TEXT-GUIDED GROUP MIXUP WITH CANONICAL MINING FOR IMBALANCED GRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) have achieved remarkable progress in text-attributed graph clustering. However, these approaches assume that different classes are uniformly distributed, which hinders their applicability in real-world, imbalanced scenarios. Towards this end, this paper studies the problem of imbalanced text-attributed graph clustering, and proposes a novel framework named Text-guided Group Mixup with Canonical Mining (TRACI) for the problem. The core of our TRACI lies in generating mixed groups with an emphasis on minority classes, guided by large language models (LLMs). In particular, we first utilize LLMs to produce diverse views for each sample and randomly assign samples into balanced groups with mixed semantics for consistency learning. To further enhance robustness, we employ LLMs to compute correlation scores among samples with respect to the synthesized groups, thereby reinforcing minority-aware group representations. In addition, we encourage canonical correlations between various augmented views of nodes to ensure semantic alignment. Extensive experiments on several benchmark datasets validate the effectiveness of the proposed TRACI, demonstrating clear advantages over state-of-the-art baselines under class-imbalanced conditions. The source code is available at https://anonymous.4open.science/r/TRACI-E087.

1 INTRODUCTION

Graph clustering (Tsitsulin et al., 2023; Ren et al., 2025; Xie et al., 2025), an unsupervised task in graph data mining, aims to assign nodes into distinct clusters that reflect underlying structural and conceptual commonalities. While recent algorithms have made significant progress (Yang et al., 2023; Liu et al., 2023a; 2024b; Kulatilleke et al., 2025), their effectiveness in real-world scenarios remains fundamentally constrained by the inherent class imbalance in graph-structured data (Shi et al., 2020; Huang et al., 2022; Ma et al., 2025). Authentic graph data, such as citation networks (Qin et al., 2025), often exhibit long-tailed distributions (Li & Jia, 2025), where head classes dominate with dense connections, while tail classes are underrepresented, suffering from sparse data and weak connectivity. These naturally occurring imbalances can impair the performance of traditional graph clustering methods, leading to suboptimal results, as such methods are typically developed under the assumption of class balance (Li et al., 2024; Ju et al., 2024; Ma et al., 2025).

To address category imbalance in graph-structured data, researchers have developed three primary paradigms: (i) ***Re-sampling methods*** (Zhang et al., 2023; Gao et al., 2023; Avelino et al., 2024; Carvalho et al., 2025; Nagler et al., 2024) that adjust class selection ratios between classes with varying sample sizes; (ii) ***Re-weighting techniques*** (Li et al., 2025) that modify loss functions based on class frequencies; and (iii) ***Augmentation-based methods*** (Song et al., 2024; Tian et al., 2024; Ding et al., 2025) that transfer knowledge from majority to minority classes using topological or feature semantics. While these approaches have shown promise for attribute graphs with shallow features, they largely neglect the rich contextual semantics in text-attributed graphs (TAGs) (Zhang et al., 2024; He et al., 2025; Hu et al., 2025). This oversight introduces semantic bias, exacerbating challenges such as term frequency-class correlation and topic distribution heterogeneity, which further amplify long-tailed distributions (Chen et al., 2024a). Therefore, effectively leveraging node textual information beyond shallow features, remains a critical challenge in imbalanced text-attributed graph clustering.

The advent of large language models (LLMs) has opened new avenues for text-attributed graph clustering (Chen et al., 2023; Fu et al., 2025). GCLR (Trivedi et al., 2024) leverages the zero-shot capabilities of LLMs to enhance clustering performance through LLM-generated feedback. However, it overlooks the challenge of class imbalance caused by disparities in textual semantics. Although SaVe-TAG (Wang et al., 2024) addresses this issue by synthesizing novel minority-class samples via LLMs for supervised classification, such a strategy is ill-suited for unsupervised clustering and may introduce semantic inconsistencies due to LLM hallucinations (Ji et al., 2023; Verma et al., 2024; Huang et al., 2025). In this work, we seek to harness the zero-shot potential of LLMs for imbalanced graph clustering while striving to preserve semantic consistency in the generated text.

To address the aforementioned challenges, we propose a novel unsupervised LLM-driven framework called Text-guided Group Mixup with Canonical Mining (TRACI) to tackle this problem of imbalance text-attributed graph clustering. The core idea of TRACI is to learn minority-aware group representations that re-balance the contributions of majority and minority classes while preserving semantic integrity as much as possible. We begin by leveraging an LLM to generate augmented texts for nodes in a way that retains their core semantic representations. This is followed by a canonical mining module to align the augmented views in the embedding space. To alleviate imbalance in an unsupervised setting, we randomly assign samples to different groups based on correlation scores provided by the LLM, thereby making better use of textual semantics. For boundary samples between clusters, TRACI utilizes the zero-shot capabilities of LLMs (Ye et al., 2025) to refine the GNN encoder through ranking-based supervision from pseudo-labels generated by the LLM. The effectiveness of TRACI is validated on text-attributed graph datasets and extensive class-imbalanced experiments against state-of-the-art baselines.

In conclusion, our main contributions in this work are summarized as follows:

- ***New Perspective.*** To the best of our knowledge, we are the first to investigate the problem of imbalanced text-attributed graph clustering enhanced by LLMs in an unsupervised manner.
- ***Novel Methodology.*** We propose TRACI, a framework that first leverages LLMs to generate text-level augmented views while preserving semantic integrity, followed by the assignment of samples into text-guided mixed groups with canonical correlation alignment.
- ***Comprehensive Experiments.*** Extensive experiments on multiple benchmark datasets under imbalanced conditions demonstrate that TRACI consistently outperforms state-of-the-art baselines.

2 Related Work

Text-attributed Graph Clustering. The classic paradigm for text-attributed graph clustering (Tsitsulin et al., 2023; Yan et al., 2023; Zhou et al., 2025; Zhu et al., 2025; Yu et al., 2025) typically involves extracting textual embeddings from shallow, context-free features (Mikolov et al., 2013; Wu et al., 2025; Zhang et al., 2025), which are then integrated with the graph's topological structure via graph neural networks (GNNs) (Liu et al., 2024b; Bhowmick et al., 2024; Wang et al., 2025b). More recently, LLM-based approaches for text-attributed graphs have emerged and can be broadly categorized into three paradigms (Chen et al., 2024b): ***LLM-as-Predictor***, ***LLM-as-Enhancer*** and ***LLM-as-Aligner***. Specifically, the ***LLM-as-Predictor*** paradigm feeds structure-aware textual inputs directly into LLMs to predict node labels (Qiao et al., 2025; Chen et al., 2023). In contrast, ***LLM-as-Enhancer*** leverages LLMs to enrich text representations, either by extracting contextualized embeddings (fine-tuned (Mavromatis et al., 2023) or frozen (Qiao et al., 2025)) or by generating auxiliary semantic signals (explanations or augmentations). More importantly, ***LLM-as-Aligner*** aims to align the outputs from GNNs and LLMs iteratively or in parallel (Liu et al., 2025). This paradigm simultaneously leverage the structural aggregation capabilities of GNNs and the semantic extraction abilities of LLMs. These methods are typically implemented through prediction alignment (Zhao et al., 2021) or embedding alignment (Hu et al., 2025). While these paradigms have been actively explored in supervised or self-supervised tasks such as node classification and link prediction, their potential in unsupervised settings like graph clustering remains largely underexplored.

Long-tailed Graph Learning. Class imbalance (Carvalho et al., 2025; Ma et al., 2025) in graph data presents a significant obstacle to the effective deployment of Graph Neural Networks (GNNs). Existing efforts to address long-tailed graph learning can be broadly categorized into three main paradigms: re-sampling methods (Carvalho et al., 2025), re-weighting techniques (He, 2024a), and augmentation-based strategies (Khan et al., 2024). On the re-sampling side, GraphSMOTE (Zhao

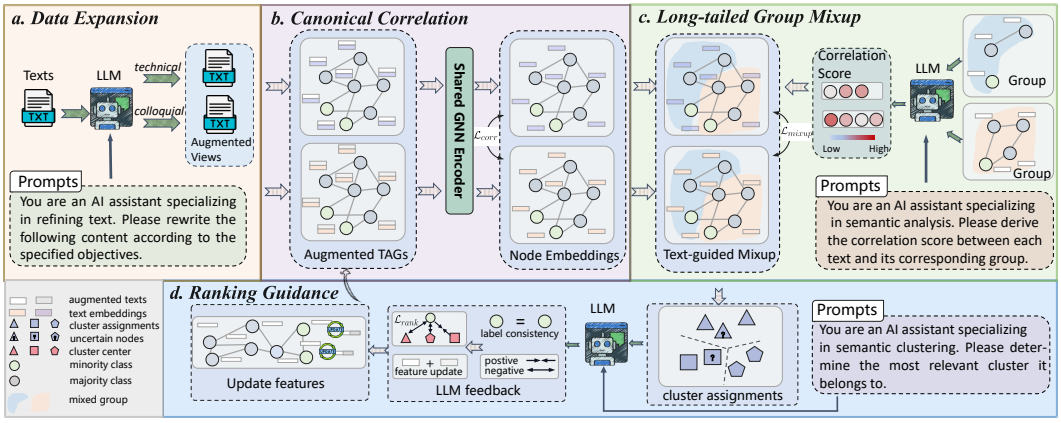


Figure 1: The framework of TRACI. TRACI consists of four key modules: (a) data expansion, (b) canonical correlation, (c) long-tailed group mixup and (d) fine-tuning with ranking guidance.

et al., 2021) and ImGAGN (Qu et al., 2021) generate synthetic samples for minority classes through oversampling and adversarial generation, respectively. On the re-weighting side, ReNode (Chen et al., 2021) adaptively adjusts node weights by quantifying influence shifts near class boundaries. In the augmentation line, RAHNet (Mao et al., 2023) enhances minority class representation through a retrieval-augmented mechanism by incorporating external knowledge. Despite these advancements, most existing methods overlook the rich contextual semantics embedded in node texts (Ghosh et al., 2024; Wang et al., 2025a). To address this, we propose TRACI, a novel framework that leverages textual semantics to generate balanced, minority-aware group representations.

3 Methodology

Problem Formulation. A TAG can be represented as $\mathcal{G} = \{\mathcal{V}, \mathbf{A}, \mathcal{D}, \mathbf{X}\}$, where $\mathcal{V}$ denotes a set of N nodes, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix, $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$ represents the text attributes associated with the nodes, and $\mathbf{X} \in \mathbb{R}^{N\times F}$ is the text embedding matrix encoded by the frozen language model Sentence-BERT (Reimers & Gurevych, 2019). In this work, we aim to partition the nodes in $\mathcal{G}$ into K disjoint clusters $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$ under a long-tailed distribution scenario, where the number of samples in each cluster are highly imbalanced. Specifically, we quantify this skewed distribution using the imbalance ratio, defined as $\rho = \frac{n_{\max}}{n_{\min}}$ (Ma et al., 2025), where $n_{\max} = \max\{|\mathcal{C}k|\}_{k=1}^{K}$ and $n_{\min} = \min\{|\mathcal{C}k|\}_{k=1}^{K}$, with $|\cdot|$ denoting the sample size of a set.

3.1 Framework Overview

The proposed framework of TRACI is illustrated in Figure 1. The pipeline consists of four key modules: data expansion, canonical correlation, long-tailed mixup and fine-tuning with ranking guidance. Specifically, we leverage an LLM to generate augmented textual views for each node, which are subsequently encoded by the language model (LM) Sentence-BERT to produce input embeddings for the GNN encoder. Subsequently, node-level embeddings are derived from the augmented TAGs using a shared GNN encoder and are aligned in the embedding space via canonical correlation. And the correlation scores computed by the LLM are employed to guide the synthesis of mixed group representations, placing greater emphasis on the minority class and thereby reinforcing minority-aware group representations. Finally, feedback responses from the LLM are utilized to fine-tune TRACI with ranking guidance, enhancing the assignment of boundary nodes.

3.2 Data Expansion with Large Language Models

In this framework, we follow the standard augmentation-contrastive paradigm, as exemplified by SimCLR (Chen et al., 2020). Previous text augmentation methods (Yan et al., 2021; Gao et al., 2021) typically apply token-level transformations, such as shuffling, dropout, or cutoff to the original text. However, these techniques can inadvertently alter the core semantics of sentences, potentially compromising the performance of downstream tasks. To address this, we forgo such destructive augmentations and instead adopt a more semantically-preserving yet stylistically diverse

strategy powered by LLMs. Specifically, the original text is input into an LLM, which is prompted to generate two stylistically distinct versions: a technical version $\mathcal{D}^{(1)} = \{\mathcal{D}_1^{(1)}, \mathcal{D}_2^{(1)}, \cdots, \mathcal{D}_N^{(1)}\}$ and a colloquial version $\mathcal{D}^{(2)} = \{\mathcal{D}_1^{(2)}, \mathcal{D}_2^{(2)}, \cdots, \mathcal{D}_N^{(2)}\}$. This approach maintains the original semantic content while introducing diverse linguistic expressions, enabling much more abundant semantic representation learning. Finally, we construct two augmented views of the original TAG $\mathcal{G}$: $\mathcal{G}^{(1)} = \{\mathcal{V}, \mathbf{A}, \mathcal{D}^{(1)}, \mathbf{X}^{(1)}\}$ and $\mathcal{G}^{(2)} = \{\mathcal{V}, \mathbf{A}, \mathcal{D}^{(2)}, \mathbf{X}^{(2)}\}$, where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are embeddings encoded by the augmentations, serving as inputs for following stages.

3.3 Canonical Correlation Maximization for Semantics Alignment

To encourage node-level semantic alignment, we adopt a maximum correlation objective (Andrew et al., 2013) to ensure consistency between augmented views. As discussed in Section 3.2, the two augmented views are processed through a shared GNN encoder to promote entity alignment across different stylistic variations. This alignment mechanism ensures that semantically similar entities are mapped closely in the embedding space, regardless of surface-level linguistic differences. Given two augmented views $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, we utilize the shared GNN encoder to extract the corresponding node embeddings $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)} \in \mathbb{R}^{N\times D}$. We further compute the centered node embeddings $\bar{\mathbf{Z}}^{(1)}$ and $\bar{\mathbf{Z}}^{(2)}$ as $\bar{\mathbf{Z}}^{(1)} = \mathbf{Z}^{(1)} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{Z}^{(1)}$ and $\bar{\mathbf{Z}}^{(2)} = \mathbf{Z}^{(2)} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{Z}^{(2)}$, respectively. Here, $\mathbf{1}_N$ is a vector length of N with all elements equal to 1. The cross-covariance matrix between the two views is calcultaed as $\mathbf{C}_{1,2} = (\bar{\mathbf{Z}}^{(1)})^\top\bar{\mathbf{Z}}^{(2)}/(N-1) \in \mathbb{R}^{D\times D}$, and the self-covariance matrix for each view is given by $\mathbf{C}_{1,1} = (\bar{\mathbf{Z}}^{(1)})^\top\bar{\mathbf{Z}}^{(1)}/(N-1)$. Finally, the canonical correlation loss between the augmented views is applied to align them in the embedding space, which is defined as:

$$\mathcal{L}_{\text{corr}}(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}) = -\text{Trace}(\mathbf{C}_{1,1}^{-1/2}\mathbf{C}_{1,2}\mathbf{C}_{2,2}^{-1}\mathbf{C}_{2,1}\mathbf{C}_{2,2}^{-1/2}). \tag{1}$$

3.4 Long-tailed Group Mixup with Textual Guidance

To alleviate the imbalance issue, we design a mixup strategy guided by textual semantics to learn minority-aware group representations. Specifically, samples from the augmented view $\mathcal{G}^{(1)}$ are randomly partitioned into M groups $\mathbb{G} = \{\mathbb{G}_1, \mathbb{G}_2, \cdots, \mathbb{G}_M\}$, where $\mathbb{G}_m$ represents the index set of samples in the m-th group. The corresponding samples from the other augmented view $\mathcal{G}^{(2)}$ are partitioned with the same index sets. Subsequently, each group of texts is fed into an LLM using the prompt $\mathcal{P}_{\text{mix}}$, which outputs a contribution score $b_{mn}^{(i)}$ and a confidence score $c_{mn}^{(i)}$ for the n-th text in the m-th group from the i-th augmented view. The contribution score $b_{mn}^{(i)}$ reflects the semantic relevance and conceptual coherence of the text within the group, while the confidence score $c_{mn}^{(i)}$ estimates the credibility of the corresponding contribution. Based on these scores, we finally derive a correlation-based weight matrix $\mathbf{S}^{(i)} \in \mathbb{R}^{M\times N}$ for each view ($i = 1, 2$) to improve awareness of minority in the mixed groups, whose element-wise definitions are stated as follows:

$$s_{mn}^{(i)} = \frac{e^{(1-b_{mn}^{(i)})\cdot c_{mn}^{(i)}}}{\sum\limits_{n\in\mathbb{G}_m} e^{(1-b_{mn}^{(i)})\cdot c_{mn}^{(i)}}}\mathbf{1}_{n\in\mathbb{G}_m}. \tag{2}$$

Here, $\mathbf{1}_{n\in\mathbb{G}_m}$ is an indicator function that equals 1 if and only if $n \in \mathbb{G}_m$. Subsequently, we construct group-level synthetic embeddings as weighted combinations of the sample representations,

$$\mathbf{h}_m^{(i)} = \sum_{n\in\mathbb{G}_m} s_{mn}^{(i)}\mathbf{z}_n^{(i)} / \| \sum_{n\in\mathbb{G}_m} s_{mn}^{(i)}\mathbf{z}_n^{(i)}\|_2, \tag{3}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm and $\mathbf{z}_n^{(i)}$ is the representation of the n-th sample in the i-th view. Following the aforementioned steps, we obtain two group-level augmented representations $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)} \in \mathbb{R}^{M\times D}$ for contrastive learning. In the contrastive learning setup, corresponding group representations from the two views form positive pairs, while all other combinations are considered negative pairs. The imbalance-aware contrastive loss is then defined as:

$$\mathcal{L}_{\text{mixup}}(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}) = -\frac{1}{M}\sum_{m=1}^{M}\log\frac{e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_m^{(2)})/\tau_1}}{e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_m^{(2)})/\tau_1} + \sum\limits_{m'\neq m} e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_{m'}^{(1)})/\tau_1} + \sum\limits_{m'\neq m} e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_{m'}^{(2)})/\tau_1}}, \tag{4}$$

where τ_1 is a temperature hyperparameter and θ denotes cosine similarity between two vectors.

3.5 Fine-tuning with Ranking Guidance

To further leverage the capabilities of LLMs to enhance the GNN encoder, we propose a two-stage optimization strategy for TRACI. In the first stage, we align two augmented views in the embedding space while applying text-guided mixup to learn minority-aware group representations. In the second stage, we utilize LLMs to annotate the boundaries of imbalanced nodes, particularly between clusters that are difficult for the GNN to distinguish. The feedback obtained from the LLM is then used as ranking guidance to fine-tune the encoder for better representation learning. To jointly enforce consistency between both augmented views and address the problem of class imbalance, we combine Equation 1 and Equation 4 as follows, where α denotes the trade-off hyperparameter:

$$\mathcal{L}_{\text{warm}} = \alpha\mathcal{L}_{\text{corr}} + (1-\alpha)\mathcal{L}_{\text{mixup}}. \tag{5}$$

Boundary Nodes for Querying LLMs. After completing the first-stage training, we obtain the embeddings $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ of the two augmented views using the frozen encoder f_Θ, expressed as,

$$\mathbf{Z}^{(1)} = f_\Theta(\mathbf{A}, \mathbf{X}^{(1)}) \text{ and } \mathbf{Z}^{(2)} = f_\Theta(\mathbf{A}, \mathbf{X}^{(2)}), \tag{6}$$

where Θ is the learned parameter. To mitigate uniform clustering under imbalanced settings, we apply smooth k-means clustering (He, 2024b) to these embeddings, yielding the predicted label sets $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$, respectively. We then identify a set of challenging nodes $\mathcal{S} = \{i \in \mathcal{V} \mid \mathcal{C}_i^{(1)} \neq \mathcal{C}_i^{(2)}\}$, which are subsequently used as queries to the LLM to obtain additional guidance for better learning.

Concept Induction for Each Cluster. Since the semantic meanings of clusters are initially unknown, we select the top-k nearest nodes to each cluster centroid and construct a representative sample set to query the LLM for inducing core concepts of each cluster via the prompt $\mathcal{P}_{\text{indu}}$, expressed as follows:

$$\mathcal{M} = \mathcal{P}_{\text{indu}}(\text{top-}k \text{ texts for each cluster}). \tag{7}$$

Decision Filtering with Ranking Guidance. Subsequently, we feed the derived concept set $\mathcal{M}$ together with the textual content of the challenging node set $\mathcal{S}$ from both augmented views into the LLM using the prompt $\mathcal{P}$pred, yielding the predicted label sets $\mathcal{C}_{\text{chal}}^{(1)}$ and $\mathcal{C}_{\text{chal}}^{(2)}$, as follows:

$$\mathcal{C}_{\text{chal}}^{(1)} = \mathcal{P}_{\text{pred}}(\mathcal{M}, \mathcal{D}_S^{(1)}),\ \mathcal{C}_{\text{chal}}^{(2)} = \mathcal{P}_{\text{pred}}(\mathcal{M}, \mathcal{D}_S^{(2)}). \tag{8}$$

Given that LLM predictions may inevitably contain errors, particularly for challenging nodes, we introduce a dual-view consensus mechanism to filter out noisy labels and mitigate additional biases introduced by the LLM. The detailed filtering process is described as follows:

$$\mathcal{C}_{\text{LLM}} = \{i \in \mathcal{S} \mid \mathcal{C}_{\text{chal}}^{(1)}(i) = \mathcal{C}_{\text{chal}}^{(2)}(i)\}. \tag{9}$$

More importantly, we further apply a ranking-based contrastive loss to incorporate the LLM's feedback and fine-tune the GNN model obtained in the first stage. Specifically,

$$\mathcal{L}_{\text{rank}} = -\sum_{i \in \mathcal{C}_{\text{LLM}}} \log \frac{\exp\left(\theta\left(\mathbf{z}_i, \boldsymbol{\mu}_{\mathcal{C}_{\text{chal}}^{(1)}(i)}\right)/\tau_2\right)}{\sum_{k=1}^{K} \exp\left(\theta\left(\mathbf{z}_i, \boldsymbol{\mu}_k\right)/\tau_2\right)}. \tag{10}$$

Here, τ_2 denotes the temperature hyper-parameter, and θ represents the cosine similarity between the two vectors. $\boldsymbol{\mu}_k$ refers to the cluster centroid of the kth cluster. We then employ the objective $\mathcal{L}_{\text{fine}}$ to fine-tune the final imbalanced graph clustering model:

$$\mathcal{L}_{\text{fine}} = \alpha\mathcal{L}_{\text{corr}} + (1-\alpha)(\beta\mathcal{L}_{\text{mixup}} + (1-\beta)\mathcal{L}_{\text{rank}}). \tag{11}$$

Consequently, we first warm up a base model capable of handling imbalanced representation learning across two augmented views. Building upon this foundation, we identify challenging nodes and leverage LLMs to further enhance the model's ability to address class imbalance. This progressive paradigm of TRACI results in more reliable predictions for imbalanced graph clustering.

3.6 Theoretical Analysis

Theoretically, our theorem establishes a tighter generalization error bound with the re-balanced mixup strategy guided by the LLM, compared to the standard contrastive loss.

Theorem 3.1. *Let $\mathcal{X}$ be the input space and $\mathcal{Z} \subset \mathbb{R}^D$ denotes the latent space. Suppose the following conditions holds: (1) **Imbalanced Distribution**: $\exists\, \rho \gg 1$ s.t. $n_{max}/n_{min} = \rho$ for cluster sizes; (2) **Group Mixup**: Samples partitioned into M groups $\{\mathbb{G}_m\}_{m=1}^{M}$ such that $||\mathbb{G}_m|| = n/M$. (3) **LLM-guided Weights**: The weight matrix $\boldsymbol{S} \in \mathbb{R}^{M\times N}$ in Eq. (2) satisfying $\sum_n s_{m,n} = 1$. Then, the generalization error bound for the mixup loss $\mathcal{L}_{mixup}$ is tighter than that of the classic contrastive loss $\mathcal{L}_{cl}$. Specifically, with probability at least $1-\delta$, for any encoder f_Θ, the following holds:*

$$\mathcal{E}(\mathcal{L}_{mixup}) - \mathcal{E}^*(\mathcal{L}_{mixup}) \leq C(\sqrt{\frac{\log M}{M}} + \sqrt{\frac{\log(1/\delta)}{N}}), \tag{12}$$

$$\mathcal{E}(\mathcal{L}_{cl}) - \mathcal{E}^*(\mathcal{L}_{cl}) \leq C(\sqrt{\frac{\log N}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}), \tag{13}$$

where $C > 0$ is a constant, $M \ll N$, $\mathcal{E}$ denotes the generalization error, and $\mathcal{E}^*$ is the Bayes optimal error. $\mathcal{L}_{cl}$ is the classic contrastive loss in the embedding space.

The proof sketch consists of three key steps: (1) reducing the effective sample complexity through group-wise mixup, (2) bounding the empirical Rademacher complexity for both of $\mathcal{L}_{mixup}$ and $\mathcal{L}_{cl}$, and (3) incorporating statistical learning theory to establish generalization bounds. A detailed proof of Theorem 3.1 can be found in the Appendix, which provides theoretical support for our TRACI.

4 Experiments

4.1 Experimental Setup

Imbalanced Datasets. In this work, we evaluate the performance of TRACI under class-imbalanced scenarios using four widely adopted TAG datasets: Cora (McCallum et al., 2000), CiteSeer (Giles et al., 1998), WikiCS (Mernyei & Cangea, 2020), and PubMed (Sen et al., 2008). To simulate real-world imbalance, we construct long-tailed variants of these datasets (Park et al., 2021) with varying imbalance ratios. Specifically, the sample size for the k-th class is given by $n_k = n_{max} \cdot \rho^{-\frac{k-1}{K-1}}$ (Ma et al., 2025), with K denoting the total number of classes. To preserve the topological structure of the original graph, nodes with higher connectivity are preferentially retained during the sampling process. Detailed statistics corresponding to different imbalance ratios ($\rho = 10, 20, 50, 100$) are illustrated in the following Figure 3. Additional details about the datasets, including their topological structures, are provided comprehensively in Table A.

Baseline Methods. We compare TRACI with several state-of-the-art deep clustering methods: DMoN (Tsitsulin et al., 2023), Dink-Net (Liu et al., 2023a), HSAN (Liu et al., 2023b), S^3GC (Devvrit et al., 2022), DGCLUSTER (Bhowmick et al., 2024), MAGI (Liu et al., 2024b) and IsoSEL (Sun et al., 2025), under class-imbalanced settings. In particular, we extend our evaluation by comparing TRACI with established graph learning frameworks such as GraphSMOTE (Zhao et al., 2021), GraphENS (Park et al., 2021), and BAT (Liu et al., 2024c), thereby providing a more comprehensive validation of its effectiveness. Following prior work, we adopt accuracy (ACC), normalized mutual information (NMI), and F1 score as evaluation metrics for comparison.

Implementation Details. The implementation of our proposed method, TRACI, is based on the PyTorch library, and both the datasets and source codes are publicly available. To encode textual information, we utilize Sentence-BERT (Reimers & Gurevych, 2019) to extract text embeddings. For representation learning, we employ a Graph Convolutional Network (GCN) as the backbone encoder for GNN to aggregate neighborhood semantics. For interactions with large language models (LLMs), we use ChatGPT (gpt-4o-mini) (Hurst et al., 2024) to provide guidance and feedback; the detailed prompt design is described in Table E. For fair comparison, we report the performance as the mean and standard deviation over five runs. In particular, more detailed information, including hyperparameter settings and the training strategy, is thoroughly provided in Table H.

Table 1: Clustering performance of TRACI compared to baseline methods under varying imbalance ratios ($\rho = 10$ and 20). Boldfaced scores indicate the best results, while underlined scores denote the second-best results. "OOM" means out-of-memory.

ρ	Dataset	Metric	DMoN	Dink-Net	HSAN	S^{3}GC	DGCluster	MAGI	IsoSEL	TRACI
10	Cora	ACC	60.05$_{\pm 3.08}$	61.67$_{\pm 1.34}$	63.41$_{\pm 2.17}$	56.68$_{\pm 2.64}$	56.86$_{\pm 4.19}$	65.34$_{\pm 0.31}$	58.66$_{\pm 6.07}$	**73.48**$_{\pm 2.21}$
		NMI	49.61$_{\pm 0.72}$	51.88$_{\pm 0.95}$	49.26$_{\pm 1.05}$	43.54$_{\pm 0.51}$	52.80$_{\pm 0.94}$	52.90$_{\pm 0.30}$	50.32$_{\pm 1.70}$	**55.60**$_{\pm 0.10}$
		F1	51.02$_{\pm 2.90}$	55.57$_{\pm 4.24}$	57.88$_{\pm 1.55}$	51.26$_{\pm 2.80}$	49.90$_{\pm 5.59}$	60.04$_{\pm 0.26}$	43.70$_{\pm 7.19}$	**67.03**$_{\pm 3.79}$
	CiteSeer	ACC	56.97$_{\pm 8.96}$	55.88$_{\pm 4.77}$	55.28$_{\pm 2.29}$	56.63$_{\pm 2.17}$	47.48$_{\pm 3.37}$	63.13$_{\pm 0.34}$	54.73$_{\pm 12.43}$	**67.15**$_{\pm 0.17}$
		NMI	39.64$_{\pm 3.42}$	37.63$_{\pm 0.56}$	37.31$_{\pm 0.58}$	40.10$_{\pm 0.56}$	36.62$_{\pm 0.83}$	40.75$_{\pm 0.40}$	37.64$_{\pm 1.15}$	**41.72**$_{\pm 0.10}$
		F1	49.17$_{\pm 10.68}$	46.28$_{\pm 4.93}$	50.56$_{\pm 3.97}$	51.59$_{\pm 1.19}$	32.09$_{\pm 4.11}$	56.44$_{\pm 0.38}$	37.39$_{\pm 10.31}$	**60.44**$_{\pm 0.14}$
	WikiCS	ACC	34.73$_{\pm 1.08}$	54.89$_{\pm 3.57}$	56.34$_{\pm 3.55}$	44.64$_{\pm 2.21}$	55.40$_{\pm 2.36}$	58.55$_{\pm 0.67}$		**63.33**$_{\pm 1.55}$
		NMI	24.14$_{\pm 1.41}$	44.30$_{\pm 1.65}$	48.28$_{\pm 1.06}$	38.37$_{\pm 0.47}$	43.93$_{\pm 1.57}$	**49.58**$_{\pm 0.15}$	OOM	49.23$_{\pm 1.25}$
		F1	27.30$_{\pm 2.12}$	46.60$_{\pm 6.97}$	50.76$_{\pm 3.88}$	36.48$_{\pm 2.58}$	46.59$_{\pm 5.44}$	47.47$_{\pm 0.56}$		**53.93**$_{\pm 1.60}$
	PubMed	ACC	55.62$_{\pm 8.84}$	49.11$_{\pm 3.90}$	49.26$_{\pm 0.03}$	54.07$_{\pm 0.75}$	58.31$_{\pm 2.96}$	41.79$_{\pm 0.15}$	58.24$_{\pm 5.59}$	**61.42**$_{\pm 5.30}$
		NMI	8.17$_{\pm 1.46}$	11.01$_{\pm 2.02}$	7.95$_{\pm 0.03}$	15.16$_{\pm 0.99}$	10.98$_{\pm 1.00}$	8.08$_{\pm 0.03}$	12.69$_{\pm 3.39}$	**20.90**$_{\pm 1.61}$
		F1	42.91$_{\pm 3.81}$	41.88$_{\pm 5.26}$	42.47$_{\pm 0.03}$	47.58$_{\pm 0.90}$	47.17$_{\pm 4.01}$	29.86$_{\pm 0.11}$	42.38$_{\pm 5.18}$	**52.45**$_{\pm 4.98}$
20	Cora	ACC	60.17$_{\pm 5.01}$	58.20$_{\pm 2.39}$	53.38$_{\pm 1.81}$	51.30$_{\pm 0.88}$	53.18$_{\pm 2.08}$	57.86$_{\pm 0.77}$	60.28$_{\pm 9.95}$	**68.89**$_{\pm 6.08}$
		NMI	50.52$_{\pm 1.01}$	47.81$_{\pm 1.16}$	46.99$_{\pm 0.80}$	43.52$_{\pm 1.45}$	48.71$_{\pm 0.27}$	49.33$_{\pm 0.33}$	48.88$_{\pm 1.41}$	**52.56**$_{\pm 3.17}$
		F1	46.05$_{\pm 3.91}$	46.05$_{\pm 3.36}$	47.22$_{\pm 2.03}$	43.10$_{\pm 1.17}$	42.97$_{\pm 1.23}$	51.68$_{\pm 0.49}$	41.68$_{\pm 11.24}$	**57.65**$_{\pm 4.92}$
	CiteSeer	ACC	59.07$_{\pm 4.98}$	50.87$_{\pm 3.30}$	49.12$_{\pm 0.67}$	53.58$_{\pm 1.01}$	47.93$_{\pm 3.01}$	58.92$_{\pm 0.91}$	50.41$_{\pm 8.28}$	**67.15**$_{\pm 6.50}$
		NMI	39.67$_{\pm 1.59}$	37.56$_{\pm 1.07}$	31.96$_{\pm 1.03}$	40.67$_{\pm 1.31}$	36.22$_{\pm 1.11}$	37.80$_{\pm 0.28}$	31.50$_{\pm 4.03}$	**42.59**$_{\pm 2.43}$
		F1	46.46$_{\pm 5.56}$	39.45$_{\pm 4.80}$	43.11$_{\pm 1.88}$	44.31$_{\pm 1.54}$	27.58$_{\pm 5.13}$	50.33$_{\pm 0.64}$	31.02$_{\pm 2.96}$	**55.67**$_{\pm 7.68}$
	WikiCS	ACC	37.29$_{\pm 0.19}$	58.73$_{\pm 5.74}$	55.61$_{\pm 5.75}$	45.41$_{\pm 0.28}$	59.28$_{\pm 0.78}$	60.01$_{\pm 0.07}$		**60.57**$_{\pm 3.37}$
		NMI	28.22$_{\pm 0.85}$	48.48$_{\pm 2.16}$	47.40$_{\pm 2.02}$	40.00$_{\pm 0.17}$	46.71$_{\pm 0.71}$	**49.75**$_{\pm 0.09}$	OOM	47.40$_{\pm 0.98}$
		F1	27.61$_{\pm 2.04}$	**48.94**$_{\pm 6.62}$	45.41$_{\pm 4.03}$	35.76$_{\pm 0.19}$	44.69$_{\pm 2.45}$	45.88$_{\pm 0.07}$		48.59$_{\pm 1.07}$
	PubMed	ACC	56.01$_{\pm 11.16}$	47.77$_{\pm 3.25}$	44.35$_{\pm 3.33}$	50.65$_{\pm 0.49}$	57.34$_{\pm 3.23}$	45.60$_{\pm 0.16}$	59.31$_{\pm 5.70}$	**64.72**$_{\pm 5.28}$
		NMI	7.34$_{\pm 1.61}$	7.49$_{\pm 0.29}$	4.92$_{\pm 0.28}$	12.12$_{\pm 0.41}$	8.24$_{\pm 0.45}$	6.79$_{\pm 0.09}$	8.03$_{\pm 0.87}$	**13.76**$_{\pm 1.35}$
		F1	40.59$_{\pm 5.52}$	34.12$_{\pm 5.48}$	32.70$_{\pm 4.19}$	40.58$_{\pm 0.52}$	38.62$_{\pm 5.11}$	29.29$_{\pm 0.04}$	36.08$_{\pm 3.80}$	**49.43**$_{\pm 3.11}$

4.2 Performance Comparison

Performance of graph clustering under imbalance. To comprehensively assess the performance of TRACI, we evaluate it against seven state-of-the-art baseline methods under imbalance ratios of 10 and 20. As shown in Table 1, TRACI consistently outperforms other state-of-the-art methods on the Cora, CiteSeer, and PubMed datasets across all three metrics (ACC, NMI, and F1 score) under both imbalance ratios ($\rho = 10$ and $\rho = 20$). Specifically, on the Cora dataset with $\rho = 10$, TRACI achieves improvements of 8.14%, 2.70%, and 6.99% in ACC, NMI, and F1 score, respectively, compared to the second-best baseline MAGI. On the WikiCS dataset, TRACI achieves the highest accuracy scores under both $\rho = 10$ and $\rho = 20$. Although the NMI and F1 scores are not the highest in all cases, they still rank among the top-performing methods. Interestingly, some baseline methods demonstrate high ACC but relatively low NMI (e.g., DGCluster on PubMed under $\rho = 10$). This discrepancy may be attributed to misclustered samples: either large clusters are fragmented into smaller ones, or small clusters are merged into a larger one, which distorts the clustering quality despite a seemingly good accuracy. In contrast, TRACI produces clustering results that more faithfully reflect the underlying long-tailed distribution, making it particularly well-suited for real-world class-imbalanced scenarios. Overall, these findings strongly affirm the effectiveness and robustness of TRACI in handling imbalanced data. Furthermore, we evaluate TRACI under even more severe imbalance conditions (ρ = 50 and 100), and the corresponding results for more imbalanced scenarios are provided comprehensively in Table 9.

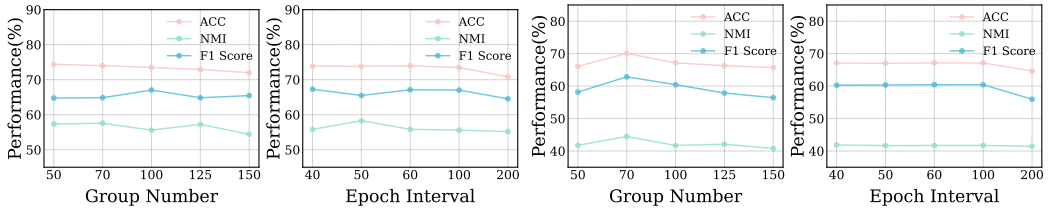
Performance of graph learning under imbalance. We evaluate the representations learned by TRACI against other imbalanced graph learning methods on our established datasets. Specifically, we use the learned representations with an additional classifier for node classification. As shown in Table 2, although TRACI performs relatively poorly on Cora, it consistently achieves superior performance on CiteSeer, WikiCS, and PubMed, demonstrating

Table 2: Graph learning Performance of TRACI in comparison with baselines.

Dataset	Metric	GraphSMOTE	GraphENS	BAT	TRACI
Cora	ACC	83.47$_{\pm 1.28}$	84.00$_{\pm 1.30}$	84.84$_{\pm 0.95}$	**84.92**$_{\pm 0.97}$
	NMI	65.31$_{\pm 2.15}$	65.71$_{\pm 3.30}$	**67.31**$_{\pm 1.39}$	65.90$_{\pm 1.96}$
	F1	77.75$_{\pm 1.89}$	79.56$_{\pm 2.50}$	**80.76**$_{\pm 1.30}$	80.47$_{\pm 1.53}$
CiteSeer	ACC	73.31$_{\pm 1.03}$	73.14$_{\pm 2.56}$	75.54$_{\pm 0.56}$	**78.91**$_{\pm 2.44}$
	NMI	46.07$_{\pm 1.70}$	46.11$_{\pm 2.85}$	48.79$_{\pm 1.17}$	**52.41**$_{\pm 5.12}$
	F1	64.86$_{\pm 1.40}$	64.25$_{\pm 2.32}$	66.35$_{\pm 1.23}$	**66.88**$_{\pm 4.37}$
WikiCS	ACC	81.32$_{\pm 1.54}$	80.28$_{\pm 0.97}$	81.56$_{\pm 0.54}$	**82.01**$_{\pm 0.93}$
	NMI	62.83$_{\pm 2.21}$	61.52$_{\pm 1.49}$	63.12$_{\pm 0.76}$	**63.35**$_{\pm 1.36}$
	F1	78.62$_{\pm 1.76}$	77.52$_{\pm 0.56}$	79.07$_{\pm 0.53}$	**79.29**$_{\pm 0.70}$
PubMed	ACC	86.40$_{\pm 1.10}$	84.72$_{\pm 0.39}$	87.14$_{\pm 0.37}$	**89.23**$_{\pm 0.91}$
	NMI	45.02$_{\pm 1.92}$	42.47$_{\pm 0.96}$	45.64$_{\pm 0.84}$	**50.23**$_{\pm 3.00}$
	F1	76.82$_{\pm 1.38}$	75.06$_{\pm 0.45}$	77.14$_{\pm 0.57}$	**79.23**$_{\pm 0.96}$

Table 3: Comparison of TRACI with various model variants in terms of ACC and F1 scores under imbalanced settings ($\rho = 10$ and 20). The best results are highlighted in bold.

Imbalance	Variants	Cora		CiteSeer		WikiCS		PubMed	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1
$\rho = 10$	**w/o LLM Expansion**	69.45$_{\pm 2.15}$	61.83$_{\pm 1.77}$	64.96$_{\pm 3.69}$	55.50$_{\pm 7.08}$	51.16$_{\pm 2.96}$	46.49$_{\pm 1.32}$	57.37$_{\pm 0.15}$	47.83$_{\pm 0.07}$
	w/o LLM Mixup	70.19$_{\pm 2.60}$	61.29$_{\pm 3.51}$	64.31$_{\pm 6.59}$	54.81$_{\pm 8.35}$	61.33$_{\pm 0.34}$	52.04$_{\pm 0.65}$	59.12$_{\pm 6.12}$	50.16$_{\pm 5.87}$
	w/o Smooth	68.60$_{\pm 4.17}$	63.11$_{\pm 3.69}$	60.06$_{\pm 5.14}$	51.36$_{\pm 6.54}$	58.23$_{\pm 3.04}$	53.13$_{\pm 1.67}$	59.72$_{\pm 5.56}$	51.13$_{\pm 4.90}$
	TRACI	**73.48**$_{\pm 2.21}$	**67.03**$_{\pm 3.79}$	**67.15**$_{\pm 0.17}$	**60.44**$_{\pm 0.14}$	**63.33**$_{\pm 1.55}$	**53.93**$_{\pm 1.60}$	**61.42**$_{\pm 5.30}$	**52.45**$_{\pm 4.98}$
$\rho = 20$	**w/o LLM Expansion**	66.12$_{\pm 3.63}$	57.35$_{\pm 3.54}$	61.63$_{\pm 7.44}$	51.78$_{\pm 7.78}$	50.92$_{\pm 4.35}$	43.57$_{\pm 2.53}$	50.29$_{\pm 0.30}$	39.40$_{\pm 0.19}$
	w/o LLM Mixup	61.77$_{\pm 3.19}$	52.79$_{\pm 0.84}$	64.08$_{\pm 6.73}$	52.11$_{\pm 5.42}$	53.83$_{\pm 3.55}$	45.30$_{\pm 1.89}$	60.44$_{\pm 0.13}$	47.38$_{\pm 0.08}$
	w/o Smooth	60.09$_{\pm 4.16}$	49.96$_{\pm 5.93}$	58.35$_{\pm 4.36}$	49.04$_{\pm 3.13}$	54.96$_{\pm 4.26}$	44.58$_{\pm 1.39}$	56.38$_{\pm 0.22}$	45.34$_{\pm 0.15}$
	TRACI	**68.89**$_{\pm 6.08}$	**57.65**$_{\pm 4.92}$	**67.15**$_{\pm 6.50}$	**55.67**$_{\pm 7.68}$	**60.57**$_{\pm 3.37}$	**48.59**$_{\pm 1.07}$	**64.72**$_{\pm 5.28}$	**49.43**$_{\pm 3.11}$



(a) Cora (b) CiteSeer

Figure 2: Sensitivity analysis of group number and epoch interval for updating correlation scores under an imbalance ratio of 10.

both the effectiveness and robustness of the proposed approach. More comprehensively, these results further validate that TRACI enhances representation learning in the hidden space, leading to more discriminative representations.

4.3 Ablation Study

To further investigate the contribution of each module in TRACI, we introduce three variant models to evaluate the effectiveness of individual components, described as follows: (1) ***TRACI w/o LLM Expansion***: This variant removes the text-level augmentation generated by LLMs and replaces it with random perturbations to node-level features. (2) ***TRACI w/o LLM Mixup***: This version omits the computation of correlation scores using LLMs and instead uses the average of sample embeddings to generate group representations. (3) ***TRACI w/o Smooth***: This variant employs hard k-means clustering in the embedding space, replacing smooth k-means which is designed to mitigate the effects of class imbalance. Based on above variants, the effectiveness of TRACI can be demonstrated.

Table 3 presents the performance of TRACI and its variants on the Cora, CiteSeer, WikiCS, and PubMed datasets under imbalance ratios $\rho = 10$ and $\rho = 20$. Several key observations can be drawn from the results: First, ***TRACI w/o LLM Expansion*** exhibits a significant performance decline compared to the full model TRACI, indicating that the diverse views generated by LLMs provide richer semantic information at both the textual and embedding levels than simple feature perturbations. Second, the performance of ***TRACI w/o LLM Mixup*** also deteriorates under imbalanced settings, highlighting the importance of learning minority-aware group representations using correlation scores derived from LLMs based on semantic relevance. Finally, removing smooth k-means in ***TRACI w/o Smooth*** leads to further performance drops, demonstrating its effectiveness in mitigating overly uniform clustering and better handling real-world imbalanced conditions. Overall, these ablation results comprehensively validate the contributions of each module in enhancing the robustness and performance of TRACI.

4.4 Sensitivity Analysis on Hyperparameters and LLM Choice

In this section, we conduct a sensitivity analysis of TRACI from two perspectives: (i) the impact of hyperparameter configurations on model performance, and (ii) the effect of different LLM choices on the overall effectiveness of TRACI. The results are presented in Figure 2 and Table 4.

Effect of Hyperparameters. We examine two hyperparameters involved in the text-guided mixup process: the number of groups and the epoch interval for updating correlation scores. By default, the group number and the epoch interval are set to 100 and 50, respectively, for both the Cora and

CiteSeer datasets, as reported in Table 1. In the sensitivity analysis, we vary the group number in {50, 75, 100, 125 150 } and the epoch interval in {40, 50, 60, 100, 200}, while keeping all other hyperparameters fixed. Figure 2 presents the ACC, NMI and F1 scores under an imbalance ratio of 10. On the Cora dataset, the performance generally declines as the group number increases (e.g., ACC drops from 74.39% to 72.00%) and as the epoch interval becomes longer (e.g., ACC drops from 73.85% to 70.82%). This observations suggest that an excessively large number of groups may impair the model's ability to capture minority-aware representations, while a prolonged epoch interval may result in insufficient updates to the mixup groups, thereby weakening the interaction between majority and minority classes in a long-tailed distribution. In conclusion, although the sensitivity trends differ slightly across datasets under an imbalance ratio of 10, TRACI consistently demonstrates robust and competitive performance across a wide range of hyperparameter settings.

Effect of LLM Selection. In our TRACI, we leverage the capability of large language models (LLMs) to generate text-level augmented views and to comprehend contextual semantics, thereby enhancing performance in imbalance graph clustering. To quantitatively assess the impact of LLM selection, we evaluate TRACI using the open-weight LLM DeepSeek-V3 (Liu et al., 2024a) and three cost-effective ChatGPT variants: GPT-3.5 (Brown et al., 2020), GPT-4o-mini (Achiam et al., 2023) and GPT-4.1-mini (OpenAI, 2025). The corresponding results are reported in Figure 4. On the Cora dataset, the variant of TRACI equipped with GPT-4o-mini generally yields the best overall performance, while the GPT-3.5 based model also delivers impressive and competitive results. Although DeepSeek-V3 shows comparatively lower performance overall, it remains competitive on certain metric(e.g., NMI of 0.4295 ± 0.0237 on CiteSeer).

Table 4: Effect of LLM selection on TRACI' performance.

Dataset	Metric	DeepSeek-V3	GPT-3.5	GPT-4o-mini	GPT-4.1-mini
Cora	ACC	$69.27_{\pm 0.39}$	$73.16_{\pm 1.05}$	$73.48_{\pm 2.21}$	$70.07_{\pm 4.77}$
	NMI	$53.81_{\pm 0.39}$	$56.52_{\pm 1.64}$	$55.60_{\pm 0.10}$	$55.22_{\pm 1.83}$
	F1	$61.41_{\pm 0.43}$	$63.79_{\pm 0.77}$	$67.03_{\pm 3.79}$	$61.11_{\pm 6.13}$
CiteSeer	ACC	$66.74_{\pm 7.13}$	$68.92_{\pm 0.83}$	$67.15_{\pm 0.17}$	$68.14_{\pm 0.03}$
	NMI	$42.95_{\pm 2.37}$	$43.68_{\pm 0.20}$	$41.72_{\pm 0.10}$	$43.40_{\pm 0.00}$
	F1	$58.75_{\pm 7.26}$	$61.34_{\pm 0.99}$	$60.44_{\pm 0.14}$	$61.77_{\pm 0.01}$

4.5 Case Study

To facilitate a more intuitive understanding of TRACI, we present two illustrative case studies. The first case, showing in Figure 4 highlights the contribution of LLM-based expansion. Specifically, we examine node 25, a sample from the minority class *Reinforcement Learning* in the Cora dataset under an imbalance ratio of 10. Without LLM expansion, this node is misclassified into the majority class *Genetic Algorithms*. In contrast, our approach leverage the LLM's strong textual understanding to generate augmented textual views for each sample. As a result, the LLM-guided view enables the correct identification of node 25 as belonging to the intended minority class. This positive feedback from the LLM further boosts the learning of minority-aware representations under class-imbalanced settings. The second case study, involving CiteSeer's minority-class node 141, labeled as *Human Computer Interaction*, demonstrates text-guided long-tailed mixup. Our method assigns it higher mixing weight based on LLM-assigned semantic relevance, enabling correct clustering, whereas equal weighting results in misclassification into the majority class *Information Retrieval*. In conclusion, these two representative cases strongly demonstrate the reliability and interpretability of TRACI in learning minority-aware representations under long-tailed class distributions.

5 Conclusion

In this work, we propose a novel text-guided group mixup framework with canonical correlation alignment to address the challenge of imbalanced text-attributed graph clustering. TRACI leverages large language models (LLMs) to generate semantically enriched text augmentations, enhancing representation consistency across views. A text-guided mixup strategy is employed to adaptively prioritize minority samples based on LLM-derived semantic relevance. Furthermore, LLM-generated ranking signals are utilized to refine the representations of boundary nodes. Extensive experiments demonstrate the effectiveness of TRACI under long-tailed imbalanced conditions. In future work, we plan to extend our method to cluster multi-modal graphs that integrate diverse information sources (e.g., text and images) (Fang et al., 2025), and further explore its scalability and applicability to real-world, large-scale imbalanced datasets.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013.

Juscimara G Avelino, George DC Cavalcanti, and Rafael MO Cruz. Resampling strategies for imbalanced regression: a survey and empirical analysis. *Artificial Intelligence Review*, 57(4):82, 2024.

Aritra Bhowmick, Mert Kosan, Zexi Huang, Ambuj Singh, and Sourav Medya. Dgcluster: A neural framework for attributed graph clustering via modularity maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11069–11077, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Miguel Carvalho, Armando J Pinho, and Susana Brás. Resampling approaches to handle class imbalance: a review from a data perspective. *Journal of Big Data*, 12(1):71, 2025.

Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. Topology-imbalance learning for semi-supervised node classification. *Advances in Neural Information Processing Systems*, 34:29885–29897, 2021.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

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

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

Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. S3gc: scalable self-supervised graph clustering. *Advances in Neural Information Processing Systems*, 35:3248–3261, 2022.

Hongwei Ding, Nana Huang, Yaoxin Wu, and Xiaohui Cui. Improving imbalanced medical image classification through gan-based data augmentation methods. *Pattern Recognition*, 166:111680, 2025.

Yi Fang, Bowen Jin, Jiacheng Shen, Sirui Ding, Qiaoyu Tan, and Jiawei Han. Graphgpt-o: Synergistic multimodal comprehension and generation on graphs. *arXiv preprint arXiv:2502.11925*, 2025.

Yiwei Fu, Yuxing Zhang, Chunchun Chen, JianwenMa JianwenMa, Quan Yuan, Rong-Cheng Tu, Xinli Huang, Wei Ye, Xiao Luo, and Minghua Deng. Mark: Multi-agent collaboration with ranking guidance for text-attributed graph clustering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6057–6072, 2025.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Xinyi Gao, Wentao Zhang, Tong Chen, Junliang Yu, Hung Quoc Viet Nguyen, and Hongzhi Yin. Semantic-aware node synthesis for imbalanced heterogeneous information networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 545–555, 2023.

Kushankur Ghosh, Colin Bellinger, Roberto Corizzo, Paula Branco, Bartosz Krawczyk, and Nathalie Japkowicz. The class imbalance problem in deep learning. *Machine Learning*, 113 (7):4845–4901, 2024.

C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pp. 89–98, 1998.

Jiangpeng He. Gradient reweighting: Towards imbalanced class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16668–16677, 2024a.

Yudong He. Imbalanced data clustering using equilibrium k-means. *arXiv preprint arXiv:2402.14490*, 2024b.

Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 448–459, 2025.

Shengxiang Hu, Guobing Zou, Song Yang, Shiyi Lin, Yanglan Gan, Bofeng Zhang, and Yixin Chen. Large language model meets graph neural network in knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17295–17304, 2025.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Zhenhua Huang, Yinhao Tang, and Yunwen Chen. A graph neural network-based node classification model on class-imbalanced graph data. *Knowledge-Based Systems*, 244:108538, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1827–1843, 2023.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *arXiv preprint arXiv:2403.04468*, 2024.

Azal Ahmad Khan, Omkar Chaudhari, and Rohitash Chandra. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Systems with Applications*, 244:122778, 2024.

Gayan K Kulatilleke, Marius Portmann, and Shekhar S Chandra. Scgc: Self-supervised contrastive graph clustering. *Neurocomputing*, 611:128629, 2025.

Shuxian Li, Liyan Song, Xiaoyu Wu, Zheng Hu, Yiu-ming Cheung, and Xin Yao. Multi-class imbalance classification based on data distribution and adaptive weights. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5265–5279, 2024.

Yang Li, Ji Zhang, Lin Zhang, and Kan Li. A client-level dynamic federated learning reweighting strategy for long-tailed classification. *Expert Systems with Applications*, pp. 128642, 2025.

Zhixin Li and Yuheng Jia. Conmix: Contrastive mixup at representation level for long-tailed deep clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collaboration: A comprehensive survey. *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7668–7684, 2025.

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

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

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

Zhining Liu, Ruizhong Qiu, Zhichen Zeng, Hyunsik Yoo, David Zhou, Zhe Xu, Yada Zhu, Kommy Weldemariam, Jingrui He, and Hanghang Tong. Class-imbalanced graph learning without class rebalancing. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 31747–31772, 2024c.

Yihong Ma, Yijun Tian, Nuno Moniz, and Nitesh V Chawla. Class-imbalanced learning on graphs: A survey. *ACM Computing Surveys*, 57(8):1–16, 2025.

Zhengyang Mao, Wei Ju, Yifang Qin, Xiao Luo, and Ming Zhang. Rahnet: Retrieval augmented hybrid network for long-tailed graph classification. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 3817–3826, 2023.

Costas Mavromatis, Vassilis N Ioannidis, Shen Wang, Da Zheng, Soji Adeshina, Jun Ma, Han Zhao, Christos Faloutsos, and George Karypis. Train your own gnn teacher: Graph-aware distillation on textual graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 157–173. Springer, 2023.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Thomas Nagler, Lennart Schneider, Bernd Bischl, and Matthias Feurer. Reshuffling resampling splits can improve generalization of hyperparameter optimization. *Advances in Neural Information Processing Systems*, 37:40486–40533, 2024.

OpenAI. Gpt-4.1-mini model, 2025. URL https://platform.openai.com/docs/models/gpt-4.1-mini.

Joonhyung Park, Jaeyun Song, and Eunho Yang. Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International conference on learning representations*, 2021.

Yiran Qiao, Xiang Ao, Yang Liu, Jiarong Xu, Xiaoqian Sun, and Qing He. 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*, pp. 232–241, 2025.

Peng Qin, Yaochun Lu, Weifu Chen, Defang Li, and Guocan Feng. Aagcn: An adaptive data augmentation for graph contrastive learning. *Pattern Recognition*, 163:111471, 2025.

Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 1390–1398, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Tao Ren, Haodong Zhang, Yifan Wang, Wei Ju, Chengwu Liu, Fanchun Meng, Siyu Yi, and Xiao Luo. Mhgc: Multi-scale hard sample mining for contrastive deep graph clustering. *Information Processing & Management*, 62(4):104084, 2025.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Toward robustness in multi-label classification: A data augmentation strategy against imbalance and noise. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 21592–21601, 2024.

Li Sun, Zhenhao Huang, Yujie Wang, Hongbo Lv, Chunyang Liu, Hao Peng, and Philip S Yu. Isosel: Isometric structural entropy learning for deep graph clustering in hyperbolic space. *arXiv preprint arXiv:2504.09970*, 2025.

Jilun Tian, Yuchen Jiang, Jiusi Zhang, Hao Luo, and Shen Yin. A novel data augmentation approach to fault diagnosis with class-imbalance problem. *Reliability Engineering & System Safety*, 243: 109832, 2024.

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

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

Mudit Verma, Siddhant Bhambri, and Subbarao Kambhampati. Theory of mind abilities of large language models in human-robot interaction: An illusion? In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 36–45, 2024.

Danny Wang, Ruihong Qiu, Guangdong Bai, and Zi Huang. Text meets topology: Rethinking out-of-distribution detection in text-rich networks. *arXiv preprint arXiv:2508.17690*, 2025a.

Leyao Wang, Yu Wang, Bo Ni, Yuying Zhao, Hanyu Wang, Yao Ma, and Tyler Derr. Save-tag: Semantic-aware vicinal risk minimization for long-tailed text-attributed graphs. *arXiv preprint arXiv:2410.16882*, 2024.

Zichong Wang, Zhibo Chu, Thang Viet Doan, Shaowei Wang, Yongkai Wu, Vasile Palade, and Wenbin Zhang. Fair graph u-net: A fair graph learning framework integrating group and individual awareness. In *proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 28485–28493, 2025b.

Feize Wu, Yun Pang, Junyi Zhang, Lianyu Pang, Jian Yin, Baoquan Zhao, Qing Li, and Xudong Mao. Core: Context-regularized text embedding learning for text-to-image personalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8377–8385, 2025.

Xuanting Xie, Bingheng Li, Erlin Pan, Zhaochen Guo, Zhao Kang, and Wenyu Chen. One node one model: Featuring the missing-half for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21688–21696, 2025.

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

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*, 2021.

Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10834–10842, 2023.

Tong Ye, Yangkai Du, Tengfei Ma, Lingfei Wu, Xuhong Zhang, Shouling Ji, and Wenhai Wang. Uncovering llm-generated code: A zero-shot synthetic code detector via code rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 968–976, 2025.

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang 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*, volume 39, pp. 13087–13095, 2025.

Chunhui Zhang, Chao Huang, Yijun Tian, Qianlong Wen, Zhongyu Ouyang, Youhuan Li, Yanfang Ye, and Chuxu Zhang. When sparsity meets contrastive models: Less graph data can bring better class-balanced representations. In *International Conference on Machine Learning*, pp. 41133–41150. PMLR, 2023.

Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1298–1301, 2024.

Yunzhi Zhang, Zizhang Li, Matt Zhou, Shangzhe Wu, and Jiajun Wu. The scene language: Representing scenes with programs, words, and embeddings. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24625–24634, 2025.

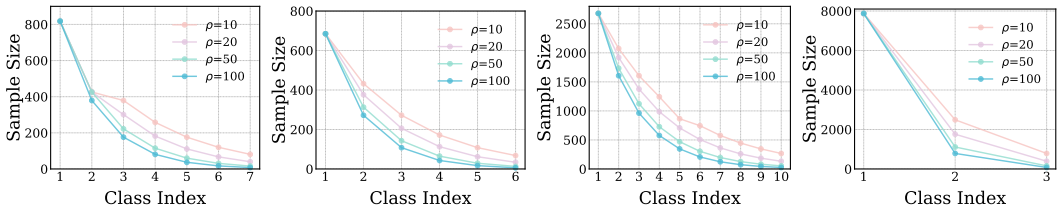
Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841, 2021.

Chuang Zhou, Jiahe Du, Huachi Zhou, Hao Chen, Feiran Huang, and Xiao Huang. Text-attributed graph learning with coupled augmentations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10865–10876, 2025.

Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pp. 2183–2197, 2025.

A DATASETS

To evaluate TRACI under class-imbalanced scenarios, we construct several novel graphs with varying imbalance ratios for these four acknowledged datasets, Cora (McCallum et al., 2000), CiteSeer (Giles et al., 1998), WikiCS (Mernyei & Cangea, 2020) and PubMed (Sen et al., 2008). The sampling criterion aims to ensure that nodes in each class follow a long-tailed distribution while preserving the overall connectivity as much as possible. Specifically, nodes with higher degrees are retained, whereas those with lower degrees are removed accordingly. Detailed statistics are provided in Table 5.



(a) Cora (b) CiteSeer (c) WikiCS (d) PubMed

Figure 3: Sample size of per class in the Cora, CiteSeer, WikiCS and Pubmed with varying imbalance ratios (ρ=10, 20, 50 and 100). It is evident that the sample size follows a long-tailed distribution.

Table 5: Statistics of datasets with varying imbalance ratios.

Imbalance	$\rho = 10$		$\rho = 20$		$\rho = 50$		$\rho = 100$			
Dataset	#Nodes	#Edges	#Nodes	#Edges	#Nodes	#Edges	#Nodes	#Edges	#Features	#Clusters
Cora	2,258	4,602	1,945	3,878	1,688	3,178	1,516	2,801	384	7
CiteSeer	1,737	2,791	1,476	2,375	1,248	2,023	1,131	1,807	384	6
WikiCS	10,848	214,593	9,117	206,770	7,496	186,880	6,645	172,124	384	10
PubMed	11,152	31,890	10,028	27,704	9,145	23,495	8,740	21,381	384	3

B BASELINES

The compared approaches are comprehensively described as follows:

- DMoN (Tsitsulin et al., 2023) is an unsupervised clustering framework for attributed graphs based on GNNs. It enables end-to-end differentiable optimization of cluster assignments through modularity maximization combined with collapse regularization.
- Dink-Net (Liu et al., 2023a) proposes a self-supervised clustering approach designed to scale to large graphs. Specifically, it jointly models representation learning and clustering by pushing apart different clusters and pulling nodes closer to their assigned clusters, using an augmentation-based discriminative strategy.
- HSAN (Liu et al., 2023b) is a contrastive deep graph clustering framework tailored to handle both hard positive and hard negative sample pairs. It introduces a similarity measure that jointly considers both attribute and structural information.
- S^3GC (Devvrit et al., 2022) leverages contrastive learning in combination with Graph Neural Networks and node features, making it well-suited for large-scale datasets.
- DGCluster (Bhowmick et al., 2024) proposes a novel framework that uses pairwise soft memberships between nodes to address the graph clustering problem through modularity maximization. Its computational complexity scales linearly with the size of the graph, making it well-suited for large-scale datasets.
- MAGI (Liu et al., 2024b) is a contrastive learning method based on modularity maximization. It forms positive pairs from nodes within the same module and negative pairs from nodes belonging to different modules, thereby effectively leveraging the graph structure.

- IsoSEL (Sun et al., 2025) proposes a Lorentz tree contrastive learning framework with isometric augmentation to refine the deep partitioning tree in hyperbolic space, while also incorporating attribute information.

For the above baselines, we retrain each model on our constructed datasets and report their performance averaged over five runs to ensure a fair comparison.

C Evaluation Metrics

In this work, we use three widely used clustering metrics: accuracy (ACC), normalized mutual information (NMI) and macro F1 score as metrics to evaluate comprehensively clustering performance of methods.

- **ACC** is a commonly used metric for evaluating classification performance. In the context of unsupervised clustering, the predicted clusters must first be aligned with the ground truth labels using the Hungarian algorithm, based on the confusion matrix $\mathbf{C} \in \mathbb{R}^{K\times K}$, where K is the number of classes and $C_{i,j}$ denotes the number of samples with ground truth label i and predicted label j. Specifically, ACC is defined as:

$$\text{ACC} = \frac{\sum_{i=1}^{K} C_{i,i}}{\sum_{i=1}^{K}\sum_{j=1}^{K} C_{i,j}}. \tag{14}$$

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

$$\text{NMI} = \frac{I(X,Y)}{\max\{H(X), H(Y)\}}, \tag{15}$$

  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 a widely used metric for evaluating multi-class classification performance. We compute the macro F1 score by taking the arithmetic mean of the per-class F1 scores, treating all classes equally regardless of their support. For a dataset with K classes, the macro F1 score is calculated as:

$$\text{F1 score} = \frac{\sum_{k=1}^{K} \text{F1 score}_k}{K}, \tag{16}$$

  where the F1 score for each class k is given by:

$$\text{F1 score}_k = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{17}$$

  where TP denotes the number of samples correctly predicted as positive; FP represents the number samples incorrectly predicted as positive; FN is the number of samples incorrectly predicted as negative; and TN refers to the number of samples correctly predicted as negative.

D Proof of Theorem 3.1

In this section, we present a detailed proof of Theorem 3.1, which is restated below for completeness.

Proof of Theorem 3.1: Without loss of generality, we assume the encoder f_Θ is L-Lipschitz continuous. Let $\mathcal{F}$ be the hypothesis class of L-Lipschitz encoders. For contrastive loss $\mathcal{L}_{cl}$ and $\mathcal{L}_{mixup}$, we can decompose them in the following formula:

$$\mathcal{L}_{cl} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\theta(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})/\tau}}{e^{\theta(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})/\tau} + \sum_{\mathbf{z}_i^-} e^{\theta(\mathbf{z}_i^{(1)}, \mathbf{z}_i^-)/\tau}} = \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{cl}}\big(f_\Theta(\mathbf{x}_i)\big), \tag{18}$$

and

$$\mathcal{L}_{\text{mixup}} = -\frac{1}{M}\sum_{m=1}^{M} \log \frac{e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_m^{(2)})/\tau}}{e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_m^{(2)})/\tau} + \sum_{\mathbf{h}_m^-} e^{\theta(\mathbf{h}_m^{(1)}, \mathbf{h}_m^-)/\tau}} = \frac{1}{M}\sum_{m=1}^{M} \ell_{\text{mixup}}\big(f_\Theta(\mathbb{G}_m)\big), \tag{19}$$

where $\theta(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}) = \frac{(\mathbf{z}_i^{(1)})^T \cdot \mathbf{z}_i^{(2)}}{||\mathbf{z}_i^{(1)}|| \cdot ||\mathbf{z}_i^{(2)}||}$ calculates the cosine similarity between two vectors.

For $\mathcal{L}_{CL}$, the empirical Rademacher complexity is:

$$\mathcal{R}(\mathcal{L}_{\text{cl}}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i \ell_{\text{cl}}\big(f(\mathbf{x}_i)\big) \right], \tag{20}$$

where $\ell_{\text{cl}}(f_\Theta(\mathbf{x}_i))$ is the contrastive loss for the positive pair $(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})$. Subsequently, we try to bound the empirical Rademacher complexity for ℓ_{cl}. We define a function ϕ as the normalization-score operation for positive-negative sample pairs:

$$\phi\left(\theta(\mathbf{z}_i, \mathbf{z}_i^+), \theta(\mathbf{z}_i, \mathbf{z}^-)\right) = -\log \frac{e^{\theta(\mathbf{z}_i, \mathbf{z}_i^+)/\tau}}{e^{\theta(\mathbf{z}_i, \mathbf{z}_i^+)/\tau} + \sum_{\mathbf{z_i}^-} e^{\theta(\mathbf{z}_i, \mathbf{z}_i^-)/\tau}}. \tag{21}$$

We then prove ϕ is $\frac{\sqrt{2}}{\tau}$-Lipschitz continuous.

Define vector $\mathbf{v} = \left(\theta(\mathbf{z}_i, \mathbf{z}_i^+), \theta(\mathbf{z}_i, \mathbf{z}_j)_{j \neq i}, \theta(\mathbf{z}_i, \mathbf{z}_j^-)_{j \neq i}\right) \in \mathbb{R}^{2N-1}$, then ϕ can be written as a function of $\mathbf{v}$

$$\phi(v) = -\log \frac{e^{v_0/\tau}}{\sum_{j=0}^{2N-1} e^{v_j/\tau}} = \log \sum_{j=0}^{2N-1} e^{(v_j - v_0)/\tau}. \tag{22}$$

The derivative of ϕ with respect to v_0 is

$$\nabla_{v_0} \phi = \frac{1}{\tau} \cdot \left(\frac{e^{v_0/\tau}}{\sum_{j=0}^{2N-1} e^{v_j/\tau}} - 1 \right) \triangleq \frac{1}{\tau} \cdot p_0, \tag{23}$$

and the derivative of ϕ with respect to $v_j (j > 0)$ is

$$\nabla_{v_j} \phi = \frac{1}{\tau} \cdot \frac{e^{v_j/\tau}}{\sum_{j=0}^{2N-1} e^{v_j/\tau}} \triangleq \frac{1}{\tau} \cdot p_j. \tag{24}$$

Thus, the ℓ_2-norm of ϕ satisfies: $||\nabla_{\mathbf{v}} \phi||_2 = \frac{1}{\tau} \sqrt{(1 - p_0)^2 + \sum_{j \geq 1} p_j^2} \leq \frac{\sqrt{2}}{\tau}$, where the inequality holds because $\sum_{j \geq 0} p_j = 1$, which implies $\sum_{j \geq 0} p_j^2 \leq 1$. Therefore, ϕ is proved to be $\frac{\sqrt{2}}{\tau}$-Lipschitz continuous. Subsequently, by applying the contraction lemma with $f \in \mathcal{F}$, we can bound the empirical complexity for $\mathcal{L}_{\text{cl}}$ as follows:

$$\begin{aligned}
\mathcal{R}(\mathcal{L}_{\text{CL}}) &= \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i \ell_{\text{cl}}\big(f(\mathbf{x}_i)\big) \right] \\
&\leq \frac{\sqrt{2}L}{\tau} \cdot \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i \|\mathbf{z}_i\| \right] \\
&\leq \frac{\sqrt{2}L}{\tau} \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i \right] \\
&\leq \frac{2L}{\tau} \sqrt{\frac{\log N}{N}},
\end{aligned} \tag{25}$$

where the first equality holds since ϕ is $\frac{\sqrt{2}}{\tau}$-Lipschitz continuous and $f \in \mathcal{F}$ is L-Lipschitz continuous, the second inequality follows from the normalization condition $||\mathbf{z}_i|| \leq 1$ is normalized, and the third inequality holds due to Massart's theorem, which states that $\mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i \right] \leq \sqrt{\frac{2 \log N}{N}}$.

For $\mathcal{L}_{mixup}$, group embeddings are derived from $\mathbf{h}_m = \sum_{n \in \mathbb{G}_m} s_{m,n} \mathbf{z}_n$ reduces the effective sample size from N to M, then the empirical Rademacher complexity is

$$\mathcal{R}(\mathcal{L}_{\text{mixup}}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{M} \sum_{m=1}^{M} \sigma_m \ell_{\text{mixup}}\big(f(\mathbb{G}_m)\big) \right]. \tag{26}$$

where $\sigma_i \in \{\pm 1\}$ are Rademacher variables, and ℓ_{mixup} is the contrastive loss for the pair $(h_m^{(1)}, h_m^{(2)})$. Subsequently, we prove the empirical Rademacher complexity is bounded for $\mathcal{L}_{mixup}$. Specifically, since $\sum_{n \in \mathbb{G}_m} s_{m,n} = 1$, then we have

$$\|\mathbf{h}_m\| \leq \sum_{n \in G_m} s_{mn} \|\mathbf{z}_n\| \leq \|\mathbf{s}_m\|_1 \cdot \|\mathbf{z}\|_\infty \leq 1, \tag{27}$$

where the first inequality follows from the triangle inequality for norms, the second inequality holds by Hölder inequality, and the third inequality holds since $\|\mathbf{z}_n\| \leq 1$. Similar to Equation 25, we can derive the bound for $\mathcal{L}_{\text{mixup}}$:

$$\begin{aligned}
\mathcal{R}(\mathcal{L}_{\text{mixup}}) &= \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{M} \sum_{m=1}^{M} \sigma_m \ell_{\text{mixup}}(f(\mathbf{h}_m)) \right] \\
&\leq \frac{\sqrt{2}L}{\tau} \cdot \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{M} \sum_{m=1}^{M} \sigma_m \|\mathbf{h}_m\|_2 \right] \qquad (28) \\
&\leq \frac{\sqrt{2}L}{\tau} \cdot \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{M} \sum_{m=1}^{M} \sigma_m \right] \qquad (29) \\
&\leq \frac{2L}{\tau} \sqrt{\frac{\log M}{M}}. \qquad (30)
\end{aligned}$$

$\sqrt{\frac{\log M}{M}} \ll \sqrt{\frac{\log N}{N}}$ since $M \ll N$ under imbalanced scenarios, then we can derive that the empirical complexity for $\mathcal{L}_{\text{mixup}}$ and $\mathcal{L}_{\text{cl}}$ satisfies that $\mathcal{R}(\mathcal{L}_{\text{mixup}}) \leq \mathcal{R}(\mathcal{L}_{\text{cl}})$, implying a tighter generalization error bound for $\mathcal{L}_{\text{mixup}}$ compared to $\mathcal{L}_{\text{cl}}$.

According to Theorem 26.5 in (Shalev-Shwartz & Ben-David, 2014), we can derive the generalization bound under the condition that ℓ_{cl} and ℓ_{mixup} are bounded, which is ensured by the normalization of $\mathbf{z}$ in the embedding space. Then, with probability at least $1 - \delta$, for any encoder $f \in \mathcal{F}$, there exists a constant C such that the generalization error bound satisfies:

$$\mathcal{E} - \mathcal{E}^* \leq 2\mathcal{R} + C\sqrt{\frac{\log(1/\delta)}{N}}, \tag{31}$$

where $\mathcal{R}$ is the empirical Rademacher complexity for loss function $\mathcal{L}$. By substituting the loss function with $\mathcal{L}$cl and $\mathcal{L}$mixup, we can derive the respective generalization error bounds for the above contrastive losses, which are expressed as follows:

$$\mathcal{E}(\mathcal{L}_{\text{mixup}}) - \mathcal{E}^*(\mathcal{L}_{\text{mixup}}) \leq C\left(\sqrt{\frac{\log M}{M}} + \sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{32}$$

$$\mathcal{E}(\mathcal{L}_{\text{cl}}) - \mathcal{E}^*(\mathcal{L}_{\text{cl}}) \leq C\left(\sqrt{\frac{\log N}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{33}$$

where C is a constant. Thus, we conclude that $\mathcal{L}_{\text{mixup}}$ yields a significantly tighter bound on the generalization error in comparison to $\mathcal{L}_{\text{cl}}$. □

E Prompt Design

Table 6 presents the prompts used to generate textual augmented views via the LLM.

Table 7 presents the prompts provided to the LLM for different purposes, including *Group Mixup*, *Concept Induction*, and *Ranking Guidance*.

F Implementation Details

The implementation details are organized into two parts: the first describes the process of updating cluster centroids and assigning soft labels using the smooth k-means approach proposed by (He, 2024b), while the second outlines the hyperparameter settings of TRACI for different datasets.

Table 6: Prompts for *Data Expansion*.

Augmentation	Prompt
Technical	You are an AI assistant specializing in text optimization. Please rewrite the following article while preserving its core ideas. ***Requirements:*** 1. Use formal academic language with domain-specific terminology. 2. Maintain strict factual consistency with the original content.
Colloquial	You are an AI assistant specializing in text optimizing. Please simplify this article for non-experts while retaining key information. ***Requirements:*** 1. Avoid technical jargon. 2. Use short sentences and everyday vocabulary.

Table 7: Prompts for *Group Mixup*, *Concept Induction* and *Ranking Guidance*.

Usage of the LLM	Prompt
Group Mixup	You are an AI assistant specializing in semantic analysis. Please evaluate the contribution and confidence scores of each text with respect to the whole cluster. ***Requirements for contribution scores:*** 1. The contribution score should range from 0.00 to 1.00. A lowest score of 0.00 indicates the lowest contribution while 1.00 reflects the highest contribution. 2. Consider its semantic relevance to the cluster, the density it contains, its conceptual representativeness, and its contextual coherence with other texts. 3. Derive an overall contribution score by synthesizing these individual evaluations. ***Requirements for confidence scores:*** 1. Fall within the range of 0.00 to 1.00. 2. Evaluate the accuracy and credibility of the contribution score.
Concept Induction	You are an AI assistant specializing in topic modeling. Please examine the core themes and contextual elements within the input texts to generate a concise, accurate topic name. ***Requirements:*** 1. Analyze the commonalities and core content of these samples. 2. Provide a concise summary of the cluster's theme. 3. Output the theme as a short name.
Ranking Guidance	You are an AI assistant specializing in text prediction. Please analyze the content to determine the most relevant topic cluster it belongs to. ***Requirements:*** 1. Consider comprehensively which cluster this article most likely belongs to. 2. The optimal clusters are ¡Topics induced by Concept Induction¿. 3. Answer the cluster number directly.

Smooth K-means. Inspired by the problem of imbalanced clustering, (He, 2024b) proposes a novel method based on k-means with a Boltzmann operator, which we adopt in place of the traditional hard k-means for clustering. Specifically, the cluster centroids $\mathbf{c}_k$ and the weighted cluster assignments $\omega n, k$ are defined as follows:

$$\omega_{n,k} = \frac{e^{-\gamma \cdot d_{n,k}}}{\sum_{k=1}^{K} e^{-\gamma \cdot d_{n,k}}} \left[1 - \gamma \left(d_{n,k} - \frac{\sum_{k=1}^{K} d_{n,k} e^{-\gamma \cdot d_{n,k}}}{\sum_{k=1}^{K} e^{-\gamma \cdot d_{n,k}}} \right) \right], \tag{34}$$

and

$$\mathbf{c}_k = \frac{\sum_n w_{n,k}\mathbf{z}_n}{\sum_n w_{n,k}}, \tag{35}$$

where $\omega_{n,k}$ denotes the weight of the n-th sample with respect to the k-th cluster such that $\sum_k \omega_{n,k} = 1$, $d_{n,k}$ is the distance between the n-th sample and the k-th cluster, γ is a smoothing hyperparameter, and $\mathbf{z}_n$ represents the embedding of the n-th sample.

Hyperparamter Settings. Additional hyperparameter details are thoroughly described in Table 8.

Table 8: Hyperparameter settings. *Hidden Dimensions* refers to the dimension of the GCN encoder in the latent space. lr_1 and lr_2 denote the learning rates used during the warmup and fine-tuning stages, respectively. α and β are hyperparameters that weight different loss components, while γ controls the strength of the smooth k-means term. τ_1 and τ_2 are the temperature parameters for the mixup loss and ranking loss, respectively. M represents the number of groups.

Dataset	Hidden Dimensions	lr_1	lr_2	α	β	γ	τ_1	τ_2	M
Cora	[64]	0.0005	0.0001	0.1	0.9	10	0.5	0.01	100
CiteSeer	[128, 64]	0.0001	0.0001	0.2	0.9	10	0.9	0.01	100
WikiCS	[256, 128]	0.0001	0.0001	0.1	0.9	16	0.9	0.09	500
PubMed	[128, 64]	0.0005	0.0001	0.1	0.9	10	0.5	0.01	500

G Performance on more imbalanced scenarios

Previously, we have evaluated the performance of TRACI against baselines under imbalance ratios of $\rho = 10$ and $\rho = 20$. In this section, we explore more extreme scenarios with imbalance ratios of $\rho = 50$ and $\rho = 100$, where the minority class contains substantially fewer samples, making the imbalanced graph clustering considerably more challenging. As shown in Table 9, TRACI consistently demonstrates competitive overall performance across these settings, further highlighting its robustness and effectiveness under severe class imbalance conditions.

Table 9: Clustering performance of TRACI compared to baseline methods under more imbalanced scenarios ($\rho = 50$ and 100).

Imbalance	Dataset	Metric	DMoN	Dink-Net	HSAN	S^3GC	DGCluster	MAGI	IsoSEL	TRACI
$\rho = 50$	Cora	ACC	53.89$_{\pm 6.54}$	47.20$_{\pm 1.84}$	47.39$_{\pm 1.61}$	47.57$_{\pm 1.69}$	56.53$_{\pm 5.56}$	48.85$_{\pm 0.70}$	**62.32**$_{\pm 2.44}$	58.73$_{\pm 4.79}$
		NMI	**48.13**$_{\pm 0.48}$	46.30$_{\pm 0.53}$	44.71$_{\pm 0.39}$	43.37$_{\pm 0.43}$	44.72$_{\pm 1.27}$	46.00$_{\pm 0.42}$	44.96$_{\pm 2.32}$	47.44$_{\pm 1.22}$
		F1	39.78$_{\pm 3.26}$	37.33$_{\pm 1.90}$	40.24$_{\pm 1.17}$	37.57$_{\pm 1.57}$	42.48$_{\pm 3.22}$	39.79$_{\pm 0.43}$	38.41$_{\pm 4.31}$	**42.62**$_{\pm 4.41}$
	CiteSeer	ACC	54.87$_{\pm 5.55}$	55.46$_{\pm 3.54}$	50.64$_{\pm 0.96}$	47.69$_{\pm 3.34}$	53.89$_{\pm 3.65}$	56.36$_{\pm 0.44}$	57.02$_{\pm 8.10}$	**59.63**$_{\pm 9.22}$
		NMI	36.40$_{\pm 1.83}$	38.96$_{\pm 1.44}$	37.99$_{\pm 0.95}$	37.24$_{\pm 1.73}$	34.08$_{\pm 0.19}$	39.21$_{\pm 0.29}$	34.72$_{\pm 1.43}$	**39.97**$_{\pm 2.12}$
		F1	39.28$_{\pm 5.89}$	39.37$_{\pm 4.18}$	39.93$_{\pm 1.21}$	38.04$_{\pm 2.69}$	25.65$_{\pm 3.00}$	**45.32**$_{\pm 0.15}$	34.09$_{\pm 6.67}$	43.93$_{\pm 8.83}$
	WikiCS	ACC	37.79$_{\pm 2.82}$	54.97$_{\pm 5.64}$	54.37$_{\pm 6.02}$	56.18$_{\pm 1.20}$	47.93$_{\pm 5.49}$	49.46$_{\pm 1.85}$		**63.19**$_{\pm 5.81}$
		NMI	31.20$_{\pm 1.71}$	46.77$_{\pm 0.90}$	47.34$_{\pm 1.58}$	41.61$_{\pm 0.43}$	47.17$_{\pm 1.09}$	47.69$_{\pm 0.16}$	OOM	**48.98**$_{\pm 1.34}$
		F1	28.41$_{\pm 4.28}$	40.38$_{\pm 2.47}$	41.13$_{\pm 2.13}$	30.91$_{\pm 0.87}$	39.96$_{\pm 3.80}$	39.88$_{\pm 0.82}$		**49.79**$_{\pm 4.54}$
$\rho = 100$	Cora	ACC	50.63$_{\pm 4.62}$	49.09$_{\pm 3.14}$	45.83$_{\pm 3.53}$	45.77$_{\pm 3.37}$	55.40$_{\pm 5.75}$	46.21$_{\pm 0.90}$	52.61$_{\pm 7.74}$	**62.18**$_{\pm 2.40}$
		NMI	**45.03**$_{\pm 2.56}$	39.40$_{\pm 1.33}$	43.45$_{\pm 1.69}$	42.03$_{\pm 1.04}$	41.63$_{\pm 0.64}$	45.01$_{\pm 0.60}$	41.84$_{\pm 2.97}$	43.64$_{\pm 1.63}$
		F1	36.33$_{\pm 2.11}$	35.89$_{\pm 0.84}$	36.83$_{\pm 0.84}$	34.82$_{\pm 1.15}$	35.65$_{\pm 3.30}$	37.02$_{\pm 0.46}$	27.68$_{\pm 6.41}$	**37.93**$_{\pm 2.07}$
	CiteSeer	ACC	56.22$_{\pm 4.03}$	46.08$_{\pm 3.06}$	45.18$_{\pm 2.50}$	47.04$_{\pm 3.77}$	56.78$_{\pm 3.81}$	56.76$_{\pm 0.57}$	56.62$_{\pm 8.49}$	**57.24**$_{\pm 0.93}$
		NMI	37.22$_{\pm 1.89}$	35.69$_{\pm 1.21}$	34.39$_{\pm 1.96}$	36.61$_{\pm 0.73}$	33.27$_{\pm 0.81}$	**39.41**$_{\pm 0.45}$	32.60$_{\pm 0.84}$	39.15$_{\pm 0.33}$
		F1	38.61$_{\pm 4.81}$	34.65$_{\pm 4.71}$	35.17$_{\pm 1.66}$	36.31$_{\pm 2.53}$	25.53$_{\pm 3.78}$	**43.99**$_{\pm 0.23}$	31.34$_{\pm 6.08}$	41.72$_{\pm 0.27}$
	WikiCS	ACC	41.35$_{\pm 3.66}$	44.64$_{\pm 2.07}$	46.89$_{\pm 5.73}$	49.58$_{\pm 1.65}$	41.44$_{\pm 3.59}$	48.98$_{\pm 3.03}$		**50.78**$_{\pm 1.03}$
		NMI	33.45$_{\pm 1.81}$	43.07$_{\pm 1.00}$	45.36$_{\pm 1.87}$	36.31$_{\pm 0.52}$	45.01$_{\pm 0.72}$	**47.41**$_{\pm 0.21}$	OOM	46.47$_{\pm 0.87}$
		F1	28.87$_{\pm 4.10}$	32.71$_{\pm 1.45}$	34.82$_{\pm 2.07}$	17.87$_{\pm 0.37}$	32.34$_{\pm 0.69}$	36.23$_{\pm 1.15}$		**40.65**$_{\pm 1.64}$

H Algorithm

We present the overall algorithm of TRACI in Algorithm 1.

I Case Study

Here, we provide a case to illustrate the effect of TRACI.

Algorithm 1: The algorithm of TRACI

Input: $\mathcal{G} = (\mathbf{A}, \mathcal{D})$, GNN encoder $\mathcal{F}_\Theta$, training epoch T, the LLM execution interval T', learning rates lr_1 and lr_2, number of selected nodes n, number of groups M.

Augment $\mathcal{D}$ into $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$, yielding $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, respectively ;

/* Warmup */

for $t \leftarrow 1$ **to** T **do**

  Encode: $\mathbf{Z}^{(1)} = \mathcal{F}(\mathcal{G}^{(1)})$ and $\mathbf{Z}^{(2)} = \mathcal{F}(\mathcal{G}^{(2)})$;

  if $t \,\%\, T' == 0$ **then**

    $\mathbf{S}^{(1)}, \mathbf{S}^{(2)} \leftarrow$ GetWeightMatrix $(\mathcal{D}^{(1)}, \mathcal{D}^{(2)})$;

    Obtain the group-level synthetic embeddings $\mathbf{H}^{(i)} = (\mathbf{h}_m^{(1)})_{m=1}^{M}$ through Eq. 3 for $i = 1, 2$;

  Calculate the warmup loss $\mathcal{L}_{\text{warm}}$ in Eq. 5 using $\mathcal{L}_{\text{corr}}$ in Eq. 1 and $\mathcal{L}_{\text{mixup}}$ in Eq. 4;

  Update: $\Theta \leftarrow \Theta - \text{lr}_1 \cdot \nabla \mathcal{L}_{\text{warm}}$;

/* Fine-tuning with Ranking Guidance */

Obtain ranking guidance in Eq. 8 with challenge nodes $\mathcal{S}$;

Update $\mathbf{X}$ via mean pooling over nodes in $\mathcal{C}_{\text{LLM}}$;

for $t \leftarrow 1$ **to** T **do**

  Encode: $\mathbf{Z}^{(1)} = \mathcal{F}(\mathcal{G}^{(1)})$ and $\mathbf{Z}^{(2)} = \mathcal{F}(\mathcal{G}^{(2)})$;

  Obtain group embeddings similar to line 4 to 6;

  Calculate the fine-tuning loss $\mathcal{L}_{\text{fine}}$ in Eq. 11 using $\mathcal{L}_{\text{corr}}$ in Eq. 1, $\mathcal{L}_{\text{mixup}}$ in Eq. 4, $\mathcal{L}_{\text{rank}}$ in Eq. 10;

  Update: $\Theta \leftarrow \Theta - \text{lr}_2 \cdot \nabla \mathcal{L}_{\text{fine}}$;

Output: Final cluster assignments.

Function GetWeightMatrix $(\mathcal{D}^{(1)}, \mathcal{D}^{(2)})$ **:**

  Randomly partition the samples into M groups $\mathbb{G} = \{\mathbb{G}_1, \mathbb{G}_2, \cdots, \mathbb{G}_M\}$;

  for $m \leftarrow 1$ **to** M **do**

    for $i \leftarrow 1$ **to** 2 **do**

      Query contribution score $b_{mn}^{(i)}$ and confidence score $c_{mn}^{(i)}$ by an LLM through $\mathcal{P}_{\text{mix}}$;

      Compute correlation-based weight score $s_{mn}^{(i)}$ through Eq. 2;

  return $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}$

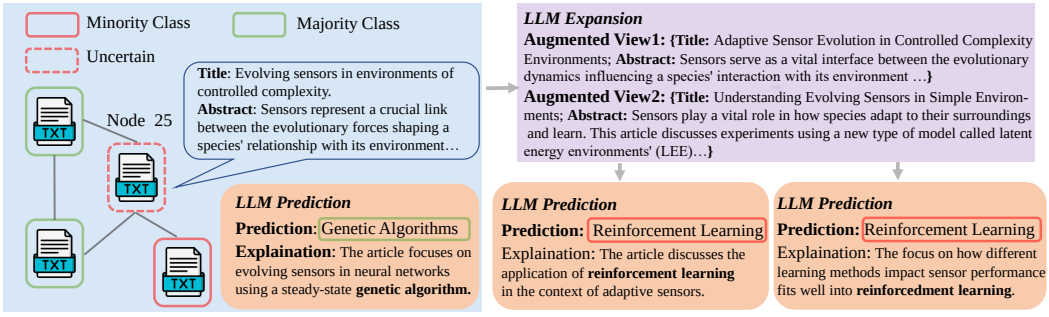


Figure 4: Case study of node 25 from the Cora dataset under an imbalance ratio of 10.

J Computational Cost

In this work, we propose TRACI, a method that integrates Graph Neural Networks (GNNs) with Large Language Models (LLMs) for graph clustering under class imbalance scenarios. The computational cost associated with the use of LLMs mainly arises from three components: the first is *Data Expansion*, where the LLM is used to generate augmented textual views for nodes; the second, termed *Group Mixup*, leverages the LLM to compute semantic correlation scores between texts within the same group to enhance contextual representations; and the third, referred to as *Ranking Guidance*, involves using the LLM to predict the most likely cluster assignment, thereby providing

feedback to guide the GNN. The total running time of TRACI consists of the time required for LLM inference on queries and the GNN's execution time when incorporating LLM-derived feedback.

To comprehensively evaluate the efficiency of TRACI, we report both its computational cost and runtime, as summarized in Table 10, under imbalance ratios $\rho = 10$ and $\rho = 20$. The computational cost is calculated based on the token-level pricing of GPT-4o-mini for both input and output tokens, while the runtime is estimated using its per-minute rate limit. As expected, larger datasets incur higher cost and longer runtime, which may hinder scalability to extremely large graphs. Notably, the *Expansion* step incurs no additional cost or runtime, as the construction of datasets with varying imbalance ratios allows it to reuse a subset of nodes from the version with a lower ρ, thus making it computationally free.

Table 10: Computational cost and Running time of TRACI on datasets under imbalanced settings ($\rho = 10$ and 20).

Imbalance	Datasets	Computational Cost ($)				Running Time (min)				
		Data Expansion	Group Mixup	Ranking Guidance	Total	Data Expansion	Group Mixup	Ranking Guidance	GNN	Total
$\rho = 10$	**Cora**	0.88	1.30	0.13	2.32	5.98	5.10	2.72	1.20	15.01
	CiteSeer	0.71	1.05	0.08	1.84	4.66	4.16	1.73	1.50	12.05
	WikiCS	7.26	8.74	0.56	16.56	66.66	45.30	13.86	3.23	129.05
	PubMed	6.55	8.12	0.13	14.81	48.27	38.15	2.73	1.72	90.86
$\rho = 20$	**Cora**	0.00	1.15	0.09	1.23	0.00	4.45	1.82	1.12	7.38
	CiteSeer	0.00	0.90	0.04	0.95	0.00	3.59	0.98	1.50	6.07
	WikiCS	0.00	7.47	0.37	7.84	0.00	38.61	9.43	3.97	52.01
	PubMed	0.00	7.35	0.12	7.48	0.00	34.37	2.63	1.67	38.67

K LLM USAGE CLARIFICATION

In this study, we use a large language model (LLM) solely to detect grammatical errors and improve sentence clarity. No content is generated automatically beyond these language edits, and all suggested modifications are carefully reviewed by the authors. All scientific aspects, including research hypotheses, experimental design, data analysis, and conclusions, are fully conceived and verified by the authors.