Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Highlights

Harnessing noisy LLM annotations: Confidence-calibrated node selection on text-attributed graphs

Pattern Recognition xxx (xxxx) xxx

Zihan Fang, Shide Du, Zhihao Wu, Zhiling Cai, Yanchao Tan, Shiping Wang*, Zhouchen Lin

- Proposes confidence-calibrated node selection of text attribute graphs.
- Propagates reliability for noisy LLM annotations using prototypes and structure.
- Shows consistent gains on real-world datasets under diverse LLM noise.

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Harnessing noisy LLM annotations: Confidence-calibrated node selection on text-attributed graphs

Zihan Fang [a,1], Shide Du [a,1], Zhihao Wu [b], Zhiling Cai [c], Yanchao Tan [a], Shiping Wang [a,*], Zhouchen Lin [d]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou, 350108, China
[b] College of Computer Science and Technology, Zhejiang University, Hangzhou, 310058, China
[c] College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou, 350002, China
[d] State Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University, Beijing, 100871, China

ARTICLE INFO

Keywords:
LLM-generated annotations
Node classification
Graph active learning
Learning from noisy label
Text-attributed graphs

ABSTRACT

Label scarcity remains a core challenge in graph-based learning, especially in text-attributed graphs (TAGs), where entangled node semantics further increase annotation difficulty. Recently, Large Language Models (LLMs) have emerged as a promising alternative oracle, but their predictions are inherently noisy and tend to be overconfident. In this paper, we propose a confidence-calibrated node selection framework that explicitly models and harnesses the inherent characteristics of noisy LLM annotations. Specifically, we estimate annotation difficulty via soft assignment to construct a diverse candidate pool while controlling annotation cost. Through empirical analysis of noisy LLM annotations, we approximate the underlying confusion pattern using a prototype-based confusion matrix, which captures inter-class ambiguity and provides an interpretable global view of confusion patterns. Building on this insight, we introduce a graph-LLM confidence calibration module, which jointly models global label reliability and performs graph-aware propagation to adjust confidence scores based on neighborhood label distributions. Extensive experiments on real-world datasets demonstrate that the proposed method significantly outperforms baselines, providing a principled approach to harnessing noisy LLM annotations.

1. Introduction

With the rapid development of online platforms, modern information systems continuously generate massive volumes of user-generated content in textual form, tightly coupled with rich interaction structures such as academic graphs and social networks. Text-attributed graphs (TAGs) provide a principled way to model such data by extending conventional graph structures with rich textual information attached to each node, enabling joint exploitation of both network topology and semantic content [1,2]. Although existing methods have made significant progress on TAG learning, they typically assume easy access to ground-truth labels, overlooking the practical difficulty of annotating massive unlabeled data due to the high cost of expert involvement [3]. To reduce annotation cost, active learning has been widely adopted to iteratively select the most informative nodes for annotation [4,5].

Recent advances in large language models (LLMs), including GPT and Qwen, have enabled a new paradigm that leverages LLMs as zero-shot annotators to generate pseudo-labels [6,7]. In contrast to traditional Oracle-based annotation, which assumes access to accurate labels from a reliable Oracle and often relies on topology-sensitive heuristics [8], this strategy greatly reduces the reliance on manually annotated ground truth. However, despite this advantage, LLM-as-Oracle frameworks remain constrained by two key limitations. On the one hand, although LLMs have shown strong generalization capabilities in text-based tasks, their predictions often suffer from semantic confusion, overconfidence, and task bias, making the reliability of these automatically generated labels questionable [9–11]. Many existing methods assume that pseudo-label noise is uniform and independent [12]. Under this assumption, it is difficult to truly align with the real behavior of LLMs when selecting nodes or filtering samples. On the other hand, the pseudo-labels and confidence estimates provided by LLMs are usually generated based on individual node text, ignoring the semantic consistency and propagation patterns between neighboring nodes in the graph structure. Intuitively, pseudo-labels should be informative, but

* Corresponding author.
E-mail addresses: fzihan11@163.com (Z. Fang), dushidems@gmail.com (S. Du), zhihaowu1999@gmail.com (Z. Wu), zhilingcai@126.com (Z. Cai), yctan@fzu.edu.cn (Y. Tan), shipingwangphd@163.com (S. Wang), zlin@pku.edu.cn (Z. Lin).
[1] Zihan Fang and Shide Du contributed equally to this work.

https://doi.org/10.1016/j.patcog.2026.114379
Received 15 February 2026; Received in revised form 28 June 2026; Accepted 29 June 2026

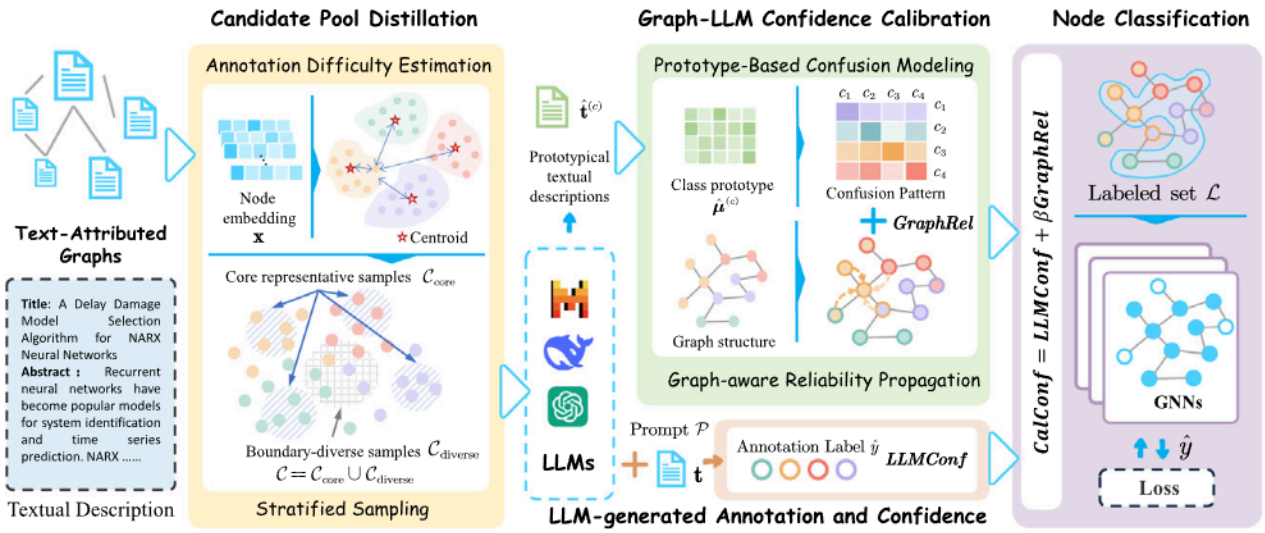


Fig. 1. Overview of the proposed framework: (1) Candidate Pool Distillation selects core and diverse samples; (2) LLM-generated Annotation and Confidence obtains label predictions and ***LLMConf***; (3) Graph-LLM Confidence Calibration derives ***GraphRel*** via prototype-based confusion modeling; and (4) Node Classification trains a GNN using the calibrated confidence ***CalConf***.

they must also be sufficiently reliable to avoid injecting excessive noise into downstream models [13]. These observations naturally lead to the following question: ***How can we effectively harness the semantic prior of LLMs together with structural cues in TAGs to guide reliable sample selection under noisy annotation conditions?***

To address these challenges, we present a two-stage selection framework for learning on TAGs with noisy LLM-generated annotations that achieves label-efficient node classification through calibrated confidence and graph-aware reliability. To ensure the selected annotations is both informative and diverse, we first construct a candidate pool via stratified sampling guided by annotation difficulty. Since LLM-generated annotations often exhibit class-dependent noise and semantic ambiguity, we then propose a confidence-calibrated learning method that enables reliable and efficient node selection under such noisy annotations. Central to the proposed method is the construction of a prototype-based confusion matrix derived from LLM semantic behavior, which captures systematic misclassification patterns. Building upon this, we design a graph-aware reliability propagation mechanism that adjusts pseudo-label reliability based on both global confusion trends and structural consistency within the graph. Accordingly, nodes are selected by jointly considering LLM-generated confidence and calibrated Graph-LLM confidence, resulting in a labeled set that is both accurate and representative. Finally, we perform subsequent classification tasks on the selected label set, enabling the training of a classification model under noisy labels within a limited budget. The overall framework is shown in Fig. 1. The key contributions of our method are summarized as follows:

- We propose an active node selection framework under noisy LLM annotations that balances informativeness and diversity. It combines difficulty-guided candidate construction with calibrated selection based on semantic confidence and graph-aware reliability.
- Motivated by the observed noise patterns in LLM predictions, we develop a graph-aware reliability propagation mechanism. This enables pseudo-label reliability calibration by jointly exploiting the LLM's global semantic prior and the graph's local structural evidence.
- We conduct extensive experiments on real-world datasets, demonstrating that our method consistently outperforms baselines under various LLM-generated noise settings, validating its robustness and generalizability.

2. Related work

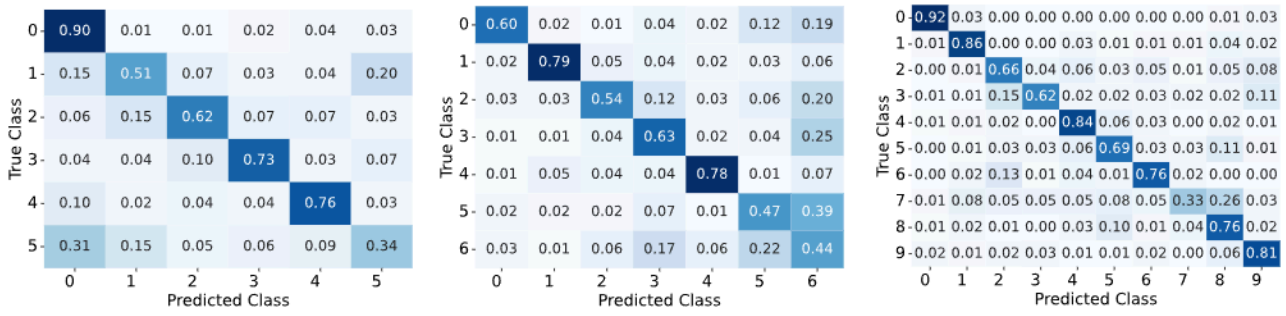
In this paper, we focus on node selection in text-attributed graphs (TAGs) under noisy LLM-generated annotations. Accordingly, we review the related work from three perspectives: Graph active learning, LLM-as-Oracle, and learning from noisy labels.

2.1. Graph active learning

Active learning aims to reduce labeling cost by selecting the most informative samples for annotation. In graph learning, selected nodes can affect their neighborhoods through message passing, making graph-aware selection especially important. Existing graph active learning methods commonly select nodes according to uncertainty, representativeness, diversity, density, centrality, or graph partition structures. For example, GraphPart [14] divides the graph into structurally distinct subgraphs and selects representative nodes from each partition, ensuring diversity while preserving local structural coherence. AGE [15] actively selects labeled nodes for graph embedding by combining uncertainty, information density, and graph centrality with time-sensitive weighting. RIM [16] connects graph active learning with influence maximization and introduces an influence quality factor to estimate the reliability of selected labels. KyN [17] provides an accurate estimate of the homogeneity distribution for GNNs through a subgraph importance sampling framework, which is applicable to heterogeneous graphs. However, most graph active learning methods still rely on a human oracle to provide queried labels, implicitly assuming that the returned annotations are reliable. Although such a setting is reasonable in conventional active learning, it becomes less practical when large-scale annotation is required, since human labeling remains costly and time-consuming. Different from existing graph active learning methods, the proposed method focuses on node selection under LLM annotations.

2.2. LLM-as-Oracle

Recent advances in LLMs have introduced a new paradigm of annotation, where LLMs provide pseudo-labels with minimal human intervention [18,19]. By leveraging task instructions, node texts, and category definitions, LLMs exhibit strong zero-shot and few-shot generalization capabilities, offering a cost-effective alternative to large-scale manual labeling. However, classical Oracle-based annotation usually assumes that queried labels are obtained from an infallible Oracle, an



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

Fig. 2. Empirical noise transition matrices $\mathbf{T}$ on three TAG datasets. Each entry $\mathbf{T}[i][j]$ denotes the probability that an instance with ground-truth class i is annotated by the LLM as class j.

assumption that no longer holds when the annotator itself is a noisy LLM [20]. In this setting, label noise becomes inevitable, and directly using LLM-generated pseudo-labels may propagate unreliable supervision into downstream learning. Several recent studies have explored LLM-based node classification on TAGs [7,10]. LLM-GNN [6] combines an LLM-based labeling difficulty heuristic with confidence-score ranking, and further measures the entropy change of labels after removing nodes from the selected set. LOCLE [21] iteratively selects critical nodes using label disharmonicity and entropy, refines pseudo-labels by combining LLM and GNN predictions, and improves performance under a limited LLM query budget. DMA [10] explicitly models LLM annotation noise distributions and selects nodes by maximizing reliable influence based on RIM. GNN-as-Judge [22] utilizes the GNN as a pseudo-label judge to exploit both the agreement and disagreement patterns between LLMs and GNNs, achieving reliable label selection. These methods demonstrate the potential of combining LLMs and graph learning for label-free node classification. However, most existing pipelines mainly use LLMs as pseudo-label generators or confidence providers, without explicitly converting the LLM's semantic knowledge into a structured prior for modeling class-level confusion.

2.3. Learning from noisy labels

Learning from noisy labels aims to train reliable models when the observed labels are corrupted or uncertain. This problem is particularly relevant to LLM-as-Oracle annotation, where LLM-generated pseudo-labels may contain errors and their reported confidence may not faithfully reflect label correctness [9]. Recent studies have shown that LLM confidence can be poorly calibrated. For example, Xiong et al. [23] systematically evaluate black-box confidence elicitation in LLMs and show that LLMs often exhibit overconfidence, assigning high confidence even to incorrect answers. These findings indicate that directly trusting raw LLM confidence can be unreliable. Some recent methods address noisy LLM-generated labels through correction or refinement. NoiseAL [24] improves robustness to noisy LLM annotations by dynamically partitioning noisy data and correcting samples with an actively prompted LLM. Ye et al. [11] iteratively refines noisy candidates using a simplex diffusion model to calibrate classifier predictions and mitigate the impact of LLM-generated noisy labels. Another line calibrates noisy labels using noisy-label transition matrix estimation, which characterizes the probability that a clean label is corrupted into a noisy label. For example, Patrini et al. [25] estimate a class-dependent noise transition matrix and use it for forward/backward loss correction to train deep networks robustly under noisy labels. Xia et al. [26] first initialize the transition matrix using high-confidence examples and then refine it with a learnable slack variable jointly optimized with the classifier. Yao et al. [27] introduce an intermediate class and factorize the original transition matrix into the product of two easier-to-estimate matrices, thereby reducing transition-matrix estimation error. Despite their effectiveness, these methods usually require clean validation samples, anchor-point assumptions, or accurate posterior estimation, which are difficult to satisfy in LLM-as-Oracle TAG annotation. Different from existing methods, the proposed method jointly exploits LLM confidence, prototype-based semantic confusion, and graph-aware reliability propagation for robust node selection under noisy LLM annotations.

In this work, we observe that LLMs are not merely annotators but also powerful representation learners whose internal semantics implicitly encode class-dependent biases and confusion patterns. This motivates our approach: instead of treating LLM annotations as structureless supervision, we aim to leverage the LLM's own semantic understanding to model its induced confusion patterns and construct a model-aware prior for the transition process.

3. Background and motivation

In this section, we begin by analyzing LLM annotation patterns on TAGs.

3.1. LLM annotation behavior

To analyze the noise distribution in LLM annotations, we construct a noise transition matrix $\mathbf{T} \in \mathbb{R}^{C\times C}$ based on LLM predictions $\hat{y}$ and ground-truth labels y:

$$\mathbf{T}[i][j] = \frac{\sum_{n=1}^{|\mathcal{V}|} \mathbb{I}(y_n = i \wedge \hat{y}_n = j)}{\sum_{n=1}^{|\mathcal{V}|} \mathbb{I}(y_n = i)}, \tag{1}$$

where $\hat{y}_n$ denotes the label assigned by the LLM to node n, and $\mathbf{T}[i][j]$ captures the empirical probability that a node with true class i is predicted as class j.

3.2. Characteristic of LLM annotations

From the class-wise confusion matrices in Fig. 2, we derive several key observations:

Observation 1: LLMs are noisy annotators. Although LLMs capture rich semantic knowledge from large-scale corpora, they may still produce noisy predictions in domain-specific classification tasks due to limited task-specific discriminative ability.

Observation 2: LLMs exhibit class-dependent prediction reliability. LLM accuracy is not uniform across classes. For instance, the 0-th category ("*Agents*") in Citeseer is predicted with high accuracy, likely because it corresponds to a more concrete and distinctive concept. Categories with clearer semantics or stronger representation in pretraining corpora are annotated more reliably than abstract or underrepresented ones.

Observation 3: LLMs reveal distinct label correlation patterns. LLM confusions are not random but often follow semantic associations

Fig. 3. t-SNE visualization on the **WikiCS** dataset. "X" denotes an incorrectly activated node, and "O" denotes a correctly activated one. Annotation ACC reflects the reliability of LLM labels on selected nodes, while predict ACC indicates the performance of the model trained on them.

learned during pretraining. For instance, in the Citeseer dataset, the 2-nd category ("*Information Retrieval*") is often misclassified as the 1-st category ("*Machine Learning*"). Such structured confusion patterns can be exploited as useful prior knowledge.

Inspired by these observations, we leverage the semantic modeling capability of LLMs to construct a class-level confusion matrix that simulates the characteristics of LLM annotations. It is not intended to exactly recover the true noise transition matrix; instead, it serves as a semantic prior that approximates class-level confusion in the LLM embedding space and guides reliability-aware node selection in a label-free setting.

3.3. LLM-annotated samples in active learning

This intuitive result in Fig. 3 highlights the following key insights:

Observation 4: LLMs achieve higher annotation quality in semantically dense regions. Correctly annotated samples are mostly located in cluster cores, whereas incorrectly annotated ones are more likely to appear near class boundaries. Empirically, degree-based selection achieves an annotation accuracy of 78.00%, substantially higher than random selection with 66.00%. This reveals that LLMs are better at making predictions in dense regions, but tend to struggle in ambiguous areas or when semantic overlap occurs.

Observation 5: Annotation correctness alone does not guarantee improved downstream model performance. Even simple random sampling can yield competitive downstream prediction accuracy when it covers more diverse decision boundaries. This suggests that the informativeness of labeled samples is more important than mere correctness in node selection.

Therefore, supervision that is diverse and representative is more valuable than labels that are correct yet homogeneous.Accordingly, we pre-select a candidate set of informative nodes from the original unlabeled node pool.

4. Method

We propose an active node selection framework that identifies reliable LLM-generated annotations for graph-based learning. Our method is composed of the following sequential stages: (1) Candidate pool distillation, (2) LLM-generated annotation and confidence, and (3) Graph-LLM confidence calibration.

4.1. Problem statement

Formally, a TAG can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^{|\mathcal{V}|}$ denotes the set of text attributes associated with the nodes. Each node $v_i \in \mathcal{V}$ is associated with an unknown class label $y_i \in \{1, \ldots, C\}$ and paired with a textual description $\mathbf{t}_i \in \mathcal{T}$, such as a title, abstract, or metadata. The edge set $\mathcal{E}$ captures structural relationships among nodes, including citation links, co-authorships, or social connections. The node embeddings, denoted as $\mathbf{x}_i$, are obtained by encoding $\mathbf{t}_i$ with a shallow text encoder.

We first distill a candidate pool $\mathcal{C} \subset \mathcal{V}$ based on annotation difficulty and query a noisy Oracle for labels. We then analyze the Oracle's annotation behavior to inform selection. Given a budget b (with $b \ll |\mathcal{V}|$), the goal is to select a labeled set $\mathcal{L} = \{(v_i, \hat{y}_i)\}_{i=1}^{b}$ from $\mathcal{C}$ such that a model trained on $\mathcal{L}$ achieves strong downstream performance.

4.2. Candidate pool distillation

To balance informativeness and diversity, we first construct a candidate set $\mathcal{C}$ such that $|\mathcal{C}| > |\mathcal{L}|$. The final labeled set $\mathcal{L}$ is then selected from this candidate pool using the proposed scoring mechanism.

4.2.1. Soft assignment-based annotation difficulty estimation

We first define a soft assignment function $SoftAssn(\mathbf{x}, \boldsymbol{\mu}^{(c)})$ that measures the semantic affinity between a sample $\mathbf{x}$ and a class prototype $\boldsymbol{\mu}^{(c)}$. In this work, we adopt Student's t-distribution kernel [28] to transform distance-based similarities into probabilistic soft assignments:

Definition 1 (*Student's t-distribution-based Soft Assignment*)**.** Given a sample embedding $\mathbf{x}$ and a class prototype $\boldsymbol{\mu}^{(c)}$, the assignment probability is

$$SoftAssn(\mathbf{x}, \boldsymbol{\mu}^{(c)}) = \frac{(1 + \|\mathbf{x} - \boldsymbol{\mu}^{(c)}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{i=1}^{C}(1 + \|\mathbf{x} - \boldsymbol{\mu}^{(i)}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{2}$$

where $\alpha > 0$ controls the heaviness of the distribution tail.

This t-distribution kernel probabilistically models class affinity. Thus, the annotation difficulty metric can be defined as:

$$\boldsymbol{AnoConf}(i) = \max_{c} SoftAssn(\mathbf{x}_i, \boldsymbol{\mu}^{(c)}). \tag{3}$$

Here, higher values indicate clearer class assignments, and samples are then sorted by ***AnoConf*** and divided into percentile bands. Then, candidate set $\mathcal{C}$ is constructed via the following stratified sampling:

$$\mathcal{C} \sim Sample(\mathcal{C}_{\text{core}} \cup \mathcal{C}_{\text{diverse}}) \tag{4}$$

- *Representative Samples*: Core region samples $\mathcal{C}_{\text{core}}$ prioritize samples in dense regions of the feature space, which are drawn from the top $q\%$ of the ranking.
- *Boundary-Diverse Samples*: Boundary region samples $\mathcal{C}_{\text{diverse}}$ ensure decision-boundary exploration and semantic coverage, which are drawn from the $q\%$ of the ranked samples after $\mathcal{C}_{\text{core}}$.

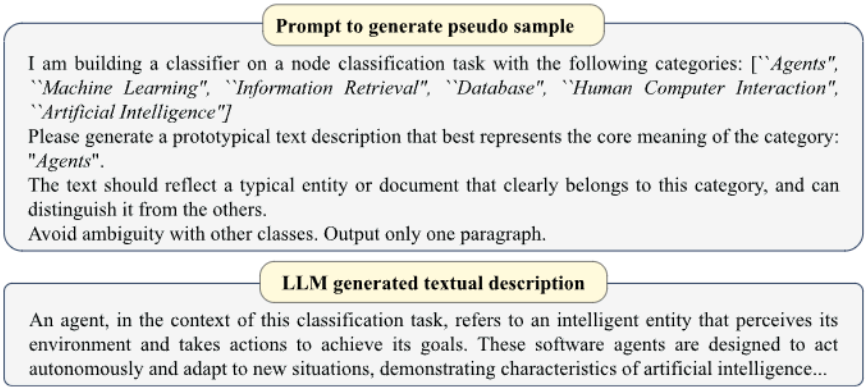


Fig. 4. An example prompt and an LLM-generated prototypical textual description for the "Agents" category in the **CiteSeer** dataset.

Specifically, given a target candidate size $|C|$, we sample $\lfloor \rho|C| \rfloor$ points from C_{core} and the remaining $|C| - \lfloor \rho|C| \rfloor$ from C_{diverse}. Here, $\rho \in [0, 1]$ is the core-diversity mixing coefficient that controls how many candidates are drawn from dense core regions versus diverse boundary regions. Intuitively, larger ρ emphasizes representativeness and stability, while smaller ρ increases exploration and coverage near the decision boundary. This step filters out redundant nodes while still retaining sufficient diversity for downstream selection.

4.3. LLM-generated annotation and confidence

For each node $v_i \in C$, we query an LLM with a domain-specific prompt $\mathcal{P}$ to obtain its pseudo-label $\hat{y}_i$ and corresponding confidence:

$$\hat{y}_i, \boldsymbol{LLMConf}(v_i) := \text{LLM}(\mathcal{P}(\mathbf{t}_i)), \tag{5}$$

where $\mathbf{t}_i$ denotes the textual description of node v_i, and $\boldsymbol{LLMConf}(v_i) \in [0, 1]$ is the confidence score associated with the LLM annotation $\hat{y}_i$, reflecting the model's estimated confidence in its prediction. A typical prompt $\mathcal{P}(\mathbf{t}_i)$ structure can be written as:

$$\langle \text{Instruct} \rangle \ [\text{Class Info}][\text{Task Description}] \ \mathbf{t}_i. \tag{6}$$

Here, "Class Info" provides textual descriptions of all candidate classes, while "Task Description" outlines the classification objective.

While ***LLMConf*** reflects the LLM's confidence in its predictions, previous studies have shown that such confidence estimates are often overconfident and poorly calibrated [29,30]. Moreover, such confidence scores capture only local textual semantics and ignore the structural reliability of a node within the graph topology. To address these limitations, we refine pseudo-label reliability by incorporating graph-aware reliability correction.

4.4. Graph-LLM confidence calibration

To approximate the inherent semantic ambiguity in LLM-generated predictions, we construct a confusion matrix using LLM-generated pseudo-samples. We then introduce a graph-aware reliability propagation mechanism that leverages both semantic confusion patterns and topological structures to refine confidence estimates.

4.4.1. Prototype-based confusion modeling via LLM

Empirically, the noise transition matrix $\mathbf{T}$ can be class-dependent (some classes are systematically "easier" to annotate) and asymmetric (confusions from class i to class j occur more frequently than those from class j to class i). Moreover, annotation results may vary with the input context, such as prompt templates. In addition, in real-world training scenarios, it is difficult to obtain enough labeled samples that reflect the true label distribution to compute the empirical noise transition matrix. Therefore, we approximate $\mathbf{T}$ by modeling class-wise semantic relationships $\mathbf{P}$ using LLM-generated prototypes.

Formally, given a set of C class labels $\mathcal{Y} = \{y^{(c)}\}_{c=1}^{C}$, we employ LLMs to generate prototypical textual descriptions $\hat{\mathbf{t}}^{(c)}$ for class $y^{(c)}$ using designed prompts $\mathcal{P}(y^{(c)}, \mathcal{Y})$:

$$\hat{\mathbf{t}}^{(c)} := \text{LLM}(\mathcal{P}(y^{(c)}, \mathcal{Y})). \tag{7}$$

Specifically, we query the LLM to generate pseudo-samples for each class using the prompt shown in Fig. 4. In this way, for each class, we generate K prototypical textual descriptions $\mathcal{T}^{(c)} = \{\hat{\mathbf{t}}_k^{(c)}\}_{k=1}^{K}$ with the LLM for each class.

Subsequently, the LLM encoder maps these textual prototypes into K pseudo-sample embeddings $\{\mathbf{h}_k^{(c)}\}_{k=1}^{K}$ for each class c:

$$\mathbf{h}_k^{(c)} = \text{Encoder}_{\text{LLM}}(\hat{\mathbf{t}}_k^{(c)}), \tag{8}$$

where $\text{Encoder}_{\text{LLM}}$ denotes the embedding function of the LLM encoder. These descriptions capture representative class semantics from the LLM's knowledge base. The class prototype $\hat{\boldsymbol{\mu}}^{(c)} \in \mathbb{R}^d$ is then computed as the centroid of K pseudo samples in the embedding space:

$$\hat{\boldsymbol{\mu}}^{(c)} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{h}_k^{(c)}, \forall c \in \{1, \ldots, C\}. \tag{9}$$

To quantify inter-class semantic relationships, we employ a Student's t-distribution kernel to compute the soft assignment probability between each pseudo sample and all class prototypes. The class-level confusion matrix $\mathbf{P} \in [0, 1]^{C \times C}$ is derived by averaging these soft assignments:

$$\mathbf{P}[i][j] = \frac{1}{K} \sum_{k=1}^{K} SoftAssn(\mathbf{h}_k^{(i)}, \hat{\boldsymbol{\mu}}^{(j)}). \tag{10}$$

Each entry $\mathbf{P}[i][j]$ represents the expected probability that a prototype sample from class i would be semantically associated with class j, thereby capturing confusion patterns in the LLM's semantic space. This prototype-based confusion matrix offers a principled representation of inter-class ambiguity and forms a key component for downstream confidence calibration.

4.4.2. Graph-aware reliability propagation

The proposed reliability assessment incorporates two critical factors derived from the graph learning paradigm:

- *Semantic Relationship Modeling*: For confusable classes, we modulate reliability scores using learned semantic relationships encoded in the confusion matrix.
- *Topological Consistency Enforcement*: Similar connected nodes should exhibit label consistency, with local conflicts indicating annotation unreliability.

We formalize reliability propagation as follows:

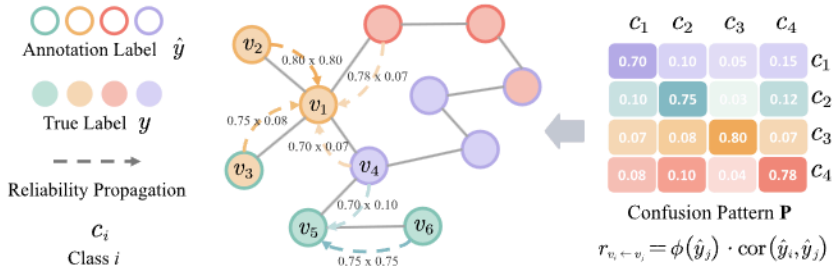


Fig. 5. Illustration of reliability propagation based on class-wise confusion pattern **P**. Given predicted labels $\hat{y}_i$ and $\hat{y}_j$, the reliability propagated from node v_j to v_i is defined as $r_{v_i \leftarrow v_j} = \phi(\hat{y}_j) \cdot \mathrm{cor}(\hat{y}_i, \hat{y}_j)$, where $\phi(\hat{y}_j) = \mathbf{P}[\hat{y}_j][\hat{y}_j]$ is the global confidence, and $\mathrm{cor}(\hat{y}_i, \hat{y}_j) = \mathbf{P}[\hat{y}_i][\hat{y}_j]$ quantifies their semantic correlation.

Definition 2 (*Label Semantic Reliability Propagation*). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph in which each node $v_i \in \mathcal{V}$ corresponds to an instance with LLM-generated annotation $\hat{y}_i$. For adjacent nodes $(v_i, v_j) \in \mathcal{E}$, the semantic reliability propagated from v_j to v_i is defined as:

$$r_{v_i \leftarrow v_j} = \phi(\hat{y}_j) \cdot \mathrm{cor}(\hat{y}_i, \hat{y}_j), \tag{11}$$

where $\phi(\hat{y}_j) \in [0, 1]$ denotes the global reliability of the predicted class $\hat{y}_j$, and $\mathrm{cor}(\hat{y}_i, \hat{y}_j)$ measures the semantic correlation between the predicted labels of v_i and v_j.

Since **P** models global semantic confusion, we define $\phi(\hat{y}_j) = \mathbf{P}[\hat{y}_j][\hat{y}_j]$ as the global reliability of the predicted label, and define $\mathrm{cor}(\hat{y}_i, \hat{y}_j) = \mathbf{P}[\hat{y}_i][\hat{y}_j]$ as the semantic correlation between node v_i and its neighbor v_j. An illustrative example of this process is shown in Fig. 5. The multiplicative form acts as a conservative reliability gate. A neighbor v_j provides strong reliability support to v_i only when its predicted label is globally reliable and semantically relevant to the predicted label of v_i. If either condition is weak, the propagated reliability is suppressed, which helps reduce the influence of unreliable or semantically inconsistent neighbors.

To quantify the graph-aware reliability of a node's pseudo-label, we define a graph-aware reliability score for each node v_i with predicted label $\hat{y}_i$ as:

$$\begin{aligned} \textbf{\textit{GraphRel}}(v_i) &= \phi(\hat{y}_i) \cdot \sum_{j \in \mathcal{N}_i} w_{ij} \cdot r_{v_i \leftarrow v_j} \\ &= \mathbf{P}[\hat{y}_i][\hat{y}_i] \cdot \sum_{j \in \mathcal{N}_i} w_{ij} \cdot \mathbf{P}[\hat{y}_j][\hat{y}_j] \mathbf{P}[\hat{y}_i][\hat{y}_j], \end{aligned} \tag{12}$$

where $\mathcal{N}_i$ denotes the neighbors of node v_i, $\phi(\hat{y}_i) = \mathbf{P}[\hat{y}_i][\hat{y}_i]$ measures the global reliability of the predicted class of v_i, and $r_{v_i \leftarrow v_j}$ denotes the semantic reliability propagated from neighbor v_j to v_i. The normalized edge similarity weight w_{ij} is computed from the distance between node embeddings. Specifically, for each neighbor $v_j \in \mathcal{N}_i$, we first define $\tilde{w}_{ij} = 1/(\|\mathbf{x}_i - \mathbf{x}_j\|_2 + \epsilon)$, where ϵ is a small constant for numerical stability. The normalized weight is then computed as $w_{ij} = \tilde{w}_{ij} / \sum_{k \in \mathcal{N}_i} \tilde{w}_{ik}$, so that neighbors with more similar node representations contribute more strongly to the reliability estimation.

Eq. (12) can be interpreted as a conservative reliability gate. A neighbor v_j provides strong reliability support to v_i only when three conditions are simultaneously satisfied: the predicted class of v_i is reliable, the predicted class of v_j is reliable, and the two predicted labels are semantically compatible. If any of these factors has a low value, the propagated reliability is suppressed. Therefore, the multiplicative form prevents unreliable or semantically inconsistent neighbors from dominating the graph-aware reliability estimation.

We note that the proposed label semantic reliability propagation relies on a mild homophily assumption: connected and semantically similar nodes are expected to provide consistent label evidence. This assumption is reasonable for many citation-based TAGs, where linked documents often share related topics or belong to semantically related classes. However, this assumption may be weakened in low-homophily or heterophilous graphs, where neighboring nodes can have different labels and direct reliability propagation may introduce misleading evidence. To mitigate this issue, GraphRel does not simply aggregate neighbor confidence. Instead, each neighbor is weighted by both the embedding-based similarity weight w_{ij} and the semantic compatibility term $\mathbf{P}[\hat{y}_i][\hat{y}_j]$, so that semantically inconsistent neighbors contribute less to the reliability estimation. Nevertheless, when the graph structure strongly violates homophily, the effectiveness of graph-aware reliability propagation may decrease.

This weighting strategy encourages greater influence from structurally similar and semantically consistent neighbors. The resulting ***GraphRel*** offers a calibrated confidence measure that jointly accounts for both the LLM-induced confusion patterns and the structural constraints of the graph.

Algorithm 1 Proposed Node Selection Algorithm

Require: A TAG $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, budget b, LLM with prompt template $\mathcal{P}$, mixing coefficient ρ, candidate size $|C|$, balance factor β, number of generated prototypical samples K
Ensure: Labeled node set $\mathcal{L}$
Distill a candidate pool $C \subset \mathcal{V}$ via Eq. (4)
for each node $v_i \in C$ **do**
  Query the LLM with textual input $\mathbf{t}_i$ using prompt $\mathcal{P}$
  Obtain pseudo-label $\hat{y}_i$ and confidence score ***LLMConf***(v_i) via Eq. (5)
end for
for each class $c \in \{1, \ldots, C\}$ **do**
  Generate prototypical texts and encode them into embeddings $\{\mathbf{h}_k^{(c)}\}_{k=1}^{K}$ via Eq. (8)
  Compute class prototype $\hat{\boldsymbol{\mu}}^{(c)}$ via Eq. (9)
end for
Estimate the confusion matrix **P** via Eq. (10)
for each node $v_i \in C$ **do**
  Compute graph-aware reliability score ***GraphRel***(v_i) via Eq. (12)
  Compute final calibrated confidence ***CalConf***(v_i) via Eq. (13)
end for
Rank all nodes in C by ***CalConf***(v_i) in descending order
Select top b nodes per class to construct labeled set $\mathcal{L}$
Return the final labeled node set $\mathcal{L}$ with LLM-generated annotations

4.5. Node classification

We define the calibrated scoring function for pseudo-labeled node selection as:

$$\textbf{\textit{CalConf}} = \textbf{\textit{LLMConf}} + \beta \cdot \textbf{\textit{GraphRel}}, \tag{13}$$

where the coefficient β balances the contribution between global structural reliability and local annotation certainty. Intuitively, nodes with higher ***CalConf*** scores are considered more trustworthy, as they are

Table 1
Statistics of the TAG datasets.

Dataset	#Nodes	#Edges	#Classes	Domain
CiteSeer	3186	4277	6	Academic
Cora	2708	5429	7	Academic
WikiCS	11,701	216,123	10	Wikipedia
PubMed	19,717	44,338	3	Academic
Arxiv	169,343	1,166,243	40	Academic

supported by both confident predictions from the LLM and agreement with reliable neighboring nodes. The final labeled set $\mathcal{L}$ is constructed by selecting nodes with high ***CalConf*** scores, thereby balancing LLM semantic confidence with graph-aware reliability. This selection strategy provides high-quality pseudo-supervision for subsequent model training. The overall procedure of our node selection strategy is summarized in Algorithm 1.

Given the selected labeled node set $\mathcal{L} = \{\mathbf{x}, \hat{y}\}$, we train a lightweight classifier using the standard cross-entropy loss. This training process effectively exploits pseudo-supervision that is both semantically reliable and structurally representative, thereby enabling robust and generalizable graph learning under noisy LLM annotations.

5. Experiments

We conduct comprehensive experiments to validate the effectiveness of our framework on the benchmark datasets.

5.1. Experimental settings

We adopt Qwen-2.5-14B-Instruct [31] as the primary LLM Oracle for our main experiments. All experiments are conducted with NVIDIA GeForce RTX 3090 GPUs (24 GB each).

5.1.1. Datasets

To comprehensively evaluate the effectiveness of our proposed framework, we conduct experiments on five widely adopted benchmark datasets for text-attributed graphs: **CiteSeer** [32], **Cora**, **WikiCS** [33], **PubMed**, and **Arxiv** (ogbn-arxiv) [34]. Most of these graphs are formed by citation links among documents, while WikiCS is constructed from Wikipedia article links. Detailed dataset statistics are provided in Table 1.

- **CiteSeer** contains 3186 scientific publications classified into one of 6 computer science categories: ["Agents", "Machine Learning", "Information Retrieval", "Database", "Human Computer Interaction", "Artificial Intelligence"].
- **Cora** comprises 2708 scientific publications categorized into 7 machine learning-related topics: ["Case Based", "Genetic Algorithms", "Neural Networks", "Probabilistic Methods", "Reinforcement Learning", "Rule Learning", "Theory"].
- **WikiCS** is constructed from Wikipedia, where each article belongs to one of 10 categories: ["Computational Linguistics", "Databases", "Operating Systems", "Computer Architecture", "Computer Security", "Internet Protocols", "Computer File Systems", "Distributed Computing Architecture", "Web Technology", "Programming Language Topics"].
- **PubMed** contains 19,717 biomedical papers from the PubMed database, classified into one of 3 diabetes-related categories: ["Diabetes Mellitus Experimental", "Diabetes Mellitus Type 1", "Diabetes Mellitus Type 2"].
- **Arxiv** is a large-scale network of arXiv computer science papers. Each node is assigned to one of 40 subject areas (*e.g.*, cs.AI, cs.LG, cs.CV), which can be accessed via the arXiv category taxonomy: https://arxiv.org/archive/cs.

Table 2
Prompt template for LLM-based single-label classification with structured JSON output.

Prompt Template:

Suppose you are an expert in computer science. Below are ten computer science subcategories: `[categories]`.
Your task is to read a paper abstract and classify it into one of the above categories.
Please return your answer strictly in the following JSON format:
{ "prediction": "`[predicted label]`", "confidence": `[confidence score]` }
What is the category of this paper: `[Paper Information]`

Example Output:

{ "prediction": "machine learning", "confidence": 85 }

5.1.2. Baselines

For comparison, we consider representative node selection baselines based on graph structure, node features, and hybrid heuristics. GraphPart [14], AGE [15], and RIM [16] are introduced in Section 2.1, while LLM-GNN [6] and LOCLE [21] are described in Section 2.2. We implement two variants of LLM-GNN, using AGE and GraphPart as the underlying selectors, respectively. For fairness, we use shallow embeddings as input features for Locle instead of pretrained language model representations. Other baselines are described below:

- **Random**: Select nodes uniformly at random without leveraging any structural or semantic information.
- **Density**: Perform clustering on the node embeddings and prioritize nodes located in high-density regions. Density is measured as the inverse of the average ℓ_2-distance to the cluster centroid.
- **Degree**: A simple topology-driven heuristic that selects high-degree nodes, selects high-degree nodes under the assumption that such nodes have stronger connectivity and thus greater influence in the label propagation process.
- **PageRank**: Computes node importance via the PageRank algorithm and selects the top-ranked nodes. This method captures both local and global structural prominence, often favoring hub-like nodes for annotation.

5.1.3. Implementation details

For active selection, we follow common semi-supervised settings with a budget of b labels per class and use a two-layer GCN as the projection head. We use the LLM with default settings to generate $K = 10$ prototypical samples per class for computing the confusion matrix, and q and ρ are selected from $\{20\%, 40\%\}$ and $\{30\%, 40\%\}$, respectively. For smaller datasets, we set the candidate set ratio to 10%, i.e., $|C| = 10\%|\mathcal{V}|$, while for larger datasets such as Arxiv, we use a 5% candidate set ratio. This design provides a trade-off between reliability and diversity across different data scales. After selecting $b \times C$ pseudo-labeled nodes via our strategy, the GCN is trained and evaluated on the remaining nodes. β is selected from {0.01, 0.1}. We report the average accuracy (ACC) and Macro-F1 over five runs by comparing predicted and ground-truth labels. It is worth noting that the empirical transition matrix $\mathbf{T}$ in Section 3 is computed with ground-truth labels only for diagnostic analysis of LLM annotation behavior. It is not used by the proposed node selection algorithm.

5.1.4. Prompts for annotation and prototype generation

To support LLM's annotation and prototype construction, we design two types of prompts tailored to different tasks. To generate class-representative prototypes, we design the prompt shown in Fig. 4. It guides the LLM to produce a textual description that encapsulates the core semantics of each class. The prompt template in Table 2 elicits structured predictions from the LLM where both the predicted label and its associated confidence score are extracted in JSON format.

Table 3
Performance comparison of all methods across five datasets under different budgets. The best and runner-up results are highlighted in red and blue, respectively (mean% ± standard deviation%). OOT means out of time.

	Method/Dataset	CiteSeer		Cora		WikiCS		PubMed		Arxiv	
	Budget	5	10	5	10	5	10	10	20	20	30
ACC	Random	41.1±2.6	55.5±3.6	44.0±3.6	48.3±1.6	66.2±1.0	69.6±0.9	72.4±0.9	69.1±0.7	36.9±1.8	36.7±1.6
	Random-*C*	62.2±3.0	62.7±4.2	39.3±1.2	48.2±5.7	64.9±0.8	64.5±0.6	64.2±1.4	69.3±0.5	39.0±1.5	39.0±1.3
	Density	57.8±1.8	59.4±2.1	36.9±4.3	42.7±4.4	55.0±0.8	60.0±0.8	42.7±0.5	57.1±1.2	35.4±1.0	34.1±1.2
	Density-*C*	63.1±0.6	65.0±0.6	29.8±4.2	48.8±4.9	63.9±1.1	63.3±0.6	54.8±1.2	64.3±0.9	36.9±1.3	39.2±0.9
	Degree	60.4±1.1	62.1±2.0	56.3±0.5	58.4±0.9	54.8±1.3	54.1±1.1	62.3±0.8	63.1±1.8	8.23±1.9	9.05±1.7
	Degree-*C*	60.2±1.3	66.0±0.7	61.7±1.7	67.6±1.6	58.4±1.6	61.3±1.4	49.6±1.8	60.3±0.7	18.9±1.3	21.6±0.9
	Pagerank	54.2±2.2	61.3±2.1	56.8±0.7	58.9±1.2	56.1±1.2	59.7±0.8	59.9±0.8	61.9±1.8	7.34±2.8	9.87±1.7
	Pagerank-*C*	61.2±3.3	65.2±1.1	58.0±2.3	63.2±2.0	67.1±1.1	62.9±1.3	69.5±0.8	72.3±0.4	20.9±1.2	24.0±0.7
	AGE	60.5±0.9	65.8±0.8	51.4±1.2	58.9±1.1	56.4±1.3	60.4±0.8	54.0±1.8	58.7±1.3	9.71±2.4	10.9±1.3
	AGE-*C*	61.6±1.9	63.8±0.8	58.5±1.9	64.3±1.8	57.9±0.8	60.1±1.3	61.5±1.3	68.9±0.9	21.8±1.7	24.2±1.4
	GraphPart	63.2±3.1	60.4±1.3	57.1±3.5	62.8±2.9	61.8±1.1	66.3±0.2	74.9±0.2	71.7±0.8	OOT	OOT
	GraphPart-*C*	62.8±1.9	64.1±1.7	47.7±6.3	65.1±5.8	64.6±0.6	64.1±0.4	70.7±0.7	75.6±0.6	OOT	OOT
	RIM	38.5±1.7	47.9±4.1	55.8±1.2	54.5±1.2	63.2±1.4	63.7±0.6	74.6±0.4	75.6±0.1	OOT	OOT
	RIM-*C*	56.6±3.6	57.9±4.3	52.3±0.9	60.0±1.4	62.6±0.6	64.0±0.4	74.6±1.1	77.7±0.5	35.8±0.3	37.6±1.2
	LLM-GNN (AGE)	55.8±1.4	60.7±0.9	52.1±1.1	58.8±1.1	55.6±0.6	58.7±0.5	54.3±1.7	59.0±1.3	8.3±1.8	9.1±1.1
	LLM-GNN (GraphPart)	56.9±3.5	61.6±1.4	47.0±2.9	60.0±2.5	61.1±1.0	62.1±0.2	76.4±0.1	71.4±0.7	OOT	OOT
	LOCLE	43.1±4.8	46.3±2.7	20.6±3.1	27.4±2.9	52.9±0.9	58.7±1.1	76.2±1.6	76.3±0.4	32.7±2.5	31.4±2.7
	CalConf	65.4±1.3	65.2±1.5	63.8±4.1	64.7±3.7	54.9±1.7	64.9±1.5	66.5±0.9	70.9±1.0	34.6±2.4	39.7±2.6
	CalConf-C	63.4±2.5	67.7±1.3	62.4±5.9	67.7±3.1	70.3±2.0	74.2±0.6	75.0±0.8	77.0±0.7	40.6±2.7	43.3±1.6
Macro-F1	Random	30.9±1.8	46.6±3.2	41.5±4.2	49.8±1.5	62.0±0.8	66.9±0.9	72.3±0.9	68.3±0.9	20.2±1.5	18.8±1.8
	Random-*C*	54.0±2.5	55.2±3.8	33.5±2.2	44.7±4.6	58.8±0.8	57.2±0.5	62.6±2.0	69.1±0.5	18.8±1.4	18.4±2.8
	Density	53.1±1.0	54.5±1.6	35.4±4.5	43.3±4.1	52.3±0.7	57.6±0.5	43.2±0.4	56.8±1.1	23.0±0.6	22.2±0.8
	Density-*C*	57.7±1.9	59.3±1.5	27.0±4.0	50.1±2.4	61.3±0.9	59.9±0.6	54.6±1.0	64.0±0.8	23.8±1.4	24.4±0.5
	Degree	53.6±2.6	56.2±2.0	57.0±0.6	59.2±0.9	51.1±1.0	50.0±1.0	61.8±0.7	63.0±1.6	8.4±1.8	8.8±1.8
	Degree-*C*	55.5±1.7	60.5±1.5	61.2±1.3	66.3±1.4	54.5±1.0	57.1±1.2	48.0±2.1	58.0±0.6	17.6±1.3	19.6±0.8
	Pagerank	47.3±2.2	56.9±1.7	57.6±0.7	59.2±1.0	55.6±1.5	57.3±0.8	58.3±1.6	61.2±0.9	7.3±3.2	10.3±2.1
	Pagerank-*C*	54.6±3.4	59.9±1.9	58.2±1.6	62.9±1.5	63.0±1.1	58.7±1.3	69.0±0.8	71.3±0.4	19.9±1.0	21.2±0.9
	AGE	55.8±1.4	60.7±0.9	52.1±1.1	58.8±1.1	55.6±0.6	58.7±0.5	54.3±1.7	59.0±1.3	8.3±1.8	9.1±1.1
	AGE-*C*	54.8±2.2	57.6±1.2	57.9±0.4	63.6±1.6	54.5±0.9	57.1±1.3	58.3±1.8	68.2±0.9	18.6±0.7	20.4±0.8
	GraphPart	56.9±3.5	61.6±1.4	47.0±2.9	60.0±2.5	61.1±1.0	62.1±0.2	76.4±0.1	71.4±0.7	OOT	OOT
	GraphPart-*C*	57.7±1.5	58.3±2.6	46.1±7.3	63.0±4.2	61.8±0.4	61.0±0.3	70.7±0.8	74.6±0.5	OOT	OOT
	RIM	22.6±2.2	37.3±5.3	45.5±1.5	47.4±1.4	50.1±0.8	50.5±0.6	73.9±0.5	74.8±0.3	OOT	OOT
	RIM-*C*	51.6±3.6	50.9±4.3	42.5±0.4	55.5±2.2	48.3±1.0	49.7±0.8	73.4±1.7	77.0±0.6	10.8±0.3	12.8±1.5
	LLM-GNN (AGE)	55.8±1.4	60.7±0.9	52.1±1.1	58.8±1.1	55.6±0.6	58.7±0.5	54.3±1.7	59.0±1.3	8.3±1.8	9.1±1.1
	LLM-GNN (GraphPart)	56.9±3.5	61.6±1.4	47.0±2.9	60.0±2.5	61.1±1.0	62.1±0.2	76.4±0.1	71.4±0.7	OOT	OOT
	LOCLE	29.9±5.7	30.9±1.4	16.0±1.6	20.9±5.0	31.8±2.0	42.3±1.6	75.5±1.7	77.0±0.5	7.3±1.9	6.4±2.1
	CalConf	58.7±2.3	60.8±2.1	59.3±4.3	62.4±3.5	51.8±1.7	60.6±1.5	66.4±0.9	70.0±1.2	25.7±0.8	28.0±1.6
	CalConf-C	58.6±2.2	62.0±1.5	58.9±5.8	65.2±2.7	67.4±2.0	71.6±0.8	74.8±0.9	76.9±0.7	27.7±1.5	28.3±0.9

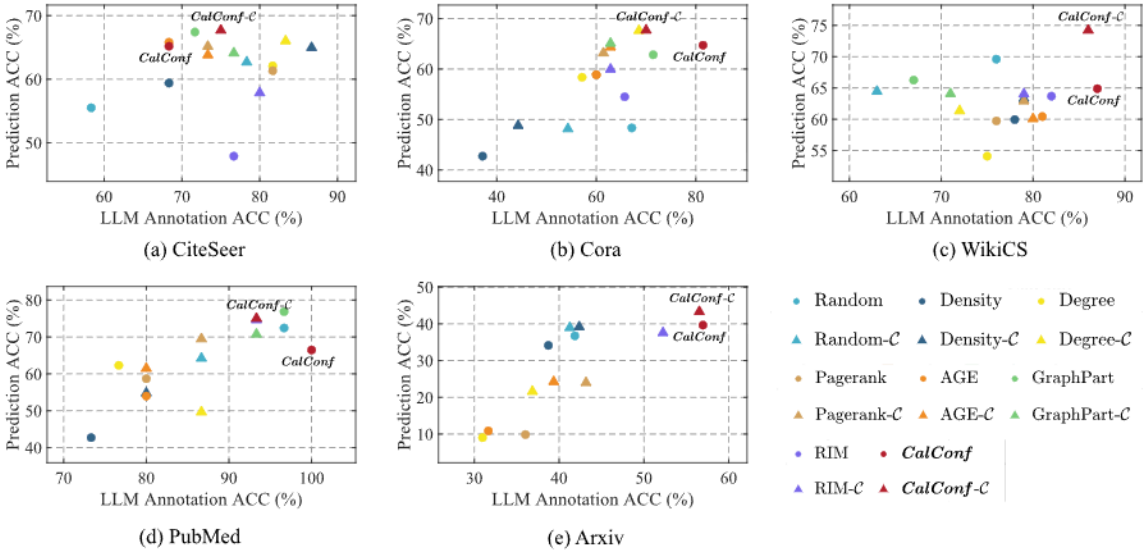


Fig. 6. Scatter plots of LLM annotation ACC for selected nodes (x-axis) *vs.* model prediction ACC (y-axis) across five datasets. Each point corresponds to a node selection strategy.

5.2. Main results

5.2.1. Performance benchmark

Table 3 presents the performance comparison across different datasets. Here, "-*C*" denotes that node selection is restricted to a pre-constructed candidate pool C. The results show that even when only 10%–20% of nodes are considered for selection, constructing a candidate pool in advance can maintain, and in some cases even improve, model performance. This suggests that the candidate pool preserves both semantic diversity and structural representativeness, effectively eliminating redundancy and enabling more efficient downstream selection. Compared with stronger baselines, our method still demonstrates strong and consistent performance across different datasets and annotation budgets. For example, LOCLE performs competitively on PubMed its performance drops significantly on Cora, WikiCS, and Arxiv, indicating that its subspace construction and local selection strategy may

Table 4
ACC (%) of ***CalConf*-*C*** with different LLMs.

Dataset	CiteSeer		Cora		WikiCS		PubMed	
Method/Budget	5	10	5	10	5	10	10	20
Qwen-14B	63.4 ± 2.5	67.7 ± 1.3	62.4 ± 5.9	67.7 ± 3.1	70.3 ± 2.0	74.2 ± 0.6	75.0 ± 0.8	77.0 ± 0.7
Qwen-30B	63.8 ± 1.9	66.7 ± 1.6	60.9 ± 6.6	67.1 ± 6.1	70.8 ± 1.0	73.2 ± 0.6	76.9 ± 1.4	72.4 ± 0.4
Mistral-7B	56.0 ± 1.9	66.6 ± 0.7	52.4 ± 4.4	59.3 ± 4.6	64.4 ± 0.9	66.2 ± 1.0	63.4 ± 1.2	67.6 ± 1.1
Mixtral-8 × 7B	64.7 ± 1.8	67.8 ± 1.2	62.1 ± 4.5	64.4 ± 4.0	63.3 ± 2.2	72.7 ± 0.5	68.9 ± 0.3	72.0 ± 0.3
Llama-8B	57.7 ± 5.9	64.8 ± 3.2	62.8 ± 4.9	65.5 ± 3.4	61.7 ± 2.1	71.6 ± 1.1	70.5 ± 0.7	79.1 ± 0.5

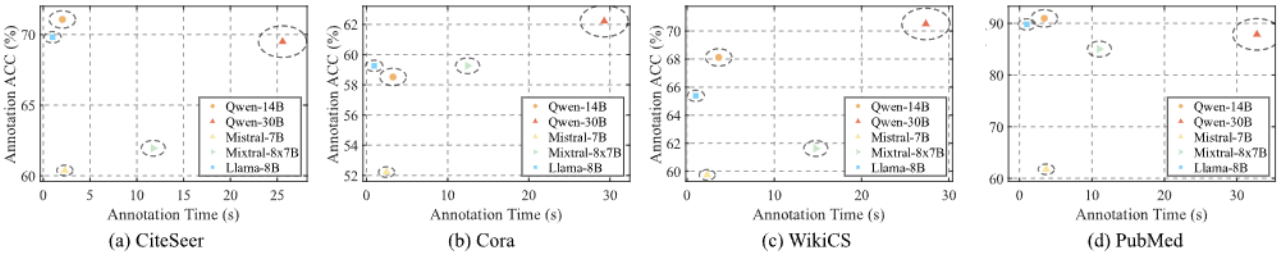


Fig. 7. LLM annotation ACC in C and spend time across four datasets, where the radius of dashed circles is proportional to the number of LLM parameters.

Table 5
ACC (%) for different ablation methods, where *w/o* denotes removal of a specific component.

Dataset	*w/o* StratSamp	*w/o* SoftAssn	*w/o GraphRel*	*w/o LLMConf*	*w/o* LLMProto	*w/o* ProtoAvg	All (ours)
CiteSeer	59.6 ± 2.4	63.5 ± 1.9	67.2 ± 1.0	65.0 ± 1.2	66.9 ± 1.8	67.5 ± 1.5	67.7 ± 1.3
Cora	45.9 ± 4.0	63.6 ± 4.8	67.0 ± 3.2	60.4 ± 6.3	63.0 ± 2.3	62.9 ± 2.2	67.7 ± 3.1
WikiCS	64.3 ± 0.7	71.1 ± 0.6	73.6 ± 0.3	49.9 ± 2.1	70.5 ± 0.9	70.2 ± 0.9	74.2 ± 0.6
PubMed	68.1 ± 1.1	70.9 ± 1.0	72.9 ± 0.7	71.0 ± 0.5	74.8 ± 0.7	74.7 ± 0.3	75.0 ± 0.8
Arxiv	42.3 ± 1.1	39.8 ± 1.5	38.4 ± 1.2	38.9 ± 1.4	40.4 ± 1.1	41.8 ± 1.2	43.3 ± 1.6

be sensitive to dataset characteristics and class imbalance. Similarly, LLM-GNN variants improve over some traditional strategies, but their gains are not consistent across all datasets, especially on Arxiv, where both LLM-GNN (AGE) and AGE show limited performance. ***CalConf*** integrates both LLM-generated confidence and graph-aware reliability, allowing it to robustly identify high-quality samples for annotation even under noisy supervision. In particular, its variant ***CalConf*-*C***, which restricts selection within the candidate pool, yields the best overall average performance, outperforming all baselines by a clear margin.

5.2.2. Decoupling LLM accuracy and model accuracy

Fig. 6 provides a comparative scatter plot analysis of LLM annotation ACC and the resulting model ACC across five benchmark datasets. Across all datasets, we observe the following trends: Variants using the candidate pool ($-C$) generally yield higher model performance, even when the LLM annotation accuracy is lower. This indicates that incorporating the candidate set effectively reduces label noise and sample redundancy, leading to improved model generalization. For example, in PubMed, although ***CalConf*** achieves high LLM accuracy, its model performance is suboptimal, likely due to overfitting noisy labels. On Arxiv, where LLM predictions are generally less accurate, only ***CalConf*-*C*** achieves usable performance, highlighting its robustness in low-quality supervision regimes. Methods such as Random and Pagerank show inconsistent improvements with candidate filtering, suggesting that their sample selection strategies may not align well with LLM guidance. ***CalConf*-*C*** consistently achieves superior model performance, validating the benefit of candidate filtering and reliability calibration.

5.2.3. Evaluation with different Oracle

We use the following instruction-tuned LLMs as comparative models to validate the generality of our findings: Qwen-30B (Qwen3-30B-A3B-Instruct-2507 [35]), Qwen-14B (Qwen-2.5-14B-Instruct [31]), Mistral-7B (Mistral-7B-Instruct-v0.2 [36]), Mixtral-8 × 7B (Mixtral-8x7B-Instruct-v0.1 [36]), and Llama-8B (Llama-3.1-8B-Instruct [37]). Each model produces per-node pseudo-labels and confidence scores given the same instruction template. The results are summarized in in Table 4. Across CiteSeer, Cora, and WikiCS, Qwen-14B offers the most reliable overall trade-off: it has the highest or runner-up test accuracy at both budgets. Mixtral-8 × 7B is the next-best generalist, often achieving runner-up performance and thus a good alternative when deployment favors sparse Mixture-of-Experts inference. Llama-8B shows high annotation accuracy on CiteSeer but weaker downstream gains than others under the same budgets, suggesting that raw annotation accuracy does not fully determine final performance; calibration and alignment with the graph learner still matter. Mistral-7B is the most compute-friendly but consistently trails in both annotation and downstream metrics. The partial Qwen-30B results are competitive but do not clearly surpass Qwen-14B; a larger model or more active parameters do not necessarily guarantee better end performance.

5.2.4. Efficiency and effectiveness of LLM-based annotation

Fig. 7 presents the annotation accuracy and latency of different LLMs across datasets. Notably, the generation and annotation times remain moderate (mostly within 20 s per sample), demonstrating the practical efficiency of our framework in real-world settings. On WikiCS, Qwen-30B delivers the best annotation ACC but with the highest time cost, while Qwen-14B is only 1%–2% lower in ACC at a fraction of the latency. On Cora, the larger Qwen again leads in ACC but keeps the same latency premium; Llama-8B is much faster (< 1 s) with slightly lower ACC. On CiteSeer, the trend flips: Llama-8B attains both the highest ACC and the lowest latency (1 s), suggesting this corpus aligns well with the smaller model's pretraining priors, whereas Qwen-14B also performs strongly with moderate latency (2 s).

In all cases, annotation ACC correlates with but does not deterministically control final node-classification accuracy. For end-to-end efficiency, Qwen-14B is the best general-purpose choice.

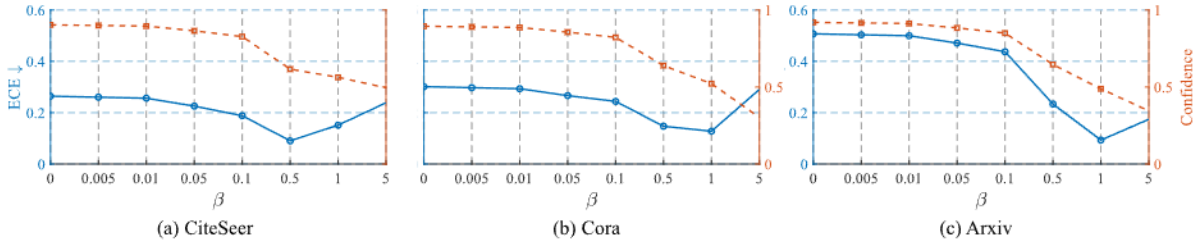


Fig. 8. Calibration analysis under different values of β on test datasets.

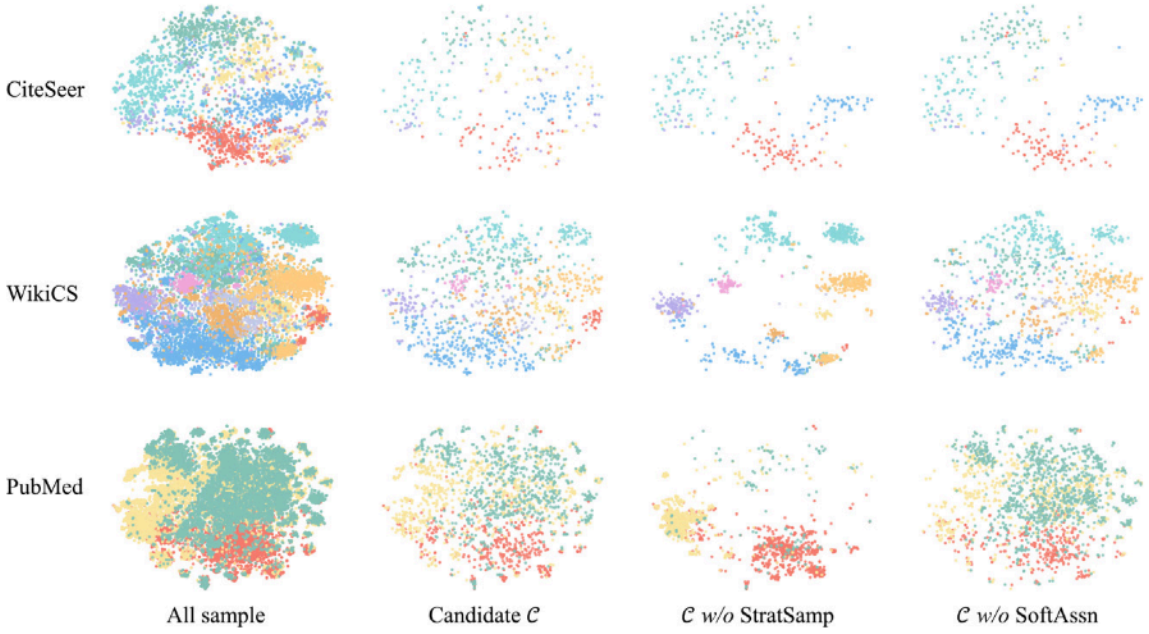


Fig. 9. t-SNE visualization of candidate selection on three datasets.

5.3. Ablation study

To understand the contribution of each component in our framework, we conduct an ablation study on all benchmark datasets, as shown in Table 5.

5.3.1. Graph-LLM confidence calibration

Excluding the ***LLMConf*** module results in drastic degradation, which shows that LLM-generated semantic priors are crucial for label quality. Additionally, removing ***GraphRel***, which propagates semantic reliability through the graph structure, notably hurts performance on homophilous graphs such as Cora, showing that leveraging graph topology is critical under noisy annotations. Fig. 8 shows the ECE and the mean adjusted confidence, ***CalConf***, under different β values. When $\beta = 0$, calibration relies only on ***LLMConf***, leading to overconfidence with high average confidence and large ECE. As β increases, ***GraphRel*** progressively corrects the raw confidence and reduces ECE. A moderate β achieves the best calibration, while an overly large β suppresses confidence too much and increases ECE again. Therefore, an appropriate β is necessary to balance semantic confidence and graph-based reliability for better calibration. Overall, *w/o* ***LLMConf*** causes large accuracy and calibration degradation, while *w/o* ***GraphRel*** also degrades performance, underscoring the value of propagating reliability along the graph topology under noisy annotations.

5.3.2. LLM-generated prototype semantic information

To verify whether LLM-generated prototype semantics can guide sample selection, we construct a variant without LLM-generated prototypes (*w/o* LLMProto). Specifically, we perform k-means clustering on the shallow features $\{\mathbf{x}_i\}_{i=1}^{|\mathcal{V}|}$ using a shallow text encoder, obtain cluster centers, and compute the confusion matrix from these centers. To assess the effectiveness of multi-prototype aggregation in capturing semantic variance, we conduct an ablation by setting $K = 1$ (*w/o* ProtoAvg), where only a single prototype is generated per class. This simplification leads to notable performance degradation across multiple datasets. The resulting confusion matrix becomes overly sensitive to specific prompt phrasing or token noise, thus failing to capture reliable inter-class semantic correlations. This confirms that using multiple prototypes enables more stable and representative class embeddings, which are critical for robust confusion modeling and subsequent confidence calibration.

5.3.3. Candidate pool distillation under component variants

We further analyze the effects of stratified sampling and soft assignment in candidate pool distillation. Removing stratified sampling (*w/o StratSamp*) reduces robustness, especially on CiteSeer, showing that balanced and diverse supervision is important. Replacing soft assignment-based difficulty scoring with raw Euclidean distance (*w/o SoftAssn*) also causes consistent degradation, indicating that probabilistic assignment better captures uncertainty.

Fig. 9 visualizes the selected samples on three datasets. The candidate pool $\mathcal{C}$ covers different semantic clusters more comprehensively,

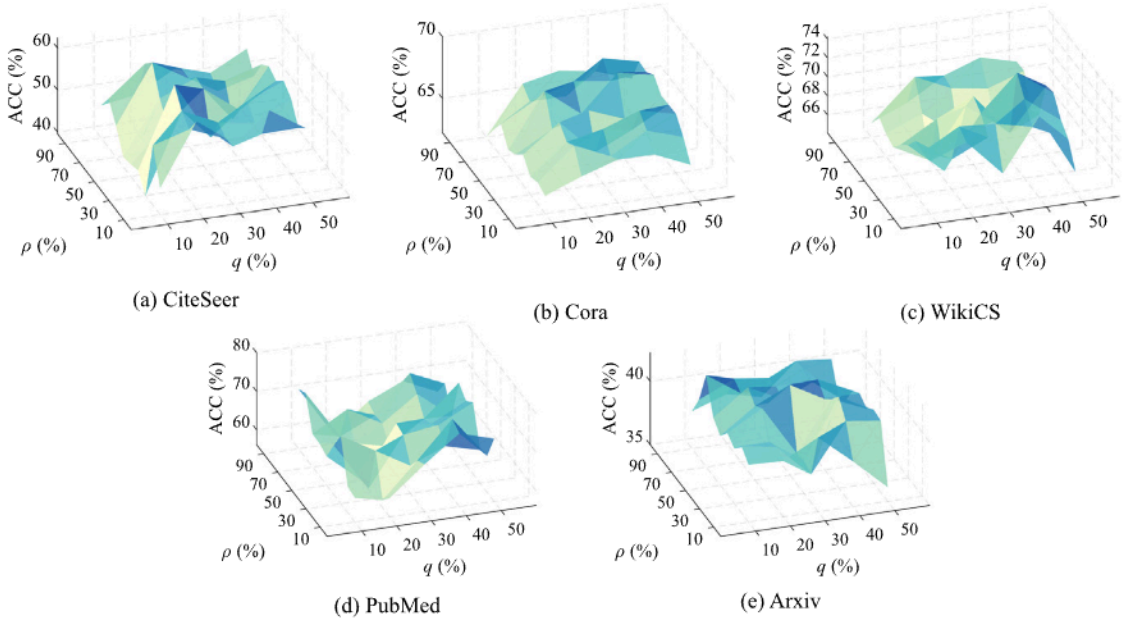


Fig. 10. Parameter sensitivity analysis of core-diversity mixing coefficients ρ and q on test datasets.

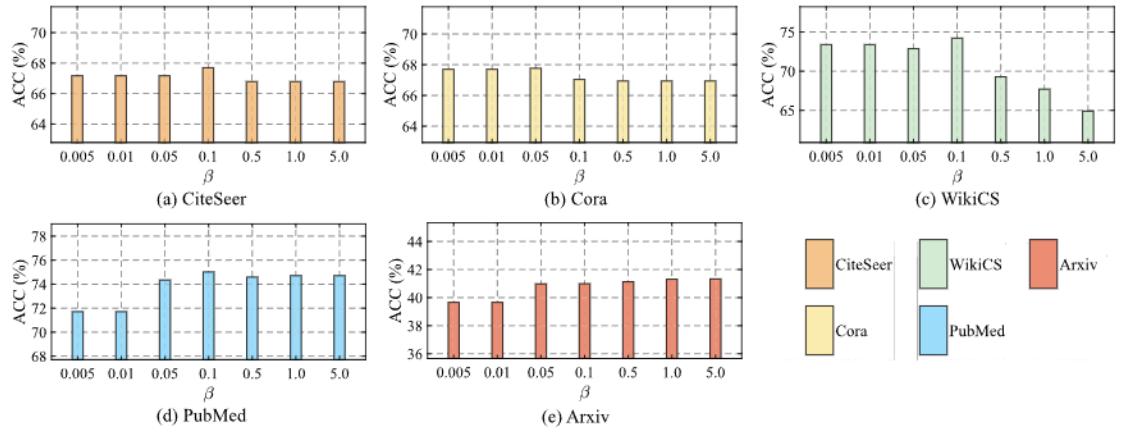


Fig. 11. Parameter sensitivity analysis of β on test datasets.

preserving both representativeness and diversity. In contrast, *C w/o* StratSamp is biased toward dense regions and may underrepresent some classes, while *C w/o* SoftAssn tends to select overly confident samples within narrow cluster areas, reducing intra-class diversity. These results demonstrate that stratified sampling and soft assignment are both necessary for constructing a balanced and informative candidate pool.

5.4. Hyperparameter sensitivity

5.4.1. Impact of core-diversity mixing coefficients ρ and q

The core region C_{core} is taken from the top $q\%$ of the ranked nodes. The boundary region C_{diverse} is drawn from the next ($q\%$) block immediately following C_{core}. Given a target candidate size $|C|$, we sample $\lfloor \rho|C| \rfloor$ nodes from C_{core} and use the remainder from C_{diverse}. We sweep $\rho \in [10\%, 90\%]$ and $q \in [10\%, 50\%]$ in Fig. 10 to study how the core-diversity mix and the ranking cutoff affect the candidate pool. Across all datasets, accuracy exhibits a clear, non-monotonic dependence on the core-diversity mixing ratio ρ and the ranking cutoff q: performance typically peaks when both are set to mid-range values, rather than at extremes. Extremely small q over-focuses on redundant easy annotation instances, whereas very large q admits noisy boundary points; likewise, very low ρ under-anchors the selection, and very high ρ under-explores decision boundaries. Concretely, the most reliable region is $q \in [20\%, 40\%]$ with $\rho \in [30\%, 60\%]$, which balances high-confidence core exemplars and boundary-diverse samples.

5.4.2. Impact of trade-off hyperparameter β

We investigate the sensitivity of the hyperparameter β, which controls the trade-off between local annotation confidence (***LLMConf***) and graph-aware reliability (***GraphRel***). As shown in Fig. 11, our method exhibits consistent robustness across a wide range of β values. As β increases further, performance begins to degrade, suggesting that overemphasizing structure may dilute the benefit of semantic guidance from the LLM. These patterns suggest that β effectively controls the strength of confidence correction: a too-small β leaves overconfident errors insufficiently corrected; moderate values balance calibration and discrimination; too large pushes the model toward overly conservative decisions. In practice, we recommend sweeping $\beta \in \{0.01, 0.1\}$, which provides a simple yet effective search range for balancing calibration

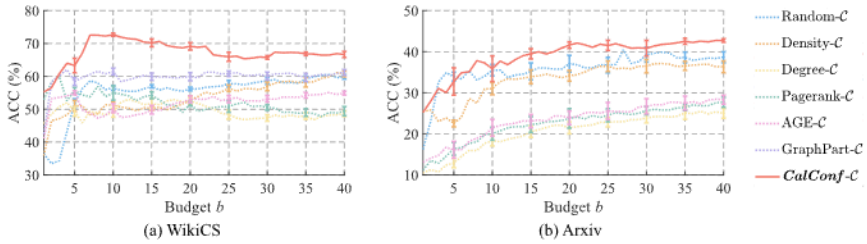


Fig. 12. Test accuracy comparison under varying budgets b on **WikiCS** and **Arxiv** datasets.

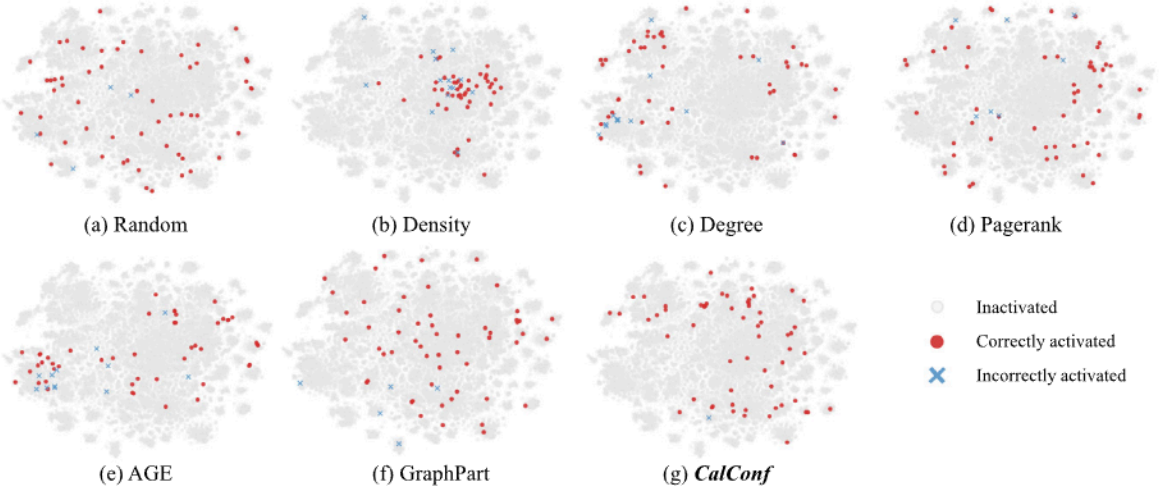


Fig. 13. Node selection distribution across different strategies on the **PubMed** dataset under budget b=20.

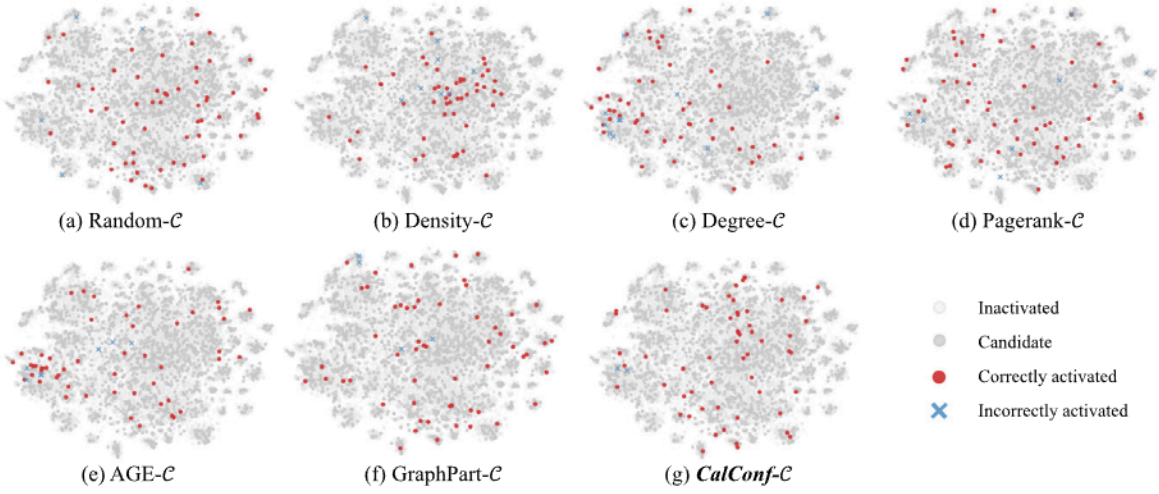


Fig. 14. Node selection distribution across different strategies with candidate $\mathcal{C}$ on the **PubMed** dataset under budget b=20.

and semantic confidence. These results confirm the importance of balancing semantic and structural cues for reliable node selection under noisy LLM-generated supervision.

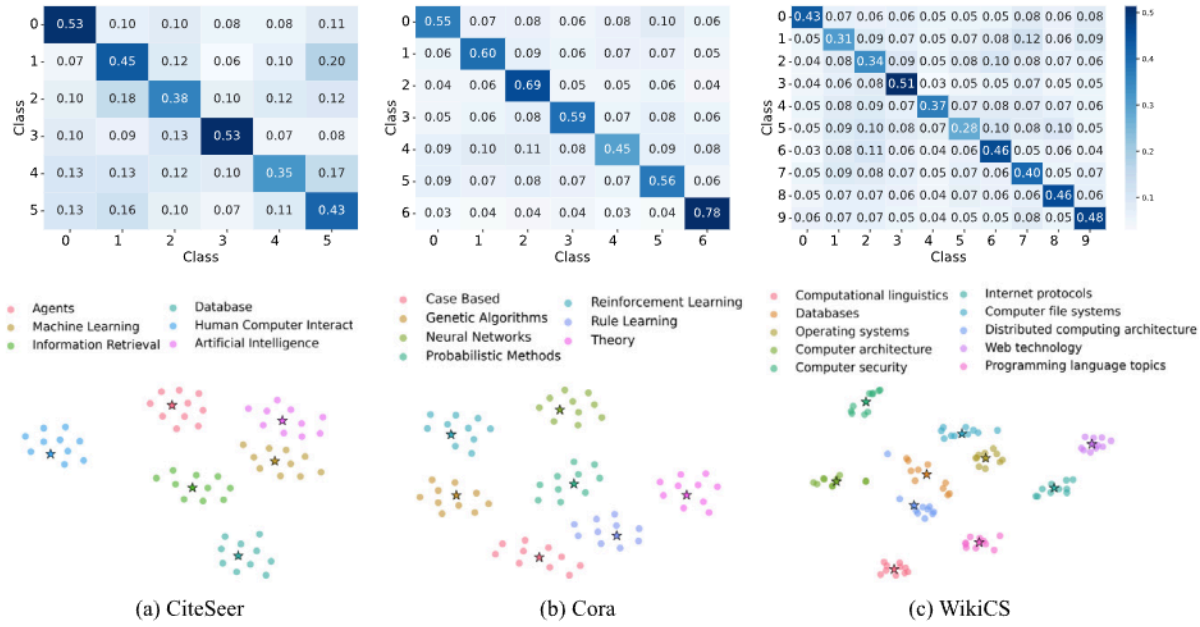
5.4.3. Impact of labeling budgets b

Furthermore, we study the performance of different node selection methods on two datasets under different labeling budgets, as shown in Fig. 12. All methods generally improve as the annotation budget increases, while ***CalConf***-$\mathcal{C}$ consistently outperforms the baselines across all budget levels. Notably, it achieves substantially higher accuracy in the low-budget regime. These findings demonstrate the effectiveness of our reliability-aware scoring function for sample selection.

5.5. Interpretability

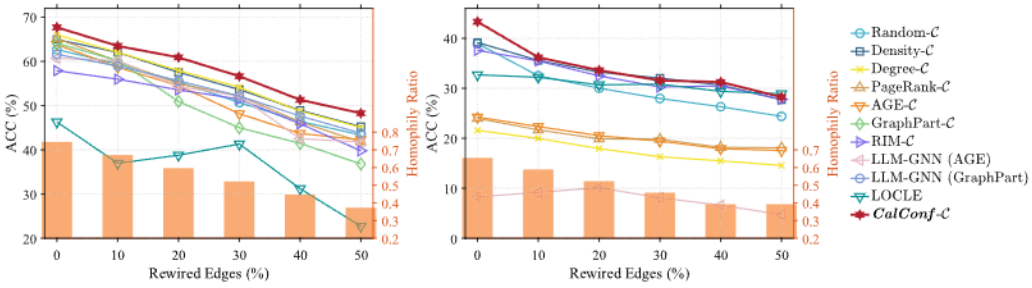
5.5.1. Node selection behavior

Fig. 13 illustrates the node selection behavior of various strategies. In each subgraph, red dots represent correctly selected nodes, while blue crosses indicate misclassified ones. Traditional strategies such as Random, Density, and Degree tend to concentrate selections around



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

Fig. 15. Top: The prototype-based semantic confusion matrices. Bottom: t-SNE visualizations of prototype embeddings.



(a) CiteSeer (b) Arxiv

Fig. 16. Performance comparison under different homophily ratio.

local hubs, resulting in limited structural coverage and high sample redundancy. By contrast, methods like GraphPart exhibit more dispersed selection patterns, capturing a wider range of graph structures. By contrast, ***CalConf*** demonstrates a more diverse and well-distributed selection pattern, with significantly fewer mis-activations.

In Fig. 14, integrating the candidate set $\mathcal{C}$ effectively filters out duplicated or less informative nodes. As a result, the selected samples are more diverse and representative, enabling more effective model updates under limited annotation budgets. Furthermore, the calibrated confidence mechanism in ***CalConf***-$\mathcal{C}$ helps suppress noisy activations and enhances selection reliability. This highlights the effectiveness of our confidence calibration and selection mechanisms.

5.5.2. Prototype embedding visualization

Fig. 15 shows the soft assignment probabilities between class prototypes, reflecting inter-class semantic ambiguity captured by the LLM. Strong diagonal mass signals self-alignment of a class, while structured off-diagonal blocks reveal systematic confusions between semantically related labels. Furthermore, the embeddings of LLM-generated prototype samples show that most class prototypes form distinguishable clusters, indicating that LLM-generated descriptions provide meaningful semantic anchors for class-level modeling. Moreover, the varying cluster compactness and inter-class distances imply different degrees of intra-class consistency and inter-class ambiguity. These observations support the use of prototype embeddings for estimating a label-free semantic confusion matrix.

5.5.3. Robustness under low homophily

To examine whether the proposed graph-aware reliability module overly relies on the homophily assumption, we conduct a low-homophily test. Edge homophily is measured as the proportion of edges whose two endpoints share the same label: $\text{Homophily}(\mathcal{G}) = |\{(u,v) \mid (u,v) \in \mathcal{E},\ y_u = y_v\}|/|\mathcal{E}|$. A larger value indicates stronger homophily, while a smaller value indicates a more heterophilous graph. Specifically, for an edge (u,v) satisfying $y_u = y_v$, we keep the source node u unchanged and replace the destination node v with a randomly sampled node from a different class. We vary the rewiring ratio from 10% to 50% and evaluate different node selection methods under the same annotation budget.

Fig. 16 reports the results on Arxiv and CiteSeer under reduced homophily. As the homophily ratio decreases, the performance of most methods deteriorates. For example, on Arxiv, the homophily ratio decreases from 0.654 to 0.393, and ***CalConf***-$\mathcal{C}$ drops from 43.32% to 28.24%; on CiteSeer, the homophily ratio decreases from 0.745 to 0.372 when the rewiring ratio reaches 50%. Despite this degradation, it remains competitive with the strongest baselines. This shows that while the proposed method benefits from meaningful graph structure, it does not blindly trust all neighbors. By combining neighbor reliability, semantic compatibility, and embedding-based edge similarity, ***GraphRel*** downweights inconsistent neighbors, enabling ***CalConf*** to degrade robustly under weakened homophily.

6. Conclusion

In this work, we investigated the problem of reliable node selection under noisy LLM-generated annotations for text-attributed graphs. We proposed a confidence-calibrated node selection framework that integrates prototype-based confusion modeling with graph-aware reliability propagation. By distilling a structurally diverse candidate set and refining confidence scores through topology-aware calibration, our approach enables robust pseudo-label selection under limited annotation budgets. The key contribution of this framework lies in its principled modeling of annotation noise. Rather than directly trusting LLM outputs, we approximate their intrinsic confusion patterns via a prototype-based confusion matrix and leverage local graph structure to construct a more reliable and topologically consistent confidence score. Empirical results across multiple public benchmarks demonstrate consistent improvements over classical selection strategies and competitive graph learning baselines. However, several limitations remain: the confusion estimation depends on LLM-generated prototypical samples, which may introduce bias. In addition, the current design is tailored to static node classification and does not directly extend to dynamic or heterogeneous graphs. Future work will investigate theoretical guarantees for noise-aware selection mechanisms and extensions to heterogeneous graph settings.

CRediT authorship contribution statement

Zihan Fang: Writing – original draft, Methodology, Investigation. **Shide Du:** Writing – review & editing, Methodology. **Zhihao Wu:** Writing – review & editing, Investigation, Formal analysis. **Zhiling Cai:** Writing – review & editing, Formal analysis. **Yanchao Tan:** Writing – review & editing. **Shiping Wang:** Supervision, Resources, Funding acquisition. **Zhouchen Lin:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part supported by the National Natural Science Foundation of China under Grants U25A20527, 62276065 and 62276004, the Fujian Provincial Natural Science Foundation of China under Grants 2024J01510026 and 2025J01585, and the Beijing Major Science and Technology Project under Grant Z251100008425006.

Data availability

The data that has been used is confidential.

References

[1] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, J. Han, Large language models on graphs: A comprehensive survey, IEEE Trans. Knowl. Data Eng. 36 (12) (2024) 8622–8642.
[2] L. Song, W. Tu, S. Zhou, E. Zhu, GANN: Graph alignment neural network for semi-supervised learning, Pattern Recognit. 154 (2024) 110484.
[3] Y. Xia, S. Mukherjee, Z. Xie, J. Wu, X. Li, R. Aponte, H. Lyu, J. Barrow, H. Chen, F. Dernoncourt, et al., From selection to generation: A survey of llm-based active learning, in: Proc. ACL, 2025, pp. 14552–14569.
[4] Q. Gu, J. Han, Towards active learning on graphs: An error bound minimization approach, in: IEEE ICDM, 2012, pp. 882–887.
[5] L. Cui, X. Tang, S. Katariya, N. Rao, P. Agrawal, K. Subbian, D. Lee, Allie: Active learning on large-scale imbalanced graphs, in: Proc. ACM Web Conf., 2022, pp. 690–698.
[6] Z. Chen, H. Mao, H. Wen, H. Han, W. Jin, H. Zhang, H. Liu, J. Tang, Label-free node classification on graphs with large language models (LLMs), in: ICLR, 2024, pp. 1–13.
[7] B. Pan, Z. Zhang, Y. Zhang, Y. Hu, L. Zhao, Distilling large language models for text-attributed graph learning, in: Proc. ACM Int. Conf. Inf. Knowl. Manag., 2024, pp. 1836–1845.
[8] Y. Wu, J. Yao, X. Xia, J. Yu, R. Wang, B. Han, T. Liu, Mitigating label noise on graphs via topological sample selection, in: ICML, 2024, pp. 1–14.
[9] S. Chen, Q. Zhang, J. Dong, W. Hua, Q. Li, X. Huang, Entity alignment with noisy annotations from large language models, NeurIPS 37 (2024) 15097–15120.
[10] Z. Sheng, W. Guo, Y. Shao, W. Zhang, B. Cui, Llms are noisy oracles! llm-based noise-aware graph active learning for node classification, in: Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min., 2025, pp. 2526–2537.
[11] L. Ye, A. Shah, C. Zhang, S. Chava, Calibrating pre-trained language classifiers on LLM-generated noisy labels via iterative refinement, in: Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min., 2025, pp. 3598–3609.
[12] Z. Zhou, R. Li, W. Ai, X. Li, Z. Teng, B. Zhang, J. Du, Affinity-aware uncertainty quantification for learning with noisy labels, Pattern Recognit. (2025) 112495.
[13] Y. Li, Z. Xie, H. Zhong, G. Gao, Noise-tolerant scheme and explicit regularizer for deep active learning with noisy oracles, Pattern Recognit. (2025) 112313.
[14] J. Ma, Z. Ma, J. Chai, Q. Mei, Partition-based active learning for graph neural networks, Trans. Mach. Learn. Res. (ISSN: 2835-8856) (2023).
[15] H. Cai, V.W. Zheng, K.C.-C. Chang, Active learning for graph embedding, 2017, arXiv preprint arXiv:1705.05085.
[16] W. Zhang, Y. Wang, Z. You, M. Cao, P. Huang, J. Shan, Z. Yang, B. Cui, Rim: Reliable influence-based active learning on graphs, NeurIPS 34 (2021) 27978–27990.
[17] W. Yang, S. Zhang, C. Ye, J. Guo, T. Xu, Z. Huang, Know your neighbors: Subgraph importance sampling for heterophilic graph active learning, in: Proc. AAAI, vol. 40, no. 33, 2026, pp. 27630–27638.
[18] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, H. Liu, Large language models for data annotation and synthesis: A survey, in: Proc. EMNLP, 2024, pp. 930–957.
[19] R. Zhang, Y. Li, Y. Ma, M. Zhou, L. Zou, LLMaAA: Making large language models as active annotators, in: Findings of ACL: EMNLP, 2023, pp. 13088–13103.
[20] H. Song, M. Kim, D. Park, Y. Shin, J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, IEEE Trans. Neural Netw. Learn. Syst. 34 (11) (2022) 8135–8153.
[21] T. Zhang, R. Yang, Y. Lai, M. Yan, X. Ye, D. Fan, Leveraging large language models for effective label-free node classification in text-attributed graphs, in: Proc. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2025, pp. 698–708.
[22] R. Xu, K. Ding, GNN-as-judge: Unleashing the power of LLMs for graph learning with GNN feedback, in: ICLR, 2024, pp. 1–14.
[23] M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, B. Hooi, Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, in: ICLR, vol. 2024, 2024, pp. 23650–23678.
[24] B. Yuan, Y. Chen, Y. Zhang, W. Jiang, Hide and seek in noise labels: Noise-robust collaborative active learning with llms-powered assistance, in: Proc. ACL, 2024, pp. 10977–11011.
[25] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: A loss correction approach, in: Proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2017, pp. 1944–1952.
[26] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, M. Sugiyama, Are anchor points really indispensable in label-noise learning? NeurIPS 32 (2019) 6838–6849.
[27] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, M. Sugiyama, Dual t: Reducing estimation error for transition matrix in label-noise learning, NeurIPS 33 (2020) 7260–7271.
[28] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: ICML, 2016, pp. 478–487.
[29] M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, M. Lucic, Revisiting the calibration of modern neural networks, NeurIPS 34 (2021) 15682–15694.
[30] Z. Jiang, J. Araki, H. Ding, G. Neubig, How can we know when language models know? On the calibration of language models for question answering, Trans. Assoc. Comput. Linguist. 9 (2021) 962–977.
[31] Qwen Team, Qwen2.5: A party of foundation models, 2024, URL https://qwenlm.github.io/blog/qwen2.5/.
[32] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93–93.
[33] P. Mernyei, C. Cangea, Wiki-cs: A wikipedia-based benchmark for graph neural networks, 2020, arXiv preprint arXiv:2007.02901.
[34] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, J. Leskovec, Open graph benchmark: Datasets for machine learning on graphs, NeurIPS 33 (2020) 22118–22133.
[35] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, 2025, arXiv preprint arXiv:2505.09388.
[36] A.Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D.S. Chaplot, D.d.l. Casas, E.B. Hanna, F. Bressand, et al., Mixtral of experts, 2024, arXiv preprint arXiv:2401.04088.
[37] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, 2024, arxiv e-prints arXiv–2407.