process (Chen et al., 2024b). Unlike GNNs, which excel at efficiently aggregating information from neighboring nodes, LLMs face significant challenges in this regard. To this end, we use $\mathcal{M}_{gen}$ to aggregate neighbors and generate virtual synthetic text $\mathcal{F}_{syn}$. Following the similarity-based neighbor selection strategy (Li et al., 2024), we select the top-k most similar samples from the neighbors of uncertain nodes to form neighbor description set $\mathcal{D}_{nei}$. Subsequently, we input both the concepts $\mathcal{F}_{con}$ learned by $\mathcal{M}_{con}$ and $\mathcal{D}_{nei}$ into $\mathcal{M}_{gen}$, which then generates new samples $\mathcal{F}_{syn}$,

$$\mathcal{F}^i_{syn} = \mathcal{M}_{gen}(\mathcal{D}^i, \mathcal{D}^i_{nei}, \mathcal{F}_{con}). \tag{5}$$

The virtual texts not only capture the topological relationships among nodes within the graph but also encapsulate the semantic context associated with those nodes. They play a key role in filtering uncertain nodes for the following agent. The detailed agent design is listed in Figure 7.

Inference Agent for Decision Filtration We construct the inference prompt by appending the cluster concepts obtained from the Concept Agent to both the uncertain node's text and the synthetic text. By querying $\mathcal{M}_{inf}$, we get their inferred cluster categories. Since the synthetic text captures both the topological and semantic information of the uncertain node, we use the consistency of the agent's feedback on the uncertain text and the synthetic text as a filtering criterion to obtain Agents-resolvable node set $\mathcal{R}$, as shown below:

$$\mathcal{F}_{inf}^{i}, \mathcal{F}_{inf}^{i\_syn} = \mathcal{M}_{inf}(\mathcal{D}^{i}, \mathcal{F}_{syn}^{i}, \mathcal{F}_{con}), \quad (6)$$

$$\mathcal{R} = \{i \in \mathcal{S} \mid \mathcal{F}_{inf}^{i} = \mathcal{F}_{inf}^{i\_syn}\}. \quad (7)$$

The detailed agent design is listed in Figure 8. In response to the LLM's feedback, we directly leverage the knowledge contained in the generated texts to refine the shallow node embeddings $\mathbf{X}$, rather than relying on explanations provided by the LLM (Qiao et al., 2025). Specifically, we encode the generated texts corresponding to the nodes in $\mathcal{R}$ using a Pre-trained Language Model (PLM), then update $\mathbf{X}$ by averaging the sum of these encodings with the original, which can be expressed as:

$$\mathbf{x}_i' = \frac{\mathbf{x}_i + \text{PLM}(\mathcal{F}_{syn}^{i})}{2}, \quad (8)$$

where $\mathbf{x}_i'$ denotes the updated features guided by the knowledge of LLMs and $i \in \mathcal{R}$.

3.5 Fine-Tuning with Ranking Guidance

Three agents collaborate to filter nodes from the GNN-uncertain node set $\mathcal{S}$, ultimately obtaining the Agents-resolvable node set $\mathcal{R}$. This process is equivalent to conducting a topology-level and semantic-level ranking of the nodes within the GNN-uncertain node set through the efforts of multiple agents. The filtered nodes are those with high rankings in the $\mathcal{S}$. We should focus on these uncertain nodes that are considered reliable by the multi-agent system (Luo et al., 2025; Liu et al., 2024a), as they can provide ranking-based supervisory signals for fine-tuning.

Existing methods (Trivedi et al., 2024) use cross-entropy to fine-tune the clustering model. However, the feedback may still be inconsistent with the true labels, even after filtering. Due to the lack of robustness of cross-entropy loss to noisy labels (Zhang and Sabuncu, 2018) and potentially inadequate margins (Liu et al., 2016), generalization performance deteriorates. Therefore, we adopt a ranking-based supervision objective during fine-tuning to enhance the robustness of node representations, thereby mitigating the adverse effects of noise.

Algorithm 1: MARK

Input: $\mathcal{G} = (\mathbf{A}, \mathbf{X}, \mathcal{D})$, GNN encoder $\mathcal{F}_\Theta$, training epoch T, multi-agent execution interval T', learning rate β, number of selected nodes n.

```
Use the graph clustering engine for pre-training;
Initialize Agents-resolvable node set R;
for i ← 1 to T do
    Augment G: Obtain G' and G'';
    Encode: H' = F(G') and H'' = F(G'');
    if i % T' == 0 then
        Get the clustering assignments C' and C'' for the augmented graphs by H' and H'';
        Construct the GNN-uncertain node set S;
        // Concept Agent
        Select top-n nodes closest to each cluster center as query samples;
        Induce concept of each cluster by M_con;
        // Generation Agent
        Select neighbors of uncertain nodes in S according to similarity ranking;
        Produce synthetic text for uncertain nodes as a summary of neighbors by M_gen;
        // Inference Agent
        Query the clusters for synthetic texts and uncertain texts;
        Update R by filtering uncertain nodes with cluster consistency by M_inf;
        Update X of the nodes in Q by Eq. 8;
    Calculate the fine-tuning loss L_ft in Eq. 10;
    Update: Θ ← Θ − β · ∇L_ft;
```

Output: Final cluster assignments.

We bring the uncertain nodes closer to the cluster identified by the agent while repelling them from other clusters. Specifically, we first calculate the similarity between the nodes in $\mathcal{R}$ and the cluster centers. We then pair each uncertain node i with its corresponding cluster center as a positive pair $\mathcal{F}_{inf}^{i}$, treating the remaining clusters as negative pairs. We apply contrastive learning to the node-cluster pairs, which acts as a calibration loss for the coarse clustering assignment, expressed as follows:

$$\mathcal{L}_{cal} = -\frac{1}{|\mathcal{R}|} \sum_{i \in \mathcal{R}} \log \frac{\exp\left(\text{sim}\left(\mathbf{h}_i, \boldsymbol{\mu}_{\mathcal{F}_{inf}^{i}}\right)/t\right)}{\sum_{k=1}^{K} \exp\left(\text{sim}\left(\mathbf{h}_i, \boldsymbol{\mu}_k\right)/t\right)}, \quad (9)$$

where t means the temperature, $\boldsymbol{\mu}$ denotes the cluster center. Subsequently, we use $\mathcal{L}_{ft}$ to fine-tune the graph clustering engine, that is,

$$\mathcal{L}_{ft} = \mathcal{L}_{eng} + \mathcal{L}_{cal}. \quad (10)$$

Overall, we consider the graph clustering model that provides uncertain nodes as the engine for three agents, establishing a progressive chain that facilitates collaboration among three agents. In turn, the collaborative decisions made by these three agents offer reliable guidance for clustering. The process of MARK is shown in Algorithm 1.

4 Experiments

4.1 Settings

Dataset We evaluate MARK on four widely used text-attributed graphs (Chen et al., 2023): Cora, CiteSeer, PubMed, and WikiCS. The detailed statistics information is listed in Table 1.

Dataset	#Node	#Edge	#Classes	#Type
Cora	2,708	5,429	7	Citation
CiteSeer	3,186	4,277	6	Citation
PubMed	19,717	44,335	3	Citation
WikiCS	11,701	215,863	10	Wikipedia

Table 1: Statistics of used datasets.

Models For the graph clustering engine backbone, we utilize the state-of-the-art DMoN (Tsitsulin et al., 2023) and MAGI (Liu et al., 2024b) as the baseline models. We select GPT-4o-mini[1] as our agent LLM to generate reliable guidance.

Metrics We adopt four metrics to evaluate clustering results: ACC, NMI, ARI, and F1. The results are reported for five runs with different random seeds. Larger values denote better performance.

4.2 Performance Comparison

To more effectively evaluate MARK's performance on graph clustering tasks, we adopt two classic clustering baselines, DMoN and MAGI, as the backbone clustering engines, deriving two versions of MARK: MARK-DMoN and MARK-MAGI. We then compare their performance using four metrics across four commonly used graph datasets: Cora, CiteSeer, WikiCS, and PubMed. Table 2 presents the clustering performance of MARK with the two distinct backbones. Additionally, DMoN and MAGI represent the original clustering results, while "PRE-TRAIN" indicates the clustering results after the graph clustering engine. Notably, MARK-DMoN and MARK-MAGI display the final clustering performance enhanced by our multi-agent framework.

[1]https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

As shown in Table 2, MARK-MAGI demonstrates a much more substantial improvement over the MAGI backbone across the Cora, CiteSeer, and WikiCS datasets, while displays a smaller increase observed on PubMed, with 1.80% improvement in NMI. Furthermore, MARK-DMoN exhibits the most significant improvement on the WikiCS dataset, with a 20.80% increase in F1 score. In conclusion, MARK not only excels across various evaluation metrics but also proposes a versatile framework for enhancing existing graph clustering methods.

4.3 Ablation Study

To validate the contributions of each proposed agent to MARK, we conduct comprehensive ablation studies on the Cora and CiteSeer datasets. Initially, the concept agent assigns topics based on a selection of typical and high-confidence samples, which significantly aids in cluster exploration. We remove the concept agent and substitute it with the highest-confidence text within each cluster to represent the cluster, resulting in this variant "W/O AGENT 1". To illustrate the necessity of the generation agent, we replace the virtual text synthesized by the generation agent with text randomly sampled from its neighborhood, referred to as "W/O AGENT 2". To demonstrate the effectiveness of selectively accepting labels predicted by the inference agent, we introduce "W/O AGENT 3", in which all labels are directly assigned by the inference agent without any filtering.

As illustrated in Table 3, removing the concept agent results in the most significant performance drop, while label filtering leads to a notable decrease in performance, ranking second in impact. Specifically, "W/O AGENT 1" achieves an ACC of 0.591 on the Cora dataset, representing a 12.2% decrease compared to MARK-MAGI. Meanwhile, the other variants, "W/O AGENT 2" and "W/O AGENT 3" experience smaller ACC reductions of 5.1% and 2.6%, respectively, on the same dataset. In conclusion, the removal of each agent individually causes varying degrees of performance degradation, underscoring the pivotal role each agent plays in the overall efficacy of the designed pipeline.

4.4 Sensitivity Analysis

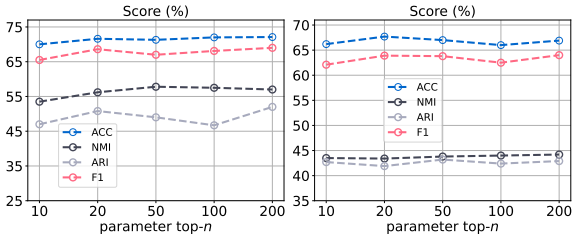
In this section, we present a sensitivity analysis examining two critical aspects: (1) hyper-parameter sensitivity and (2) the impact of LLM selection on

Dataset	Metric	DMoN	Pre-train	MARK-DMoN	Imp ↑	MAGI	Pre-train	MARK-MAGI	Imp ↑
Cora	ACC	0.628±0.080	0.671±0.024	0.675±0.017	4.70%	0.664±0.057	0.697±0.018	**0.713±0.032**	4.90%
	NMI	0.490±0.042	0.515±0.011	0.548±0.005	5.80%	0.529±0.010	0.533±0.004	**0.578±0.007**	4.90%
	ARI	0.437±0.069	0.452±0.001	0.473±0.014	3.60%	0.464±0.048	0.489±0.003	**0.490±0.022**	2.60%
	F1	0.565±0.072	0.618±0.028	0.632±0.026	6.70%	0.614±0.055	0.644±0.035	**0.670±0.037**	5.60%
CiteSeer	ACC	0.628±0.023	0.633±0.023	0.646±0.016	1.80%	0.624±0.059	0.650±0.055	**0.670±0.022**	4.60%
	NMI	0.391±0.005	0.419±0.008	0.423±0.012	3.20%	0.411±0.028	0.424±0.025	**0.438±0.009**	2.70%
	ARI	0.384±0.013	0.401±0.013	0.409±0.017	2.50%	0.396±0.043	0.418±0.039	**0.432±0.014**	3.60%
	F1	0.598±0.016	0.608±0.025	0.620±0.012	2.20%	0.586±0.068	0.616±0.059	**0.638±0.015**	5.20%
WikiCS	ACC	0.326±0.019	0.481±0.021	0.520±0.032	19.40%	0.534±0.049	0.552±0.037	**0.606±0.033**	7.20%
	NMI	0.235±0.023	0.396±0.016	0.416±0.023	18.10%	0.467±0.018	0.476±0.012	**0.493±0.010**	2.60%
	ARI	0.090±0.016	0.213±0.014	0.258±0.044	16.80%	0.381±0.051	0.405±0.039	**0.466±0.021**	8.50%
	F1	0.255±0.023	0.430±0.026	0.463±0.035	20.80%	0.447±0.047	0.457±0.039	**0.515±0.052**	6.80%
PubMed	ACC	0.495±0.026	0.576±0.054	0.613±0.001	11.80%	0.590±0.001	0.589±0.001	**0.615±0.011**	2.50%
	NMI	0.157±0.046	0.194±0.035	0.163±0.001	0.60%	0.180±0.001	0.179±0.001	**0.198±0.018**	1.80%
	ARI	0.140±0.058	0.163±0.024	0.179±0.001	3.90%	0.154±0.002	0.152±0.001	**0.192±0.018**	3.80%
	F1	0.462±0.031	0.576±0.053	0.608±0.001	14.60%	0.590±0.001	0.589±0.001	**0.611±0.009**	2.10%

Table 2: Clustering performance of MARK with two distinct backbones, DMoN and MAGI separately. The boldfaced scores represent the best results.

Dataset	Metric	w/o agent 1	w/o agent 2	w/o agent 3	MARK-MAGI
Cora	ACC	0.591±0.052	0.687±0.036	0.662±0.046	**0.713±0.032**
	NMI	0.482±0.029	0.556±0.012	0.544±0.020	**0.578±0.007**
	ARI	0.346±0.038	0.488±0.041	0.428±0.036	**0.490±0.022**
	F1	0.558±0.073	0.646±0.037	0.614±0.055	**0.670±0.037**
CiteSeer	ACC	0.540±0.056	0.638±0.070	0.608±0.040	**0.670±0.022**
	NMI	0.336±0.031	0.414±0.041	0.400±0.023	**0.438±0.009**
	ARI	0.292±0.045	0.409±0.058	0.374±0.039	**0.432±0.014**
	F1	0.500±0.055	0.595±0.077	0.570±0.056	**0.638±0.015**

Table 3: Ablation Study of MARK-MAGI on Cora and CiteSeer Datasets.



(a) Cora (b) CiteSeer

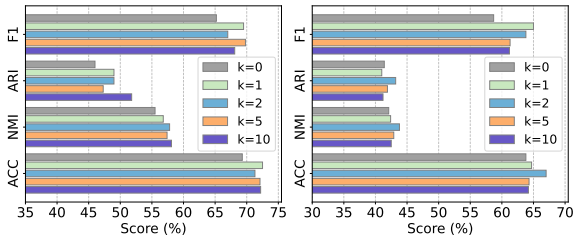
Figure 4: Sensitivity analysis regarding the number of high-confidence samples n for the concept agent. Panels (a) and (b) present the sensitivity analysis results on the Cora and CiteSeer datasets, respectively.

MARK's performance.

To assess the sensitivity of the proposed framework MARK to hyper-parameters, we conduct two analyses on the Cora and CiteSeer datasets. The first analysis examines the hyper-parameter associated with the number of high-confidence samples selected by the concept agent in each cluster, as these samples directly impact the quality of the cluster topics. The second analysis focuses on the hyper-parameter controlling the number of neighbors fed into the generation agent, which greatly influences the quality of the virtual text it synthesizes.

In the context of the concept agent, we investigate the effect of varying the number of high-confidence samples on the topic name derived by the agent, which subsequently influences the cluster assignments determined by the inference agent and ultimately impacts the model's performance. As depicted in Figure 4, a limited number of samples may lead to inaccurate or incomplete topic representations, while an increasing number of samples facilitates the exploration of inter-sample similarities. However, the introduction of additional relationships within the group can cause the clusters to become fragmented, resulting in performance fluctuations that are sensitive to the hyper-parameter settings. As shown in Figure 5, it is observed that the inclusion of solely the uncertain text may significantly impact the quality of the newly generated text, thereby introducing greater uncertainty into MARK. Consequently, we propose generating virtual text that incorporates both the uncertain node and its surrounding neighborhood, leading to a notable performance improvement compared to the approach that disregards neighborhood semantics.

As illustrated in Table 4, we focus on the effect of LLM selection on the final performance of MARK-MAGI. We have incorporated two additional LLMs, GPT-3.5-turbo and Deepseek-R1, to examine the impact of LLM selection. Notably, all three LLMs clearly enhance clustering performance when integrated with ranking guidance, al-



(a) Cora (b) CiteSeer

Figure 5: Sensitivity analysis regarding the number of neighbors k for the generation agent. Panels (a) and (b) present the sensitivity analysis results on the Cora and CiteSeer datasets, respectively.

DATASET	LLM	ACC	NMI	ARI	F1
CORA	GPT-3.5-TURBO	**0.714±0.029**	0.547±0.015	0.502±0.029	**0.683±0.039**
	GPT-4O-MINI	0.713±0.032	**0.578±0.007**	0.490±0.022	0.670±0.037
	DEEPSEEK-R1	0.707±0.034	0.561±0.019	**0.509±0.036**	0.676±0.022
CITESEER	GPT-3.5-TURBO	0.660±0.051	0.432±0.023	0.427±0.038	0.628±0.045
	GPT-4O-MINI	**0.670±0.022**	**0.438±0.009**	**0.432±0.014**	**0.638±0.015**
	DEEPSEEK-R1	0.666±0.036	0.436±0.023	0.431±0.032	0.613±0.046

Table 4: The impact of the chosen LLM on MARK-MAGI's performance. The best results are highlighted in bold.

though the extent of improvement varies depending on the specific LLM, evaluation metrics, and datasets used.

4.5 Case Study

Concept Agent Reveals More Semantic Essence We present cluster 3 from the CiteSeer dataset as a representative example, which is associated with *Artificial Intelligence*. However, we observe that node 1528, labeled as *Machine Learning*, emerges as the highest-confidence sample within cluster 3. This highlights a potential limitation: relying solely on the highest-confidence sample to represent a cluster may not accurately capture the cluster's semantic essence. To address this, we feed the top-n confidence samples into the concept agent, and then provide a comprehensive summary of the cluster's topic. For instance, when using the top-10 confidence samples, the concept agent assigns *Probabilistic Models and Learning in Intelligent Systems* to cluster 4. In contrast, when leveraging the top-200 samples, the derived topic name shifts to *Probabilistic Reasoning and Machine Learning Techniques*. Obviously, the latter aligns much better with the ground truth *Machine Learning*, demonstrating a larger sample size may yields a much more rich semantics.

Generation Agent Synthesizes Context-aware Texts To better explore the quality of generated text considering neighborhood semantics, we take node 1383 from the CiteSeer dataset as a case study. Specifically, node 1383 is one of the uncertain nodes identifying by the graph clustering engine, labeled as *Human Computer Interaction*. As described in Table 9, we then compare the outcomes of two strategies for novel and virtual text generation: the former generates text based solely on the target node 1383 text, while the latter incorporates both the target and neighborhood texts. The target text presents a novel ASL recognition method, which aligns better with *Human Computer Interaction* and is somewhat related to *Artificial Intelligence*. The first generation prompt, focused on ASL frameworks, leads to a misleading predicted label of *Artificial Intelligence*. In contrast, the second prompt, utilizing the graph structure and neighborhood information, generates a text about gesture recognition, which is accurately labeled with *Human Computer Interaction*. In conclusion, including neighborhood texts in the generation prompt may contribute to higher quality text generation, providing a more effective and diverse perspective and enhancing text embeddings.

Inference Agent Filters low quality labels Although the inference agent can predict labels for uncertain nodes by integrating both texts and contexts to some extent, there still exists some randomness and misclassification in the label predictions. Therefore, placing complete trust in the labels predicted by the inference agent could introduce additional noise into our pipeline. Instead, we filter out low-confidence labels by assessing the correspondence between the predicted label for the target text and the synthesized text. As shown in Table 10, the uncertain node 16 from the CiteSeer dataset is misclassified as *Machine Learning* by the inference agent, while its ground truth is *Information Retrieval*.

5 Conclusion

In this paper, we address the challenge of text-attributed graph clustering by leveraging the complementary strengths of graph neural networks (GNNs) and large language models (LLMs). Our proposed MARK is a novel framework that integrates GNNs with multi-agent LLM collaboration to generate robust, ranking-based guidance. To overcome the limitations of current approaches, such as narrow single-agent perspectives, insufficient noise filtering, and fragile supervision, we de-

sign three collaborative agents: the concept agent extracts cluster semantics, the generation agent synthesizes topology-aligned text for uncertain nodes, and the inference agent provides ranking-based feedback by contrasting uncertain and synthetic texts. By using consistent LLM feedback as supervisory signals and incorporating a ranking-aware loss, MARK effectively reduces noise interference and enhances clustering robustness. More importantly, extensive experiments validate that MARK offers a powerful pipeline for augmenting existing GNN-based clustering methods with LLMs.

6 Limitations

While the proposed MARK demonstrates promising performance in text-attributed graph clustering, an important limitation remains to be considered. Although topology alignment is enforced for generated texts, the framework does not explicitly verify factual consistency between synthetic texts and real-world knowledge. This may introduce subtle noise in scenarios requiring strict semantic fidelity.

7 Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant No. 32270689 and Grant No. 62176184).

References

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. 2024. Lpnl: Scalable link prediction with large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3615–3625.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. 2020. Spectral clustering with graph neural networks for graph pooling. In *International conference on machine learning*, pages 874–883. PMLR.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. 2024a. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024b. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2023. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2024c. Label-free node classification on graphs with large language models (llms). In *The Twelfth International Conference on Learning Representations*.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. 2021. Can cross entropy loss be robust to label noise? In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 2206–2212.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Willard W Hartup. 2022. Friendships and their developmental significance. In *Childhood social development*, pages 175–205. Psychology Press.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523*.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. Can gnn be good adapter for llms? In *Proceedings of the ACM on Web Conference 2024*, pages 893–904.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2024. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*.

Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. 2023. Patton: Language model pretraining on text-rich networks. *arXiv preprint arXiv:2305.12268*.

Rui Li, Jiwei Li, Jiawei Han, and Guoyin Wang. 2024. Similarity-based neighbor selection for graph llms. *arXiv preprint arXiv:2402.03720*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Honglin Liu, Peng Hu, Changqing Zhang, Yunfan Li, and Xi Peng. 2024a. Interactive deep clustering via value mining. *Advances in Neural Information Processing Systems*, 37:42369–42387.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. 2016. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295*.

Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Stan Z Li. 2023a. Dink-net: Neural clustering on large graphs. In *International Conference on Machine Learning*, pages 21794–21812. PMLR.

Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2023b. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 8914–8922.

Yunfei Liu, Jintang Li, Yuehe Chen, Ruofan Wu, Ericbk Wang, Jing Zhou, Sheng Tian, Shuheng Shen, Xing Fu, Changhua Meng, et al. 2024b. Revisiting modularity maximization for graph clustering: A contrastive learning perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1968–1979.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. 2025. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*.

Yiran Qiao, Xiang Ao, Yang Liu, Jiarong Xu, Xiaoqian Sun, and Qing He. 2025. Login: A large language model consulted graph neural network training framework. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 232–241.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500.

Puja Trivedi, Nurendra Choudhary, Edward W Huang, Vassilis N Ioannidis, Karthik Subbian, and Danai Koutra. 2024. Large language model guided graph clustering. In *The Third Learning on Graphs Conference*.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. 2023. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21.

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

Zhihao Wen and Yuan Fang. 2023. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 506–516.

Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4425–4445.

Xixi Wu, Yifei Shen, Fangzhou Ge, Caihua Shan, Yizhu Jiao, Xiangguo Sun, and Hong Cheng. 2025. A comprehensive analysis on llm-based node classification algorithms. *arXiv preprint arXiv:2502.00829*.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. 2021. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31.

Taiyan Zhang, Renchi Yang, Yurui Lai, Mingyu Yan, Xiaochun Ye, and Dongrui Fan. 2025. Leveraging large language models for effective label-free node classification in text-attributed graphs. *arXiv preprint arXiv:2412.11983*.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. Clusterllm: Large language models as a guide for text clustering. *arXiv preprint arXiv:2305.14871*.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2023a. Learning on large-scale text-attributed graphs via variational inference. In *The Eleventh International Conference on Learning Representations*.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023b. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.

Yusheng Zhao, Qixin Zhang, Xiao Luo, Weizhi Zhang, Zhiping Xiao, Wei Ju, Philip S Yu, and Ming Zhang. 2025. Dynamic text bundling supervision for zero-shot inference on text-attributed graphs. *arXiv preprint arXiv:2505.17599*.

Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. 2024. Efficient tuning and inference for large language models on textual graphs. *arXiv preprint arXiv:2401.15569*.

A Prompts design for multi-agent framework

The detailed instructions for three agents, concept agent, generation agent and inference agent, are illustrated in Figure 6, Figure 7 and Figure 8.

 # Concept Agent
You are an AI assistant specializing in text induction. Please generate a topic name based on the provided input texts.
Below are the high-confidence samples related to <dataset field> from a specific cluster. Analyze the commonalities and core content of these samples and provide a concise summary of the cluster's theme. Output the theme as a short name without adding any extra explanations. The cluster has the following high-confidence samples: <high-confidence texts of the i-th cluster>. Please comprehensively consider the commonalities between these samples and then conclude the topic of the cluster concisely. Output the newly generated topic name as a dictionary, with keys "answer" and "explanation".

Figure 6: The detailed instruction design of the concept agent for cluster induction.

 # Generation Agent
You are an AI assistant specializing in text generation. Please create a virtual text based on the surrounding input texts.
Given a target article related to <dataset field>: <the text of the target article>. The topic of this article may fall under one of the following clusters: <topic name from *Concept Agent*>. It has the following important neighbors which have citation relationship to this article, from most related to least related: <the texts of Neighbors>. Please consider the information from the target article and its neighbors, and generate a virtual article similar to the given one, with a title of no more than 15 words and an abstract limited to 300 words. Output the newly generated virtual article as a dictionary, with keys "answer" and "explanation".

Figure 7: The detailed instruction design of the generation agent for neighbor summary.

 # Inference Agent
You are an AI assistant specializing in text inference. Please identify the most likely cluster to which the given text belongs.
Given a target article related to <dataset field>: text. Please determine which cluster this <dataset field> most likely belongs to. The optional clusters are: <topic name from *Concept Agent*>. Please comprehensively consider which cluster this article most likely belongs to, only answer the cluster number directly as a dictionary, with keys "answer" and "explanation".

Figure 8: The detailed instruction design of the inference agent for decision filtration.

B Evaluation Metrics

In our work, we adopt four widely recognized metrics: accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI), and F1 score (F1), to comprehensively evaluate the clustering performance of MARK.

- **ACC** quantifies the alignment between predicted cluster labels and ground-truth labels. In the context of unsupervised clustering, the predicted clusters are first matched to the true labels using

the Hungarian algorithm. The value of ACC is then calculated from the resulting confusion matrix $C \in \mathbb{R}^{K \times K}$ as follows:

$$\text{ACC} = \frac{\sum_{i=1}^{K} C_{i,i}}{\sum_{i=1}^{K} \sum_{j=1}^{K} C_{i,j}}, \tag{11}$$

where $C_{i,j}$ denotes the number of samples whose true label is i and predicted label is j.

- **ARI** measures similarity between two assignments. Given a set S and two clustering results $X = (X_1, X_2, ..., X_r)$ and $Y = (Y_1, Y_2, ..., Y_s)$. Let $C = \sum_{ij} \binom{n_{ij}}{2}$, $D = \sum_i \binom{a_i}{2}$, $E = \sum_j \binom{b_j}{2}$, where $n_{ij} = |X_i \cap Y_j|$, $a_i = \sum_j n_{ij}$, and $b_j = \sum_i n_{ij}$. Then we can calculate ARI as follows:

$$\text{ARI} = \frac{C - (D \cdot E)/\binom{n}{2}}{\frac{1}{2}(D + E) - (D \cdot E)/\binom{n}{2}}. \tag{12}$$

- **NMI** calculates consistency between the predicted and true labels. Given two clustering results $X = (X_1, X_2, ..., X_r)$ and $Y = (Y_1, Y_2, ..., Y_s)$,

$$\text{NMI} = \frac{I(X, Y)}{\max\{H(X), H(Y)\}}, \tag{13}$$

where $I(X, Y)$ is the mutual information between X and Y, $H(X)$ and $H(Y)$ are the entropy of X and Y respectively.

- **F1 Score** is computed as the arithmetic mean of the per-class F1 scores and is used to assess clustering performance. It balances precision and recall, providing a comprehensive measure of a model's accuracy.

C Generated concepts

Here are three examples of the ground truth alongside the concepts generated by the concept agent.

Ground Truth	Predicted Concept	Explanation
Reinforcement Learning	Reinforcement Learning and Function Approximation	The texts consistently discuss various aspects and techniques of reinforcement learning, particularly focusing on the challenges and methodologies associated with integrating function approximation methods...
Case Based	Adaptive Case-Based Reasoning Systems	The samples in this cluster revolve around the development and enhancement of case-based reasoning systems, focusing on adaptability, learning from past experiences, and implementing various mechanisms...
Rule Learning	Probabilistic Reasoning and Decision-Making	The cluster focuses on the development and application of probabilistic models...
Probabilistic Methods	Bayesian Methods and Applications	The samples in this cluster focus on various aspects of Bayesian statistical methods, including parameter estimation, model comparison, posterior distributions, and applications of Markov Chain Monte Carlo (MCMC)...
Theory	Adaptive Learning and Inductive Bias	The cluster focuses on various machine learning techniques that emphasize the importance of learning algorithms' adaptability...
Genetic Algorithms	Evolutionary Algorithms in Problem Solving	The cluster highlights the development and application of evolutionary algorithms, particularly genetic algorithms and genetic programming, in solving complex computational problems across various domains....
Neural Networks	Neural Network Theory and Applications	The cluster comprises diverse studies focusing on the theoretical underpinnings, optimization, architectures, and various applications of neural networks...

Table 5: Concepts generated by the concept agent on Cora dataset.

Ground Truth	Predicted Concept	Explanation
Agents	Multi-Agent System Architectures and Methodologies	The cluster is characterized by a focus on various aspects of multi-agent systems including design methodologies, architectures, collaboration mechanisms, and the handling of complex interactions among agents...
Machine Learning	Probabilistic Learning and Decision Systems	The cluster comprises various papers focusing on the integration of probabilistic frameworks and learning algorithms across multiple domains, including artificial intelligence, machine learning...
Information Retrieval	Intelligent Information Retrieval and Web Search Techniques	The cluster encompasses diverse research on advanced methodologies and algorithms for enhancing information retrieval and search capabilities on the web...
Database	Query Optimization and Maintenance in Data Warehousing	The cluster focus on various techniques and methodologies for optimizing query performance and maintaining data integrity within data warehousing systems...
Human Computer Interaction	Ubiquitous Computing and Interactive Interfaces	The cluster focus on enhancing human-computer interaction through ubiquitous computing, mobile devices, and augmented reality...
Artificial Intelligence	Vision-Based Learning and Interaction	The cluster encompasses various approaches and systems involving vision-based techniques for obstacle detection, navigation, gesture recognition, and human-computer interaction...

Table 6: Concepts generated by the concept agent on CiteSeer dataset.

Ground Truth	Predicted Concept	Explanation
Computational linguistics	Multilingual Terminology and Language Processing	The samples revolve around terminological databases and technologies related to language processing and translation across multiple languages...
Databases	Classic Video Game Developments	The cluster comprises detailed accounts of various classic video games...
Operating systems	Live Operating Systems and Unix Variants	This cluster is composed of various texts that revolve around live operating systems...
Computer architecture	Microprocessor Architectures and Technologies	The cluster encompasses a variety of microprocessor architectures, including RISC, CISC, and several proprietary designs...
Computer security	Emerging Cybersecurity Threats and Defenses	The texts are related to various aspects of cybersecurity, including vulnerabilities, attack methodologies...
Internet protocols	Network Protocols and Management	The cluster consists of a wide range of texts detailing various network protocols...
Computer file systems	Evolution of Windows Operating Systems	The cluster contains detailed descriptions and features related to various iterations of Microsoft's Windows operating systems...
Distributed computing architecture	Service-Oriented Architectures and Integration Technologies	The samples collectively emphasize concepts and technologies related to service-oriented architectures (SOA), integration frameworks, and data sharing methodologies...
Web technology	Mobile Operating Systems and Devices	The provided texts explore detailed information on various mobile operating systems (like iOS and Android)...
Programming language topics	Programming Languages and Their Implementations	The provided samples focus on various programming languages, their features, implementation techniques...

Table 7: Concepts generated by the concept agent on WikiCS dataset.

D Computational Cost and Runtime Analysis

MARK introduces a novel paradigm to enhance conventional graph clustering methods by incorporating multi-agent ranking guidance. Specifically, we select two clustering methods, DMoN and MAGI, as the graph clustering engine separately to identify uncertain node texts, which are then processed by an LLM and fine-tuned through multi-agent ranking guidance. Consequently, the computational cost and runtime difference between MARK and its backbone methods stems from the execution of multi-agent module.

Table 8 presents the computational cost and running time of MARK-MAGI when using the API (based on GPT-4o-mini). Following the approach of TAPE (He et al., 2023), we estimate both cost and runtime based on the number of input and output tokens.

Dataset	Cora	CiteSeer	WikiCS	PubMed
Cost($)	0.25	0.18	0.55	0.53
Runtime(min)	3.79	2.79	12.15	13.46

Table 8: Computational cost and runtime analysis.

E Case Study

Factual Accuracy of Generated Texts To evaluate the quality of the generated texts, we use the ROUGE metric (Lin, 2004), where the score ranges from 0 to 1, with a higher score indicating greater factual alignment between the generated and referenced texts. Specifically, we calculate the ROUGE score between the generated texts and the referenced target node text, as well as its similar neighboring texts. For comparison, we also compute the ROUGE score between randomly sampled texts and the same referenced texts. On the Cora, CiteSeer, and PubMed datasets, the ROUGE scores for the LLM-generated texts are 0.39, 0.37, and 0.50, respectively, while the scores for the randomly sampled texts are 0.25, 0.25, and 0.27.

Case Study of Generated Texts Here, we present two case studies to illustrate the quality of the generated texts. Table 9 shows a comparison between generated texts with and without incorporating neighborhood texts. Table 10 demonstrates how label filtering, guided by ranking, helps prevent the introduction of additional noise into the graph clustering engine.

Target Text. **Title:** Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes. **Abstract:** In this paper we present a novel approach to continuous, whole-sentence ***ASL recognition*** that uses phonemes instead of whole signs as the basic units... We subsequently train ***Hidden Markov Models (HMMs)*** to recognize the phonemes... **Ground Truth: Human Computer Interaction.** **Predicted Label: Artificial Intelligence.** **Explanation:** This falls under Vision-Based Learning and Interaction as it involves recognizing and interpreting visual gestures within a computational framework.
Text generated based solely on the target text. **Title:** Enhancing ASL Interpretation Through Phoneme-Based Recognition Frameworks. **Abstract:** In this paper, we introduce an innovative framework for the interpretation of American Sign Language (ASL) by focusing on ***phoneme-level decomposition*** rather than traditional sign recognition...We utilize ***advanced Hidden Markov Models (HMMs)*** to train our system in recognizing these phonemes... **Predicted Label: Artificial Intelligence.** **Explanation:** The use of Hidden Markov Models also indicates an emphasis on computational techniques related to vision-based systems.
Text generated based on both the target and adjacent texts. **Title:** Improving Continuous Gesture Recognition with Gesture Segmentation and Classification. **Abstract:** In this research, we introduce an innovative framework for ***continuous gesture recognition*** that emphasizes the segmentation of gestures into smaller, classifiable units known as gesture segments... **Predicted Label: Human Computer Interaction.** **Explanation:** The article focuses on improving continuous gesture recognition, which is directly related to human-computer interaction.

Table 9: Comparison of generated text quality with and without considering neighborhood texts.

F Generalization Across Backbones and Datasets

To better demonstrate the generalizability and applicability of MARK, we extend it to text-rich graphs in other domains, such as e-commerce networks. Furthermore, we incorporate additional deep graph clustering frameworks, namely Dink-Net (Liu et al., 2023a) and HSAN (Liu et al., 2023b), as backbones to further explore the potential of MARK in enhancing deep graph clustering through this novel paradigm.

Target Text.
Title: Error-Driven Pruning of Treebank Grammars for Base Noun Phrase.
Abstract: Identification Finding simple, non-recursive, base noun phrases is an important subtask for many ***natural language processing*** applications...
Ground Truth: Information Retrieval.
Predicted Label: Machine Learning
Explanation: The article relates to natural language processing (NLP) and discusses algorithms for identifying base noun phrases, which involves machine learning methods.

Generated text by generation agent.
Title: Streamlined Algorithms for Identifying Core Verb Phrases in Textual Data.
Abstract: Identifying core verb phrases within textual data is a foundational task in various natural language processing (NLP) applications...we introduce a streamlined algorithm specifically designed to accommodate the simpler nature of ***identifying core verb phrases***...
Predicted Label: Information Retrieval.
Explanation: The article focuses on natural language processing (NLP) applications, specifically identifying core verb phrases in textual data, which is closely related to intelligent information retrieval.

Table 10: Label filtering based on consistency between target text label and generated text label

Generalization Across Backbones MARK introduces a paradigm to enhance deep graph clustering methods by incorporating multi-agent ranking guidance. In our paper, we use DMoN and MAGI as baselines, but other deep graph clustering frameworks can also serve as backbones. To further explore this, we have additionally incorporated Dink-Net (Liu et al., 2023a) and HSAN (Liu et al., 2023b) from your references as graph clustering engines. Their performance on the Cora dataset is presented below:

METRIC	DINK-NET	PRE-TRAIN	MARK-DINK-NET	IMP ↑	HSAN	PRE-TRAIN	MARK-HSAN	IMP ↑
ACC	0.557±0.029	0.613±0.053	0.669±0.020	11.2%	0.646±0.018	0.671±0.023	0.676±0.046	3.0%
NMI	0.434±0.012	0.516±0.013	0.517±0.006	8.3%	0.511±0.014	0.525±0.010	0.544±0.019	3.3%
F1	0.502±0.040	0.561±0.042	0.590±0.035	8.8%	0.593±0.035	0.621±0.023	0.638±0.044	4.5%

Table 11: Clustering performance of MARK-Dink-Net and MARK-HSAN.

Generalization Across Datasets We evaluate MARK on the e-commerce network Books-History (Yan et al., 2023), which consists of 41,551 nodes, 358,474 edges, and 12 classes. Specifically, the node attributes represent book descriptions, while the edges indicate that two books are co-purchased or co-viewed. The performance of MARK is presented in the following table:

METRIC	MAGI	PRE-TRAIN	MARK-MAGI	IMP ↑
ACC	0.341±0.024	0.362±0.012	0.367±0.015	2.6%
NMI	0.326±0.010	0.338±0.002	0.341±0.006	1.5%
ARI	0.153±0.010	0.165±0.011	0.207±0.014	5.4%

Table 12: Clustering performance of MARK on Books-History datasets.