LANO: LARGE LANGUAGE MODELS AS ACTIVE ANNOTATION AGENTS FOR OPEN-WORLD NODE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Node classification is a fundamental task in graph learning. While Graph Neural Networks (GNNs) have achieved remarkable success in this area, their effectiveness relies heavily on large amounts of high-quality labels, which are costly to obtain. Moreover, GNNs are typically developed under a closed-world assumption, where all nodes belong to a fixed set of categories. In contrast, real-world graphs follow an open-world setting, where newly emerging nodes often stem from out-of-distribution (OOD) classes, making it challenging for GNNs to generalize. Motivated by the strong zero-shot reasoning and generalization ability of Large Language Models (LLMs), we propose LANO (LLMs as Active Annotation Agents for Open-World Node Classification). Our framework first aligns GNN representations with LLM token embeddings via instance-aware and feature-aware self-supervised learning, enabling LLMs to serve as zero-shot predictors for graph tasks. LANO then employs an influence- and uncertainty-driven strategy to select the most representative nodes and leverages LLMs for cost-effective pseudo-label generation. To suppress the spread of inaccurate labels and mitigate labeling bias, a soft feedback propagation mechanism disseminates bias-reduced pseudo labels to neighboring nodes with label decay mechanism, followed by iterative GNN optimization. Extensive experiments on multiple benchmarks demonstrate that LANO consistently outperforms popular baselines, showcasing the great potential of LLMs as active annotation agents for advancing open-world graph learning.

1 INTRODUCTION

Node classification is one of the most typical research directions in graph analysis (Xiao et al., 2022), with broad applications in citation network, amazon networks, and recommender systems. Under the closed-world assumption, graph neural networks (GNNs) have achieved remarkable success in this task (Wang et al., 2024b). Despite their effectiveness, GNN-based models face several inherent limitations. *First*, they are notoriously label-hungry—their performance heavily relies on abundant high-quality labeled, as shown in Figure 1, which is often costly and labor-intensive to obtain (Chen et al., 2023). *Second*, most existing models assume that labeled and unlabeled nodes come from the same set of predefined categories. However, this assumption rarely holds in real-world open-world scenarios, where newly added nodes may belong to entirely novel, out-of-distribution (OOD) categories. As a result, models trained solely on seen classes struggle to generalize to unseen categories, severely restricting their applicability in open-world graph learning (Wang et al., 2024b).

To address this challenge, prior work has explored OOD detection and open-world learning on graphs. Energy-based approaches replace softmax confidence with energy functions to distinguish in-distribution (ID) from OOD nodes (Liu et al., 2020). Other efforts, such as ORCA (Cao et al., 2021), design joint objectives for classification and clustering to progressively discover novel categories, while OODGAT (Song & Wang, 2022) explicitly models interactions between ID and OOD nodes via attention. Although these methods can mitigate misclassification and detect unknown nodes, their performance often degrades significantly under distributional shifts, limiting their generalization capacity in truly open-world environments (Li et al., 2022).

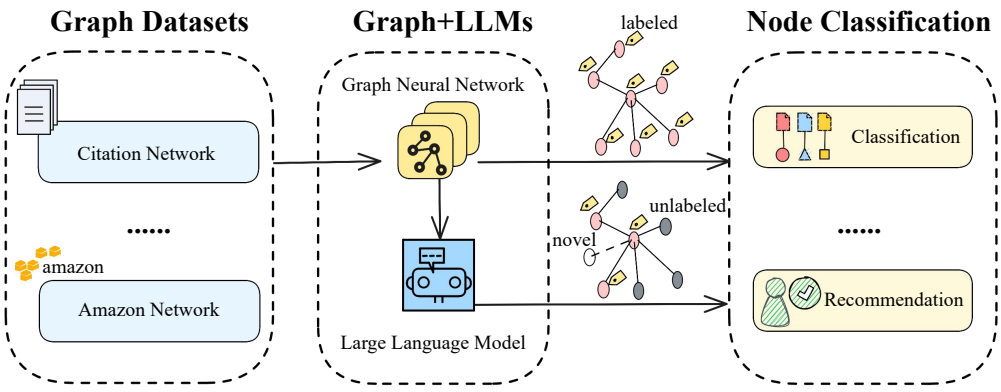


Figure 1: Motivation of this work. Traditional GNNs rely on abundant labeled data and closed-world assumptions, while real-world open-world graphs face label scarcity and the emergence of OOD nodes. Faced with such challenges, the integration of GNNs and LLMs with strong generalization capabilities demonstrates success in node classification.

Recent advances in Large Language Models (LLMs) shed light on addressing the challenge. LLMs exhibit remarkable zero-shot ability, often achieving strong performance without requiring labeled data (Chen et al., 2023). Thus as illustrated in Figure 1, LLMs provide a promising direction to alleviate both label scarcity and OOD generalization issues. Compared with costly human annotation, LLM-assisted labeling substantially reduces supervision cost. However, directly applying LLMs to graph-based node classification remains challenging: (1) *Structure awareness*: LLMs are not inherently designed to capture the relational and structural information present in graphs (Wang et al., 2024a); (2) *Node selection*: annotating all nodes with LLMs is infeasible, making it essential to identify the most representative nodes for labeling; and (3) *Label reliability*: LLM-generated pseudo-labels are susceptible to hallucinations and biases, requiring mechanisms that can mitigate noise while amplifying their benefits for GNN training (Sheng et al., 2025).

In this paper, we propose **LANO**, a novel framework that leverages LLMs as active annotation agents for open-world node classification. Specifically, our method first employs instance-aware graph learning to learn embeddings from unlabeled nodes, and introduces feature-aware self-supervised alignment to map GNN representations into the LLM token embedding space, thereby enabling LLMs to serve as zero-shot predictors for graph tasks. To further reduce annotation cost, LANO computes node influence and uncertainty to select the most representative nodes for LLM labeling. The obtained pseudo-labels are then propagated to neighboring nodes via a soft label propagation mechanism with label decay, which not only improves efficiency but also mitigates bias in pseudo-labeling. Finally, GNN training is iteratively refined using the enriched supervision, enhancing performance in both ID and OOD settings.

Our main contributions are summarized as follows: (1) *New Perspective*: We introduce a novel perspective that leverages LLMs with strong zero-shot abilities as active annotation agents to address label scarcity and OOD challenges in open-world node classification. (2) *New Framework*: We propose LANO, which integrates GNN–LLM representation alignment, influence- and uncertainty-based node selection, and bias-reduced soft label propagation into a unified framework to iteratively optimize GNN training. (3) *Experiments*: Extensive experiments across multiple datasets demonstrate that our framework consistently outperforms strong baselines, achieving extraordinary results in open-world node classification.

2 Preliminaries

Notations. We define the graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and $\boldsymbol{X}$ denotes the initial node features. Let $|\mathcal{V}| = N$ be the total number of nodes. The node set $\mathcal{V}$ is partitioned into a labeled subset $\mathcal{V}_l$ and an unlabeled subset $\mathcal{V}_u$. The classes of labeled nodes are denoted as $\mathcal{C}_l$ (seen classes), while the classes of unlabeled nodes are denoted as $\mathcal{C}_u$, which may

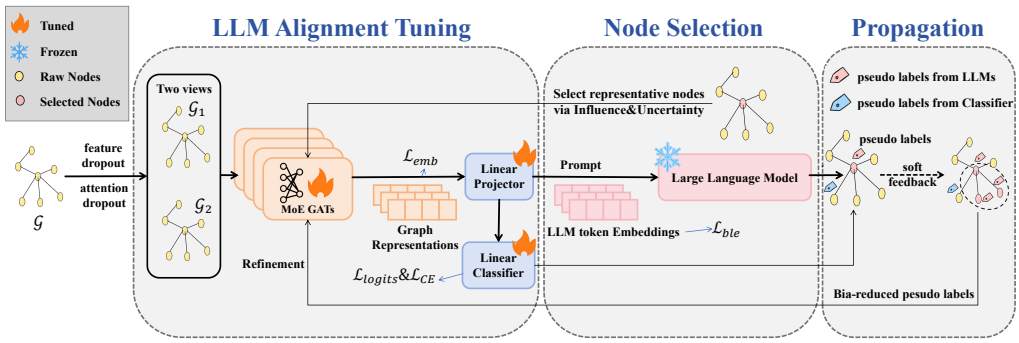


Figure 2: The overall LANO framework. After aligning GNN representations with LLM embeddings, LANO selects the most representative nodes for LLM annotation based on influence and uncertainty, propagation, thereby generating pseudo-labels. These pseudo-labels are further propagated through soft feedback, and the debiased labels are ultimately leveraged to iteratively optimize the training of the GNN.

also contain novel classes. Thus, nodes in $\mathcal{V}_u$ can belong either to $\mathcal{C}_l$ or to unseen classes in $\mathcal{C}_u$, i.e., $\mathcal{C}_l \cap \mathcal{C}_u \neq \emptyset$. The adjacency matrix $\boldsymbol{A}$ satisfies $\boldsymbol{A}[i,j] = 1$ if an edge exists between nodes i and j. The set of manual labels for nodes in $\mathcal{V}_l$ is denoted by $\mathcal{Y}_l = \{y_i \mid v_i \in \mathcal{V}_l, y_i \in \mathcal{C}_l\}$.

Graph Neural Networks. Graph Neural Networks (GNNs) are widely used for learning representations of nodes in graph-structured data. The central idea is to iteratively aggregate information from a node's neighborhood to capture both structural and feature dependencies. Formally, a generic GNN layer is given by:

$$\boldsymbol{h}_v^{l+1} = update\big(\boldsymbol{h}_v^l,\ aggregate(\{\boldsymbol{h}_u^l \mid u \in \mathcal{N}_v\})\big), \tag{1}$$

where $\boldsymbol{h}_v^l$ is the representation of node v at the l-th layer, and $\mathcal{N}_v$ is its neighborhood. The $aggregate$ function summarizes information from neighbors (e.g., mean, sum, or attention-based weighting), while $update$ refines the representation, often via multilayer perceptrons (MLPs) (Sheng et al., 2025). By stacking multiple layers, nodes can capture higher-order structural information. After L layers, the final representation $\boldsymbol{h}_v^L$ is obtained, which can be applied to downstream tasks.

Problem Definition. We investigate the task of node classification in an open-world graph setting (Wang et al., 2024b). Given the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \mathcal{Y}_l)$, the node set $\mathcal{V}$ consists of a labeled subset $\mathcal{V}_l$ with labels $\mathcal{Y}_l$, and an unlabeled subset $\mathcal{V}_u$. Among them, the unknown labels can be represented as $\mathcal{Y}_u = \{y_i \mid v_i \in \mathcal{V}_u, y_i \in \mathcal{C}_u\}$. The objective is to learn a function

$$\mathcal{F} : \mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \mathcal{Y}_l) \longrightarrow \mathcal{Y}_u, \tag{2}$$

such that nodes belonging to $\mathcal{C}_l$ are accurately assigned to their corresponding seen classes, while nodes from $\mathcal{C}_u$ are not only detected as unknown but further distinguished and categorized into their respective novel classes. This formulation extends traditional closed-world node classification by requiring the model to handle both recognition and classification of previously unseen categories within the same unified framework.

3 Method

We propose a novel semi-supervised graph learning framework, **LANO**, which leverages LLMs as active agents to facilitate open-world node classification. The framework consists of three key modules: (1) *Graph Self-supervised Learning for LLM Alignment*, (2) *Influence and Uncertainty Maximization-aware LLM Annotation*, and (3) *Learning with Bias-Reduced Pseudo Labels*. An overview of the framework is illustrated in Figure 2.

3.1 Graph Self-supervised Learning for LLM Alignment

In **LANO**, GNNs encoder is used to generate node representations, while an LLM, owing to its strong generalization ability, serves as a zero-shot predictor for node classification (Wang et al.,

2024a). However, LLMs cannot directly process graph-structured data. To bridge this gap, we propose a self-supervised alignment scheme that maps GNN representations into the LLM token embedding space. Our approach combines *instance-aware* graph learning and *feature-aware* alignment to capture both structural invariance and semantic compatibility.

Instance-aware Graph Learning. To ensure robustness of node embeddings against structural perturbations, we introduce instance-aware graph learning to enhance structural representation capacity. We first generate node representations via GNNs, adopting a mixture-of-experts (MoE) architecture where multiple experts are combined, each specialized in capturing distinct structural patterns. We then construct two perturbed views of the graph via the feature-based random dropout mechanism of GAT alongside the attention head random dropout mechanism, denoted as $\mathcal{G}_1$ and $\mathcal{G}_2$, for contrastive learning. Formally, given the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{A})$, we generate two randomly perturbed views $\mathcal{G}_1 = (\tilde{\boldsymbol{A}}_1, \tilde{\boldsymbol{X}}_1)$ and $\mathcal{G}_2 = (\tilde{\boldsymbol{A}}_2, \tilde{\boldsymbol{X}}_2)$. Each view is encoded by the GNN to produce node embedding matrices:

$$U_* = f_{GNN}(\tilde{\boldsymbol{A}}_*, \tilde{\boldsymbol{X}}_*) \in \mathbb{R}^{N \times d}, * \in \{1, 2\}, \tag{3}$$

where d is the dimension size of node representations. For node v_i, we denote its embeddings from the two views as $\boldsymbol{u}_i$ and $\boldsymbol{u}'_i$, respectively. Representations of the same node under different views are regarded as a positive pair, while those of different nodes are treated as negative pairs. The model learns discriminative embeddings in an unsupervised manner by maximizing the consistency between positive pairs and minimizing the similarity between negative pairs. We adopt a contrastive objective that encourages representations of the same node across views to be close, while pushing apart different nodes. The corresponding loss function is defined as:

$$\mathcal{L}emb = -\log \frac{\exp(sim(\boldsymbol{u}i \cdot \boldsymbol{u}'i/\tau))}{\sum k = 1^N \mathbf{1}_{[k \neq i]} \exp(sim(\boldsymbol{u}i \cdot \boldsymbol{u}'k/\tau))} \tag{4}$$

where $sim(\cdot)$ denotes cosine similarity, τ is the temperature parameter, and $\mathbf{1}_{[k \neq i]}$ is an indicator function that takes the value 1 if $k \neq i$, and 0 otherwise.

Feature-aware LLM Alignment Tuning. To bridge the gap between GNN node representations and the semantic space of LLM token embeddings, the goal is to ensure that when GNN outputs are converted into a sequence of token embeddings and provided as prompts, the LLM can perform zero-shot reasoning. This is achieved in two steps: first, a feature-aware contrastive alignment is conducted to align the feature axes (columns) of the GNN with the LLM token space; second, a linear projector is trained to map the GNN's central node representation into K token embeddings. The projector is fine-tuned for alignment while keeping the LLM frozen. Through this Feature-aware LLM Alignment Tuning, GNN outputs are directly adapted to the LLM embedding space, thereby improving generalization across tasks and datasets.

$$\mathcal{L}_{ble} = -\log \frac{\exp(\text{sim}(\boldsymbol{m}_i, \boldsymbol{n}_i)/\tau)}{\sum_{k=1}^{d} \mathbf{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{m}_i, \boldsymbol{n}_k)/\tau)}, \tag{5}$$

where $\boldsymbol{m}_i$ and $\boldsymbol{n}_i$ are the i-th feature vectors from two augmented embeddings U_1 and U_2. Then the overall self-supervised objective is a weighted combination:

$$\mathcal{L}_{SCL} = \sum_{i=1}^{N} (\lambda_1 \mathcal{L}_{emb} + \lambda_2 \mathcal{L}_{ble}) + \mathcal{L}_{logits} \tag{6}$$

where λ_1, λ_2 balance the two terms. $\mathcal{L}_{logits}$ denotes the supervised contrast loss introduced on the output logits of the classification layer, which is used to directly improve the differentiation of the final prediction results.

Consequently, neither the GNN nor the LLM requires task-specific fine-tuning. Instead, by mapping graph representations into token embeddings via the projector and feeding them into a unified LLM prompt template, the framework enables cross-task and cross-dataset reasoning in a zero-shot manner (Wang et al., 2024a).

3.2 Influence Maximization-aware LLM Annotation

High-quality annotations are crucial for graph learning, yet manual labeling is prohibitively expensive. In scenarios with scarce and noisy labels, it is therefore essential to select the most informative

nodes for annotation, striking a balance between performance improvement and labeling cost. To this end, we propose to leverage node influence for guiding LLM-based annotation, and further design task-specific prompts that enable LLMs to effectively capture graph information for zero-shot node classification.

Influence Estimation for Node Selection. Building on reliable influence-based active learning (Zhang et al., 2021), we adopt a joint strategy that combines uncertainty estimation with influence maximization to identify representative nodes for annotation. We first compute the global uncertainty of each node by considering its relation to cluster centroids obtained via k-means and neighbor propagation. Specifically, a Student-t distribution is used to assign each node a probability over clusters, and the entropy of this distribution quantifies the uncertainty. This uncertainty is further refined via neighbor aggregation:

$$u(v_i) \;=\; u(v_i) \;+\; \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i)} \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j) \cdot u(v_j), \tag{7}$$

where $\mathcal{N}(v_i)$ denotes the k-nearest neighbors of v_i, $\boldsymbol{z}_i$ is the node representation, and $\text{sim}(\cdot, \cdot)$ is a similarity function. Next, we estimate the influence of each node by considering multi-hop propagation paths (self, 1-hop, and 2-hop neighbors). Following RIM (Zhang et al., 2021), the influence score from node v_i to v_j after k propagation steps is defined as

$$Q(v_j, v_i, k) \;=\; r_{v_i} \cdot I(v_j, v_i, k), \tag{8}$$

where r_{v_i} denotes the influence quality of node v_i, and $I(\cdot)$ measures the reliable influence of v_i on v_j after k-step feature/label propagation. Combining global uncertainty and influence, the selection score for node v_i is given by

$$\text{score}(v_i) = u(v_i) \cdot \sum_{v_j \in \mathcal{N}_k(v_i)} Q(v_j, v_i, k), \tag{9}$$

and the top-K nodes are selected for LLM annotation:

$$S = \arg \text{topK}_{v \in \mathcal{V}} \, \text{score}(v). \tag{10}$$

Prompt Engineering for Annotation. The prompts are structured into three components: task information, graph information, and output rules. The task information is expressed as a question + option set (Wang et al., 2024a); the graph information consists of node graph token embeddings plus a title; and the output rules specify the classification result and confidence. For example: *Your task: Classify the target node into predefined categories or detect a new category. Predefined categories (represented by semantic tokens): category 1: ... Target node: ... Rules: Known/New/Uncertain Category and Confidence Level.* The complete prompt design is provided in Appendix D.

After identifying the nodes requiring annotation by the LLM, we first align them to the LLM's semantic space and construct carefully designed prompts as input. The output from the LLM falls into three categories: (1) If the LLM outputs a seen class label, we assign a soft label as a confidence-weighted one-hot vector; (2) If the LLM outputs an unseen class, we use the classification head's predicted distribution, also weighted by confidence; (3) If the output is invalid or malformed, the result is discarded. We retain only valid annotations and discard outdated nodes to ensure high-quality supervision in the subsequent training stage.

3.3 Learning with Bias-Reduced Pseudo Labels

Although LLMs can provide pseudo-labels via carefully designed prompts, these annotations are not guaranteed to be correct. To mitigate the risk of propagating noisy labels, we introduce two mechanisms: *soft feedback propagation* and *bias-reduced pseudo label concordance*.

Soft Feedback Propagation. To efficiently expand the utility of LLM-generated pseudo-labels, we propagate them to structurally and semantically similar neighbors. However, directly propagating hard labels risks amplifying errors. We therefore adopt a soft feedback propagation strategy, where only high-confidence outputs from LLMs are allowed to propagate. Specifically, for each node v_i^s in the selected set S, its LLM-generated pseudo-label y_i^{LLM} is propagated to uncertain neighbors v_j based on feature similarity:

$$s_j = (1 - \text{sim}(z_j, z_i^s)) \cdot s_j \;+\; \text{sim}(z_j, z_i^s) \cdot y_i^{LLM}, \tag{11}$$

where s_j denotes the soft prediction vector of node v_j, and $\text{sim}(\cdot,\cdot)$ measures the similarity between embeddings z_j and z_i^s.

The propagated pseudo-label for v_j is then assigned as

$$y_j^{prop} = \begin{cases} y_i^{LLM}, & \text{if } \operatorname{argmax}(s_j) = y_i^{LLM}, \\ -1, & \text{otherwise}, \end{cases} \tag{12}$$

where y_i^{LLM} is a one-hot vector with the predicted class set to 1. In this way, pseudo-labels are propagated only when the updated prediction of v_j is consistent with the LLM-assigned class, reducing the risk of error amplification.

Bias-reduced Pseudo Label Concordance. Since the model is not pretrained, early-stage pseudo-labels are often noisy. To alleviate bias accumulation, we introduce a label decay mechanism that gradually attenuates the influence of outdated pseudo-labels. At each iteration, a batch of L new pseudo-labels is generated by LLMs, and only the most recent M labels are preserved for propagation together with the original labeled set. The historical pseudo-labels are maintained across iterations but scaled down by a decay factor $\gamma < 1$:

$$\hat{\mathcal{Y}}^t = \gamma \cdot \hat{\mathcal{Y}}^{t-1} + \hat{\mathcal{Y}}^t_{\text{new}}, \tag{13}$$

where $\hat{\mathcal{Y}}^t$ denotes the aggregated pseudo-label matrix at iteration t. This decay ensures that earlier noisy annotations gradually vanish, while recent high-quality LLM feedback dominates the training process. Together, soft feedback propagation and bias-reduced concordance mitigate the risks of error amplification and label bias, enabling the model to effectively exploit LLM annotations in an open-world setting (Liang et al., 2024; Wang et al., 2024b).

3.4 Overall Optimization

To optimize GNN training, we incorporate bias-reducing pseudo labels into iterative GNN training. Our loss function during training primarily includes: (1) Supervised Contrastive Loss ($\mathcal{L}_{SCL}$), designed to better separate known and unknown classes. (2) Cross-Entropy Loss ($\mathcal{L}_{CE}$), used to learn valuable synthetic labels. We update the model using the following overall loss formula:

$$\mathcal{L}_{ours} = \eta \mathcal{L}_{CE} + \mathcal{L}_{SCL} \tag{14}$$

where η is the scaling factor. Contrastive loss is applied to GNN-output embeddings, projected LLM embeddings, and classification layer logits. This encourages similar samples (different perspectives from the same node) to cluster closely in embedding space while keeping dissimilar samples apart. The $\mathcal{L}_{CE}$ calculated from classification head logits is used for labeled training nodes, ensuring the model correctly classifies known categories.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method on several commonly used benchmark datasets for node classification tasks, including Citeseer (Kipf, 2016), Amazon_photos (Shchur et al., 2018), Amazon Computers (Shchur et al., 2018), Coauthor_CS (Shchur et al., 2018) and Coauthor_Physics (Shchur et al., 2018). More detailed statistics of the datasets are provided in Appendix B.

Evaluation Metric. Under the open-world setting, node categories are divided into seen and unseen classes. A prediction is considered correct only if the model assigns the node to its ground-truth label. For LLM-based annotation, we additionally provide two options—decidable and undecidable. A prediction is counted as correct if the LLM selects decidable and its output matches the true label; if undecidable is chosen, the annotation is regarded as invalid and excluded from accuracy computation. The more details of the metric is given in Appendix C. All experiments are repeated 10 times with different splits, and the reported accuracy is averaged across runs.

Baselines. We compare our method against a broad set of baselines applicable to open-world node classification. These include open-world node classification algorithms OODGAT (Song & Wang,

Table 1: Performance comparison on different datasets under open-world settings with test accuracy (%). The best results in each column are highlighted in bold and pink, the second-best results in each column are highlighted in yellow.

Method	Citeseer			Coauthor_CS			Coauthor_phy			Amazon_photos			Amazon_computers		
	all	seen	novel	all	seen	novel	all	seen	novel	all	seen	novel	all	seen	novel
OODGAT	46.4	56.9	37.5	68.1	68.8	65.6	68.3	69.4	62.5	63.0	71.1	54.5	61.3	63.3	55.9
OpenWGL	62.4	71.0	54.2	58.6	67.1	50.3	73.3	85.0	68.1	71.8	74.8	69.3	57.6	65.9	44.6
ORCA-ZM	58.3	72.8	44.4	75.0	74.2	73.5	64.7	81.1	55.9	74.6	89.9	58.2	63.8	73.7	52.6
ORCA	58.2	68.0	49.0	73.9	81.6	68.3	66.2	84.8	58.2	76.2	87.1	64.9	60.9	67.8	53.7
SimGCD	61.5	70.6	53.4	71.2	84.2	61.2	60.9	81.1	52.8	80.5	90.0	70.8	61.9	73.8	50.3
OpenLDN	62.3	73.9	51.6	68.4	80.6	60.3	62.2	72.4	57.2	80.9	90.6	71.9	63.3	76.5	51.8
OpenCon	68.8	**75.0**	62.1	73.5	83.4	67.5	65.8	**95.0**	55.4	82.6	**92.1**	72.8	62.3	74.9	51.2
OpenCon	66.7	73.7	60.0	71.0	81.9	64.8	62.6	83.8	54.4	82.9	87.9	78.1	59.4	69.0	53.2
InfoNCE	68.1	70.7	65.2	72.2	72.8	72.7	60.6	58.1	60.2	76.3	78.5	75.1	56.1	51.3	59.1
InfoNCE+SupCon	68.1	71.9	64.1	75.6	80.3	72.0	56.3	52.5	58.9	72.4	75.1	71.0	60.5	59.7	59.8
InfoNCE+SupCon+CE	68.1	73.6	62.6	76.4	80.5	72.9	55.8	54.7	56.5	74.4	77.1	73.0	62.8	**79.4**	56.1
OpenIMA	68.1	71.8	64.3	77.1	78.3	75.9	78.0	93.6	72.2	83.6	89.9	77.3	67.8	77.8	59.0
ours(LANO)	**70.2**	73.8	**66.2**	**83.4**	**85.2**	**80.7**	**80.2**	79.6	**72.6**	**84.3**	86.2	**83.1**	**70.3**	70.4	**70.2**

2022) and OpenWGL (Wu et al., 2021), as well as baseline methods for end-to-end open-world semi-supervised learning, namely ORCA (Cao et al., 2021), ORCA-ZM (Cao et al., 2021), SimGCD (Wen et al., 2023), OpenLDN (Rizve et al., 2022), OpenCon (Sun & Li, 2022), InfoNCE (Oord et al., 2018), and OpenIMA (Wang et al., 2024b). More detailed descriptions of these methods can be found in Appendix E.

Implementation Details. Our model builds upon the architecture of OpenIMA by extending its original GNN backbone. Specifically, we treat a single GAT network as one head responsible for encoding a single token, and the number of heads is determined by the number of tokens required as input to the LLM. To capture diverse structural representations of the graph, different heads are designed with variations in hop size, hidden dimensionality, number of attention heads, dropout rate, and whether residual connections are applied. To align the input dimension of the projection layers, we set the output dimension of all heads to 256. We employ Adam as the optimizer with a batch size of 4096. Training is conducted over 40 epochs with a learning rate of 0.004. For detailed model and parameter configurations, please refer to Appendix F.

4.2 Results and Analysis

Table 1 presents the classification accuracy of our method and various baselines on both seen and unseen classes under the open-world setting. Overall, across most datasets and evaluation scenarios, our approach outperforms all competing methods in terms of overall accuracy and unseen-class recognition, while maintaining strong competitiveness on seen-class classification.

We attribute these performance gains to several key factors. *First*, incorporating LLMs as pseudo-label generation agents aligns the semantic information encoded by GATs and leverages the LLMs' semantic discrimination capability, enabling more efficient and accurate annotation of unseen classes and providing more valuable supervision signals for training. *Second*, by combining node influence and uncertainty metrics, we selectively propagate pseudo-labels only to the most representative unlabeled nodes, effectively reducing noise interference and enhancing the reliability and diversity of pseudo-labels.*Third*, the original framework relies solely on selecting the top $\rho\%$ of nodes closest to cluster centroids as pseudo-labels, which can lead to label oscillations and unstable training when decision boundaries are still ambiguous. To address this, we introduce soft feedback propagation and label decay mechanisms, mitigating the spread of biased pseudo-labels and reducing the negative impact of oscillations. *Finally*, as suggested by theoretical insights from the training framework, the introduction of LLMs partially alleviates the imbalance of supervision signals between seen and unseen classes, further enhancing overall classification performance.

4.3 Ablation Studies

We further conduct ablation experiments to investigate the contribution of each component of our framework. Specifically, we design six variants: 1) Removing LLM-assisted pseudo-label generation; 2) Treating projection heads as fixed rather than learnable parameters; 3) Omitting the un-

Table 2: Ablation studies by overall test accuracy (all, seen, novel). The best results in each column are highlighted in bold.

Method	Citeseer			Coauthor_CS		
	all	seen	novel	all	seen	novel
ours	**70.2**	73.8	66.2	**83.4**	85.2	80.7
w/o LLMs	68.7	71.7	64.0	77.4	83.5	71.0
w/o projector	69.5	71.7	66.2	82.6	**85.9**	78.4
w/o influence	69.1	**74.8**	63.7	82.7	82.3	**81.8**
w/o uncertainty	68.2	66.2	**69.4**	80.1	79.8	81.6
variant1	67.1	69.3	65.7	68.7	70.8	67.5
variant2	67.3	68.7	66.3	81.4	82.2	80.0

certainty measure in high-value node selection; 4) Omitting the maximum activation criterion in high-value node selection; 5) Replacing informed selection with random node sampling (variant1); 6) Replacing the multi-head architecture with a single head (variant2). Based on these six variants, we perform repeated experiments on the Citeseer and Coauthor_CS datasets, with the results summarized in Table 2. The experiments reveal that excluding LLM-assisted pseudo-label generation leads to a substantial drop in overall classification accuracy. A further breakdown between seen and unseen classes shows that performance on seen classes remains largely unaffected, whereas the accuracy on unseen classes degrades significantly. This highlights the critical role of the LLM in enhancing the recognition of unseen classes under the open-world setting. Regarding high-value node selection, removing the uncertainty measure causes a larger performance decline compared to removing the maximum activation criterion, suggesting that uncertainty is more effective in identifying valuable unlabeled nodes. Moreover, any metric-based selection strategy yields a clear advantage over random sampling. For the multi-head design, replacing it with a single-head structure results in performance degradation, as the absence of multi-view semantic information hampers the model's ability to learn well-defined decision boundaries.

4.4 Hyper-parameters Sensitivity Analysis

We further investigate the impact of hyperparameters on the performance of the proposed method. Specifically, we focus on three key hyperparameters: (1) τ_{LLM}, the temperature used in the supervised contrastive loss applied to features projected into the low-dimensional LLM embedding space; (2) n_{token}, the number of heads in the multi-head GNN, corresponding to the number of encoded tokens; and (3)ρ, the ratio of cluster-labeled nodes assigned as pseudo-labels. For a detailed sensitivity analysis of the hyperparameters n_{token} and ρ, see Appendix G. In our experiments, we adopt a single-factor control strategy, i.e., when analyzing one hyperparameter, the others are fixed at their optimal values.

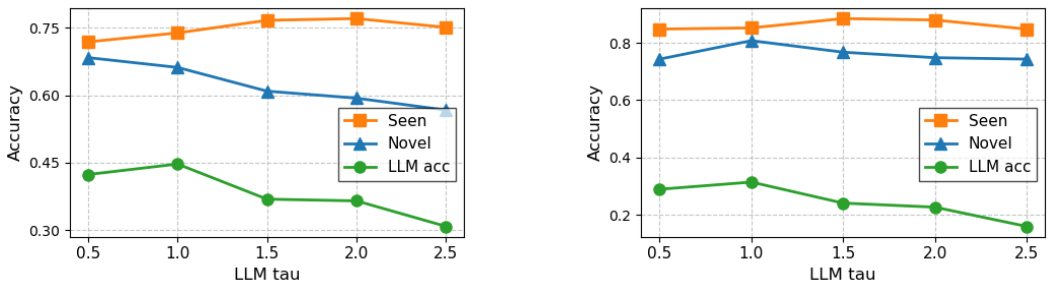


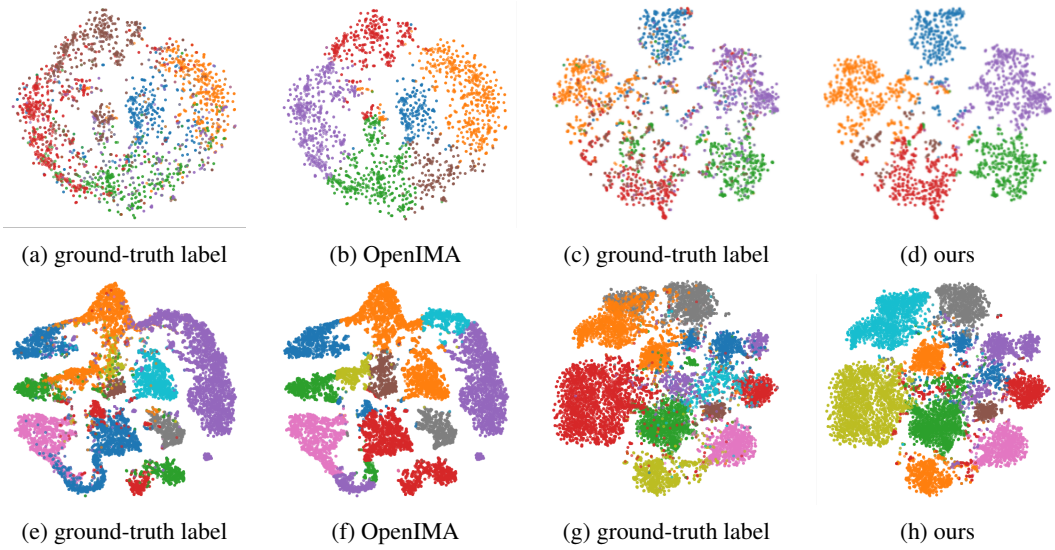
Figure 3: Hyper-parameters sensitivity analysis of τ_{LLM} on Citeseer and Coauthor_CS datasets.

Figure 3 illustrates the effect of τ_{LLM} on the classification accuracy of seen classes, unseen classes, and LLM annotation. The overall trend exhibits an **increase-then-decrease** pattern: as τ_{LLM} increases (i.e., the similarity computation becomes more relaxed), there exists an optimal point for distinguishing between same-class and different-class nodes. At moderate values, τ_{LLM} effectively

pulls together nodes of the same class while pushing apart nodes of different classes, thereby enhancing discriminative capability. However, when τ_{LLM} becomes too large, the similarity distribution is overly smoothed, weakening the constraints of contrastive learning. Consequently, the model's discriminative power decreases, and both the classification accuracy for seen and unseen classes as well as the reliability of LLM-generated pseudo-labels decline significantly.

4.5 VISUALIZATION

To further illustrate the effectiveness of our model, we present the visualization results of the proposed method compared with OpenIMA, along with the corresponding predicted categories and ground-truth labels in Figure 4. On the Citeseer dataset, the node distribution produced by OpenIMA is relatively scattered. While it reveals some degree of class separation, the clusters are neither compact nor well-defined. On the Coauthor_CS dataset, OpenIMA yields comparatively tighter clusters, yet the overall cluster boundaries remain indistinct. We attribute this to its pseudo-labeling strategy, which assigns labels only to the top-ρ% of nodes closest to cluster centroids. When decision boundaries are ambiguous, this approach tends to cause oscillations in pseudo-label assignments for the same node, leading to unstable supervision signals and consequently undermining both discriminative power and clustering quality. In contrast, our method integrates a multi-head architecture with LLM-assisted pseudo-label generation, further enhanced by soft labeling and label-smoothing updates. The visualizations clearly demonstrate that, across both datasets, nodes of the same class form compact and well-delineated clusters. These results suggest that our approach is capable of learning higher-quality and more discriminative graph representations.



(a) ground-truth label (b) OpenIMA (c) ground-truth label (d) ours

(e) ground-truth label (f) OpenIMA (g) ground-truth label (h) ours

Figure 4: The T-SNE visualizations of target node representations for the Citeseer and Coauthor_CS dataset, comparing a baseline model OpenIMA with our method (colors indicate classes).

5 CONCLUSION

In this work, we address the challenge of node classification under the open-world setting, where novel classes inevitably emerge beyond the scope of training labels. To tackle the limitations of conventional GNN-based approaches, we propose **LANO**, a novel framework that leverages LLMs as active annotation agents. Specifically, LANO aligns GNN representations with LLM token embeddings through instance- and feature-aware self-supervised learning, enabling LLMs to serve as zero-shot predictors for graph tasks. An influence- and uncertainty-driven node selection strategy is introduced to identify representative samples for annotation, while a soft feedback propagation mechanism effectively suppresse label noise and incorporate bias-reduced pseudo labels into iterative GNN training . Extensive experiments on multiple benchmarks demonstrate the effectiveness of LANO, highlighting the potential of integrating LLMs with GNNs for open-world graph learning.

REFERENCES

Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021.

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.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7328–7340, 2022.

Jinggui Liang, Lizi Liao, Hao Fei, Bobo Li, and Jing Jiang. Actively learn from llms with uncertainty propagation for generalized category discovery. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7838–7851, 2024.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

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

Mamshad Nayeem Rizve, Navid Kardan, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Openldn: Learning to discover novel classes for open-world semi-supervised learning. In *European Conference on Computer Vision*, pp. 382–401. Springer, 2022.

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

Zeang Sheng, Weiyang Guo, Yingxia Shao, Wentao Zhang, and Bin Cui. Llms are noisy oracles! llm-based noise-aware graph active learning for node classification. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2526–2537, 2025.

Yu Song and Donglin Wang. Learning on graphs with out-of-distribution nodes. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1635–1645, 2022.

Yiyou Sun and Yixuan Li. Opencon: Open-world contrastive learning. *arXiv preprint arXiv:2208.02764*, 2022.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Advances in Neural Information Processing Systems*, 37:5950–5973, 2024a.

Yanling Wang, Jing Zhang, Lingxi Zhang, Lixin Liu, Yuxiao Dong, Cuiping Li, Hong Chen, and Hongzhi Yin. Open-world semi-supervised learning for node classification. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 2723–2736. IEEE, 2024b.

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16590–16600, 2023.

Man Wu, Shirui Pan, and Xingquan Zhu. Openwgl: open-world graph learning for unseen class node classification. *Knowledge and Information Systems*, 63(9):2405–2430, 2021.

Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. Graph neural networks in node classification: survey and evaluation. *Machine Vision and Applications*, 33(1):4, 2022.

Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. Rim: Reliable influence-based active learning on graphs. *Advances in neural information processing systems*, 34:27978–27990, 2021.

A Related Work

A.1 Open-World Semi-Supervised Graph Learning

Open-world semi-supervised learning (OW-SSL) (Cao et al., 2021) on graphs aims to address the realistic scenario where unlabeled test nodes may belong to novel classes that are unseen during training. A representative method, ORCA (Cao et al., 2021), originally proposed in the vision domain, introduces an uncertainty-based adaptive margin to balance intra-class variance. Within an end-to-end framework, it jointly tackles both classification and clustering, enabling the discovery of novel categories in an open-world setting. More recently, OpenIMA (Wang et al., 2024b) has explored a more practical OW-SSL setting by introducing a two-stage training framework. Through contrastive learning and bias-reduced pseudo-labeling, OpenIMA mitigates the imbalance between seen and novel classes and improves classification accuracy. Despite these advances, existing OW-SSL approaches still face key challenges: (1) they often rely on extensive human annotations for initial supervision, (2) their performance can degrade under distributional shifts

A.2 Large Language Models for Graph Learning

Although LLMs are not inherently designed to capture relational structures in graphs (Wang et al., 2024a), recent studies have begun to explore their potential in graph reasoning tasks. The performance of such methods, however, strongly depends on graph encoding strategies, prompt engineering, and the structural properties of the input graph (Fatemi et al., 2023). For example, ALUP (Liang et al., 2024) investigates generalized category discovery (GCD) by integrating LLMs with active learning strategies, thereby improving the recognition of novel classes while maintaining reliable feedback and significantly reducing annotation costs. Similarly, DMA (Sheng et al., 2025) highlights that LLM-based annotation can be noisy and is sensitive to both dataset characteristics and the choice of LLM, suggesting the need for robust mechanisms to handle annotation variability. In summary, while open-world graph learning methods focus on OOD detection and novel class discovery, they remain constrained by annotation costs and distributional shifts. LLM-based approaches, on the other hand, offer strong generalization and zero-shot capabilities but suffer from structural limitations, annotation noise, and dependency on encoding strategies. These gaps motivate the design of a hybrid framework that integrates GNN-based representation learning with LLM-driven annotation to advance open-world node classification.

B Datasets

This section provides a more detailed introduction to the popular datasets commonly used for node classification, including Citeseer, Coauthor_CS and Coauthor_Physics, Amazon_Photos, and Amazon_Computers, as shown in Table 3.

In our experimental setup, we randomly select 50% of the classes in each dataset as known classes, while the remaining classes are treated as unknown classes. For each known class, 50 nodes are randomly sampled as the training set, another 50 nodes as the validation set, and the rest are used for testing. To ensure robustness, we generate 10 independent train/validation/test splits using 10 different random seeds.

C Evaluation Metric

LLM Labeling Accuracy. During the process of generating pseudo-labels with LLMs, we evaluate their consistency with the ground-truth labels. According to our prompt design rules, if a node is classified by the LLM as belonging to a visible class, the pseudo-label is considered correct only if it matches the ground truth. If a node is classified as belonging to an unseen class, we instead refer to the output of the classification head as its pseudo-label, and correctness is determined by whether

Table 3: Statistics of Datasets

Graph	Type	Nodes	Edges	Features	Class
Citeseer	Citation network	3327	4277	3703	6
Coauthor CS	Co-author network	18333	81894	6805	15
Coauthor Physics	Co-author network	34493	247962	8415	5
Amazon Photos	Amazon network	7650	119082	745	8
Amazon Computers	Amazon network	13752	245861	767	10

this label matches the ground truth. If the LLM outputs "unrecognizable," the labeling result is discarded and excluded from evaluation. All other cases—including assigning incorrect labels or producing non-standardized outputs—are regarded as labeling errors.

D PROMPTS

In this part, we show the complete prompt designed to query LLMs for annotations.

Prompt for Large Language Model Annotation in Open-World Node Classification

Your task is to classify the given target node into one of the predefined categories or determine if it belongs to a new category based on semantic similarity.
Please carefully read the following set of predefined categories, where each category is represented by a list of representative semantic tokens:

```
 { % for category in categories -%
category {{ category.id }}:  % for token in category.tokens
%{{token}}% endfor %;
% endfor
```

Now, here is the target node represented by a token sequence:

```
 % for token in target_tokens %{{token}}% endfor %;
```

When classifying the target node, please follow these rules:
1. **Match to Known Category:**
If the target node's semantics strongly align with one of the known category X (where X is a number), output the result in the format:
`[X][ConfidenceLevel]`
Confidence Levels:
`A`: $\geq 99\%$ confidence
`B`: $\geq 75\%$ confidence
`C`: $\geq 50\%$ confidence
`D`: $\geq 25\%$ confidence

2. **New Category Detection:**
If the target node's semantics are inconsistent with all known categories but indicate a novel and coherent class, return:
`[N][ConfidenceLevel]`

3. **Uncertain Classification:**
If the classification is ambiguous or insufficient evidence is available, return:
`[-1]`

Final Rule: Please only return the final classification label (`[X][ConfidenceLevel]`, `[N][ConfidenceLevel]`, `[-1]`). Do not output explanations or reasoning.

E BASELINES

This section introduces the baselines used for comparison with our method in the experiments, as summarized in Table 1. The details are as follows:

- **OODGAT**: Out-of-Distribution Graph Attention Network (OODGAT) is a GNN model that explicitly models the interactions among different types of nodes and separates in-distribution and out-of-distribution nodes during feature propagation.
- **OpenWGL**: A new paradigm for open-world graph learning, whose objective is not only to classify nodes belonging to visible classes correctly but also to classify nodes outside the known classes into the unseen category.
- **ORCA**: A model specifically designed to address open-set recognition under long-tailed distributions. Its main objective is to prevent the model from misclassifying rare unknown samples as rare known classes. ORCA without the margin mechanism is denoted as ORCA-ZM.
- **SimGCD**: A simple yet effective method for generalized category discovery (GCD), aiming to simultaneously recognize known classes and discover unknown classes.
- **OpenLDN**: OpenLDN employs a pairwise similarity loss to discover novel classes. Leveraging a bi-level optimization scheme, the pairwise similarity loss exploits available information from the labeled set to implicitly cluster samples of new categories while identifying samples from known categories.
- **OpenCon**: A method targeting open-world classification and novel class clustering by combining semi-supervised learning with contrastive losses. It determines the total number of classes, introduces a classification head, and optimizes using contrastive objectives tailored for labeled, unlabeled, and novel categories.
- **InfoNCE**: Noise Contrastive Estimation Loss is a widely used self-supervised loss function for representation learning. Rooted in information-theoretic principles, it learns model parameters by contrasting the similarity between positive and negative samples.
- **OpenIMA**: OpenIMA is designed for open-world semi-supervised node classification. It trains a classifier from scratch using unbiased pseudo-labels and contrastive learning, effectively mitigating intra-class imbalance and improving classification accuracy. Compared to many existing node classification approaches, OpenIMA demonstrates superior performance.

F IMPLEMENTATION DETAILS

Design of GNN Heads. All GNN heads adopt Graph Attention Networks (GAT) as the feature encoder. To enable multi-perspective node representation, we vary the hop number, hidden dimension, number of attention heads, dropout rate, and whether to apply residual connections across different heads. For compatibility with the projector, the output dimension of the final GAT layer is fixed to 256. During training, we employ the Adam optimizer with a weight decay of 1×10^{-4}. Since both feature and attention mechanisms involve dropout, we sample each input twice to construct positive pairs for contrastive learning.

Obtaining Low-dimensional LLM Embeddings. To efficiently leverage the semantic representations of LLMs under limited GPU memory, we perform dimensionality reduction on their high-dimensional embeddings. As the datasets used lack bag-of-words information, semantic textual features cannot be directly obtained. Instead, we encode node attributes as tokens and feed them into the LLM to obtain semantic embeddings. Taking QWen3-8B as an example, the embedding dimension typically exceeds 4000, which would incur prohibitive GPU memory costs if directly used for supervised contrastive loss. Therefore, we apply PCA to reduce embeddings to 1000 dimensions. The projection head then maps generated word vectors into this reduced space, where supervised contrastive loss is computed for semantic alignment.

Choice of LLM. Prior studies suggest that LLM performance saturates around 7B parameters, with marginal gains from larger models. Considering model scale and release time, we evaluate Mistral-7B, QWen3-8B, and Deepseek-r1-7B. Experiments show that Mistral-7B often outputs "uncertain" predictions with very low confidence, while Deepseek-r1-7B, due to its built-in chain-of-thought mechanism, generates unnecessarily long reasoning when fed non-standard token embeddings, leading to high inference cost. In contrast, QWen3-8B provides stable outputs with controllable reasoning and more efficient inference. Consequently, we adopt Qwen3-8B as the pseudo-label generator in subsequent experiments.

Soft Pseudo-labels from LLM. Since the input tokens fed into the LLM are multi-head encoded rather than complete natural language text, directly producing hard labels may introduce bias. To address this, we design the LLM outputs as "label + confidence level," a soft pseudo-label format analogous to a softmax distribution. This not only reflects predictive uncertainty but also better supports pseudo-label propagation and forgetting mechanisms, thereby improving overall robustness.

Experimental Hyperparameter Settings. When inheriting default hyperparameters from the framework, we set the scaling factor $\eta = 1$, temperature $\tau = 0.7$, and pseudo-label selection rate $\rho = 75\%$. An additional supervised contrastive loss is introduced for low-dimensional projection alignment. To prevent unstable training caused by large gradients, we apply a scaling factor of 0.025 to the GAT embeddings. Parameter search shows that as model complexity increases, stronger performance on the *amazon_computers* and *amazon_photos* datasets requires increasing η to 30 and slightly lowering τ, making the model more confident in pseudo-labeling while slowing down the forgetting rate to retain high-confidence pseudo-labels. In contrast, for *citeseer*, where validation accuracy is significantly lower, we adopt a stricter strategy by reducing the pseudo-label selection rate ρ and maintaining a higher forgetting rate, so as to mitigate the adverse impact of noisy pseudo-labels in early training.

G More Hyper-parameters Sensitivity Analysis

We conduct hyperparameter sensitivity analysis on the number of heads n_{token} and the ratio of cluster-labeled nodes assigned as pseudo-labels ρ.

The number of heads n_{token}. Figure 5 illustrates the impact of n_{token} on the accuracy of LLM annotations. The results show that as n_{token} increases, the accuracy of LLM annotations gradually improves. However, performance bottlenecks are observed on both *citeseer* and *coauthor_cs*: on *citeseer*, the optimal performance is achieved with approximately 3 tokens, whereas on *coauthor_cs*, a larger number of tokens is required to sufficiently capture semantic information. This discrepancy may be attributed to the smaller number of classes and nodes in *citeseer*, where fewer tokens are sufficient to cover node features, thus reaching the performance optimum earlier.

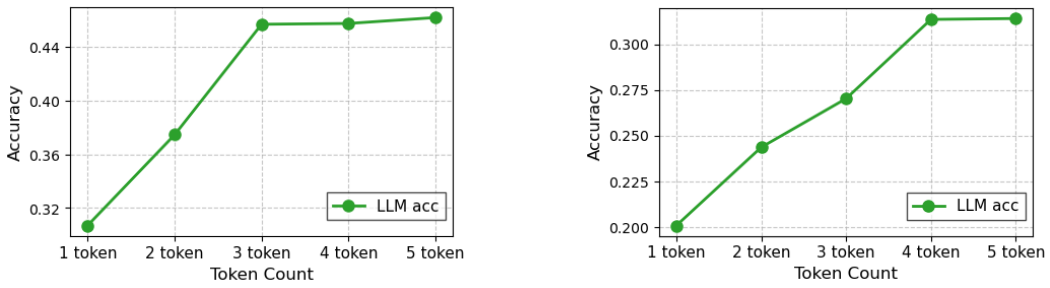


Figure 5: Hyper-parameters sensitivity analysis of n_{token} on Citeseer and Coauthor_CS datasets.

The ratio of cluster-labeled nodes assigned as pseudo-labels ρ. Figure 6 illustrates the impact of the pseudo-label ratio ρ on performance. On both *citeseer* and *coauthor_cs*, the accuracies of seen classes, unseen classes, and LLM annotations all achieve their optimum when ρ is set to 50% or 75%. However, as ρ continues to increase, the performance degrades, with the accuracy of LLM

annotations exhibiting a cliff-like drop. We attribute this phenomenon to the fact that an excessively large ρ leads the model to overconfidently assign incorrect pseudo-labels to nodes, which disrupts feature encoding, blurs the decision boundary, and directly results in a drastic decline in the reliability of LLM annotations.

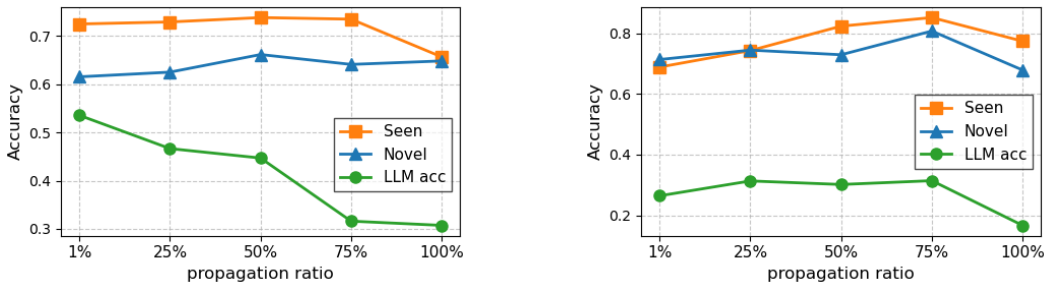


Figure 6: Hyper-parameters sensitivity analysis of ρ on Citeseer and Coauthor_CS datasets.

H Large Language Models Usage Statement

In the preparation of this research, large language models (LLMs) were employed strictly as a limited-purpose auxiliary tool. The models were used exclusively for language polishing tasks, including grammar checking, sentence structure optimization, and wording refinement to improve the readability and linguistic fluency of portions of the text. The LLMs played no role in any core research activities, including but not limited to: research ideation, theoretical development, experimental design, data analysis, result interpretation, or scientific decision-making. All intellectual contributions to this work originate solely from the human authors. The authors take full responsibility for the entire content of this paper, including text polished by LLMs, and affirm its originality, accuracy, and academic integrity.