# LLM–GNN Co-Teaching for Few-Shot Graph Learning

**Zhuoyi Peng**
The Hong Kong University
of Science and Technology
zpengaf@connect.ust.hk

**Hanlin Gu**
The Hong Kong University
of Science and Technology
ghltsl123@gmail.com

**Lixin Fan**
WeBank
lixin.fan01@gmail.com

**Yi Yang**
The Hong Kong University
of Science and Technology
imyiyang@ust.hk

## Abstract

Text-attributed graphs (TAGs) underlie real-world applications such as citation networks, social media, and e-commerce. Few-shot graph learning on TAGs is hard: with only a handful of labels per class and the rest of the graph unannotated, neither GNNs nor LLMs can learn well on their own. GNNs read topology and fail on cold nodes; LLMs read text and fail on text-ambiguous nodes. Existing LLM-GNN methods all follow the same recipe: *designate one model as the golden teacher and use its outputs (e.g., features or pseudo-labels) to supervise the other.* We argue this golden-teacher assumption breaks under sparse supervision: neither model is golden, and treating either as such transfers its blind spots into the student. We therefore ask: *can we avoid designating either model as the golden teacher, and still perform effective graph learning?* We answer with LLM-GNN Co-Teaching, a bidirectional co-teaching framework in which neither model is fixed as teacher. The GNN and LLM exchange their most confident pseudo-labels under an architecture-specific small-loss criterion, and both update every round. Supervision is then mined from the trajectory: whenever a node moves from cross-model contradiction at round $t$ to cross-model agreement at round $t+1$, the LLM's two answers on the same input form a preference pair for DPO training. We call this Round-based Pseudo-Label Preference Optimization (RPL-PO). On six benchmarks, LLM-GNN Co-Teaching consistently outperforms GNN-as-Judge and all prior methods; improvements carry over to 5-shot and to zero-shot cross-dataset transfer. Error-structure analysis further shows that abandoning the golden-teacher assumption substantially improves the LLM's graph learning capability on challenging samples. Code: `https://github.com/realjeremybot/LLM-GNN-Coteaching`.

## 1 Introduction

Text-attributed graphs (TAGs) [1–4] underlie real-world applications such as citation networks, social media, recommendation, and e-commerce, where each node carries raw text alongside graph topology. The rise of Large Language Models (LLMs) [5–7] has driven growing interest in using them for TAG learning [4, 8–12]. Most existing work on TAG learning, however, focuses on the supervised setting where abundant labels are available and both models can be reliably fine-tuned [4, 9–14]. Real-world TAGs are rarely labeled at this scale: only a handful of labels per class are typically available, and the bulk of the graph carries no supervision [15–19]. Under this few-shot regime, neither GNNs [20–23] nor LLMs work well alone: GNNs read topology and fail on cold (low-degree)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

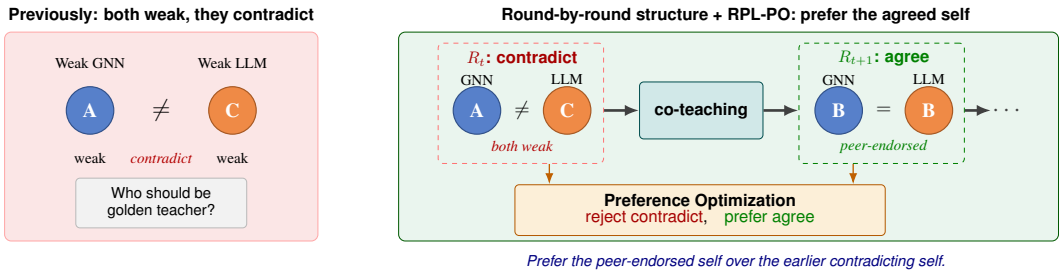


Figure 1: **Co-teaching without a golden teacher.** A single round between two weak models leaves them contradicting on node $v$, with no way to choose which should serve as the golden teacher (left). After one more round of bidirectional co-teaching, both models evolve, and if they now agree on $B$, the LLM's $R_t$ contradicting answer $C$ and $R_{t+1}$ peer-endorsed answer $B$ together form a preference pair: the earlier self is rejected, the peer-endorsed self is preferred (right). The reward signal comes from the trajectory itself. No golden teacher, no human label, no reward model, no external judge.

nodes whose neighborhoods provide too little signal [24, 25], while LLMs read text and fail when text is short or class-ambiguous [26–29]. Their disjoint failure modes have motivated a substantial line of work combining them.

Existing LLM-GNN methods share a common structure: one model is designated as a fixed teacher whose outputs are treated as ground truth, and the other is trained to match those outputs. We refer to this shared structural assumption as the *golden-teacher assumption*. Prior approaches differ only in which side is designated as golden. *LLM-as-Enhancers* [4, 13, 30, 31] freeze LLM-derived features or explanations and train a downstream GNN to imitate them. *LLM-as-Predictor* methods [9–12, 14, 32, 33] treat the once-instruction-tuned LLM as the golden predictor, typically prompting it with structural tokens. *GNN-as-Judge* [34] reverses the direction: a once-trained GNN's verdicts filter or re-weight pseudo-labels for LLM fine-tuning. In every case, supervision flows in one direction from a fixed teacher, and the student has no way to revise what the teacher said.

The golden-teacher assumption breaks under sparse supervision. **With few labels per class, neither model is reliable enough to serve as the golden teacher: the GNN cannot learn good representations for cold nodes, and the LLM cannot disambiguate short or class-ambiguous text without exemplars.** Treating either as golden transfers its blind spots into the student wholesale, and unidirectional supervision leaves the student no way to revise what the teacher said. The question that few-shot LLM-GNN learning has not yet asked is therefore:

> *Can we avoid designating either model as the golden teacher, and still perform effective graph learning?*

The question is not trivial: with only a few labeled anchors as direct supervision, two weak models updating each other freely can collapse onto each other's mistakes rather than converge toward truth. The framework needs a mechanism that extracts reliable supervisory signal from their joint dynamics.

Our answer is LLM-GNN Co-Teaching, a co-teaching framework that does not designate either side as the golden teacher and instead lets the GNN and LLM evolve together. Training proceeds in rounds: in each round, every peer extracts its most confident pseudo-labels under an architecture-specific small-loss criterion (cross-entropy fit for the GNN, minimum token log probability for the LLM) and passes them to the other model, so that both peers grow from weak to strong round by round. To create additional supervision, we further mine a preference signal from this trajectory: whenever a node transitions from cross-model contradiction at round $t$ to cross-model agreement at round $t+1$, the LLM's two answers on the same node, the earlier contradicting one and the later peer-endorsed one, form a natural preference pair, which we feed to direct preference optimization (DPO) [35]. We call this Round-based Pseudo-Label Preference Optimization (RPL-PO). The reward signal comes from the trajectory itself: no golden teacher, no human label, no reward model, no external judge.

**Contributions.** (1) **We abandon the golden-teacher assumption.** LLM-GNN Co-Teaching is the first LLM-GNN method in which neither model is designated as authoritative, with both updating every round and supervising each other through a small-loss criterion. (2) **RPL-PO: a self-supervised**

**preference-pair generator.** A node that transitions from cross-model contradiction at round $t$ to cross-model agreement at round $t+1$ yields a DPO preference pair from the LLM's two answers on the same input. RPL-PO requires no human labels, no reward models, and no external judges, and is structurally inaccessible to single-round or frozen-teacher pipelines. (3) **State-of-the-art on six benchmarks.** LLM-GNN Co-Teaching achieves absolute accuracy gains of $7.57\%$ on Cora and $6.99\%$ on ogbn-arxiv over GNN-as-Judge under 3-shot supervision, with the same lead carrying over to 5-shot and to zero-shot cross-dataset transfer. The error-structure analysis in §5.6 shows that abandoning the golden-teacher assumption substantially improves the LLM's graph learning capability on challenging samples.

## 2 Related Work

**LLM-GNN methods for graph learning.** Combining LLMs and GNNs for TAGs has been extensively explored [4, 8–10, 26–29, 36, 37]. *LLM-as-Enhancers* [4, 13, 30, 31] freeze LLM-derived features or explanations as enriched node input to a downstream GNN. *LLM-as-Predictors* [8–12, 14, 32, 33] frame node classification as text generation, typically with structural prompts or graph tokens. *GNN-as-Judge* [34] reverses the direction: a once-trained GNN's verdicts filter pseudo-labels for fine-tuning the LLM, with a theoretical lower bound on agreement-set accuracy under conditional independence; Sheng et al. [38] similarly treats LLM annotations as noisy oracles for graph active learning. In every case, one model is fixed as the golden teacher and supervision flows in one direction. LLM-GNN Co-Teaching instead designates no golden teacher: both models update every round and judge each other across multiple rounds.

**Co-teaching, noisy labels, and pseudo-label selection.** Co-teaching [39] trains two networks simultaneously, each selecting small-loss samples for its peer. Co-Teaching+ [40] adds disagreement filtering, DivideMix [41] introduces mixture-model selection, and earlier co-training [42, 43] variants pair networks of the same architecture. The broader noisy-label literature [44–48] likewise treats noise as homogeneous across views, and recent work warns that LLMs trained on their own outputs can degrade over time [49]. Closely related is pseudo-labeling [50], which augments small labeled sets with model-generated labels, with mining of both easy and hard samples shown to be crucial [51, 52]. On graphs, prior work explores multi-stage self-training [19], label-propagation hybrids [24, 53], confidence-aware filtering [54], and active labeling [55, 56], with single-round GNN-LLM agreement filters [34] the closest to our setup. All of this prior work pairs homogeneous networks of the same architecture and uses a single-round selection. We are the first to co-teach across *heterogeneous* architectures (GNN + LLM) iteratively, whose complementary inductive biases (structural vs. semantic) provide stronger error independence than random-initialization diversity.

**Preference optimization.** LLM alignment from feedback originates in RLHF [57–59], with DPO [35] and its variants [60–65] replacing the reward model with pairwise preferences. On graphs, GNN-as-Judge [34] and InstructGraph [14] apply preference tuning to within-round GNN-LLM disagreements; RPL-PO instead exploits the *temporal* structure of co-teaching, contrasting the same LLM's predictions across consecutive rounds.

**Our novelty.** LLM-GNN Co-Teaching contributes on two fronts. (i) *A new LLM-GNN framework without an explicitly designated golden teacher*, operationalized via heterogeneous co-teaching in which the GNN and LLM iteratively pseudo-label each other under a small-loss criterion and both update every round. (ii) *A novel preference optimization signal mined from the learning trajectory*, which constructs preference pairs from cross-round agreement transitions and requires no golden teacher, no human label, no reward model, and no external judge, fully releasing the supervision signal latent in sparse-label graph learning.

## 3 Preliminaries

We consider few-shot semi-supervised node classification on text-attributed graphs (TAGs) defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{T})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, and $\mathbf{T} = \{t_v\}_{v \in \mathcal{V}}$ are the per-node text attributes (e.g. paper title and abstract in citation networks). Each node $v$ has a label $y_v \in \{1, \ldots, C\}$. Given a small labeled set $\mathcal{V}_{\text{train}}$ in which up to $k$ labeled nodes are sampled per class, along with a validation set $\mathcal{V}_{\text{val}}$, the goal is to predict labels for the test nodes

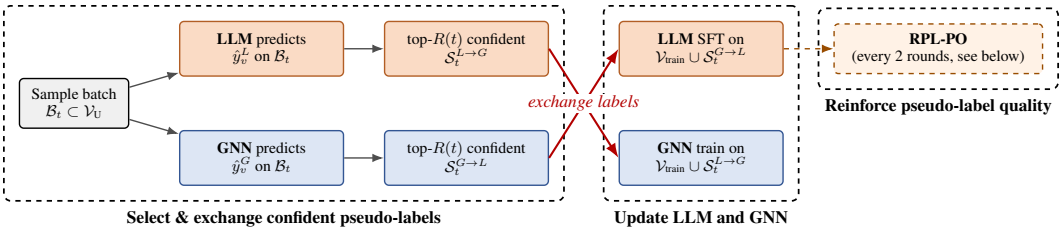


Figure 2: Round-$t$ co-teaching flow. The **GNN** and **LLM** predict on the same unlabeled batch $\mathcal{B}_t$ and score by a model-specific self-confidence signal. The top-$R(t)$ confident pairs are cross-exchanged: $\mathcal{S}_t^{G\to L}$ supervises the LLM in §4.2, and $\mathcal{S}_t^{L\to G}$ supervises the GNN in §4.2. Every two rounds the LLM additionally undergoes RPL-PO (zoomed in Fig. 3, §4.3).

$\mathcal{V}_{\text{test}} = \mathcal{V} \setminus (\mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}})$. Few-shot *semi-supervised* learning trains the models on both (i) the small labeled set $\mathcal{V}_{\text{train}}$ together with its ground-truth labels and (ii) the unlabeled nodes $\mathcal{V}_U \subseteq \mathcal{V} \setminus \mathcal{V}_{\text{train}}$ and their text contents, e.g., by predicting on $\mathcal{V}_U$ and using the most confident predictions as pseudo-labels.

**LLM predictor.** A large language model $f_{\text{LLM}}$ predicts node labels by framing classification as text generation. Given node $v$, we build a prompt $\mathcal{P}(v)$ containing $v$'s text $t_v$, the texts of $v$'s neighbors, and the candidate class names. The LLM emits a class name, giving prediction $\hat{y}_v^L = f_{\text{LLM}}(\mathcal{P}(v))$.

**GNN predictor.** A graph neural network $f_{\text{GNN}} : \mathcal{V} \to \mathbb{R}^C$ produces class logits by aggregating information from a node's local neighborhood via message passing. Layer $\ell$ computes

$$\mathbf{h}_v^{(\ell)} = \text{UPDATE}^{(\ell)}\left(\mathbf{h}_v^{(\ell-1)},\ \text{AGG}^{(\ell)}\left(\{\mathbf{h}_u^{(\ell-1)} : u \in \mathcal{N}(v)\}\right)\right), \tag{1}$$

with $\mathbf{h}_v^{(0)} = \mathbf{x}_v$ (a numerical embedding of $t_v$) and final prediction $\hat{y}_v^G = \arg\max f_{\text{GNN}}(v)$.

## 4 LLM-GNN Co-teaching

**Why no golden teacher?** Existing LLM-GNN methods all designate one model as a fixed teacher whose outputs are treated as ground truth. Under sparse supervision, this assumption breaks down: with few labels per class, neither model is reliable, so freezing either side as the teacher transfers its blind spots into the student wholesale and the unidirectional supervision leaves no path to revise mistakes. Effective few-shot LLM-GNN learning therefore needs a framework where neither model is fixed as teacher and both can correct each other through their joint dynamics.

**Difference from classical co-teaching.** Classical Co-Teaching [39] pairs two networks of the same architecture and relies on random-init diversity for the two peers to make different mistakes; this diversity is fragile and shrinks as the two peers converge. LLM-GNN Co-Teaching instead pairs a GNN and an LLM, whose inductive biases differ: the LLM's text-based view includes neighbor context and fails on semantically ambiguous descriptions, while the GNN's structural view fails on low-degree, sparsely connected nodes. Each peer is reliably strong where the other is weak, so confident pseudo-labels from one carry information the other could not have produced alone, providing a far more durable basis for mutual correction than random initialization. §5.6 confirms this complementarity empirically through per-degree error rates.

LLM-GNN Co-Teaching runs for $T$ rounds. Figure 2 visualises one round, organised in three stages.

1. **Select and exchange confident pseudo-labels** (§4.1). Both models predict on the same unlabeled batch, each picks its most confident subset, and the two subsets are swapped.
2. **Update the LLM and GNN** (§4.2). The LLM is fine-tuned on the GNN-selected pseudo-labels and the GNN is trained on the LLM-selected pseudo-labels, both starting from the previous round's checkpoint.
3. **Reinforce pseudo-label quality with RPL-PO** (§4.3). Every two rounds, nodes whose LLM prediction flipped from disagreeing-with-GNN to agreeing-with-GNN form a temporal preference pair. The LLM is then updated by DPO on these pairs, with no human annotation or external reward model.

### 4.1 Selecting and Exchanging Confident Pseudo-Labels

**Intuition.** Two weak models can still teach each other if they are honest about what they do not know. We therefore let each model speak only on the nodes where it is confident, and we send those confident answers to the peer as supervision; the peer treats them as ground truth on *its* blind spots. Concretely, each model passes to its peer only the pseudo-labels it is most certain about. Certainty is measured by how strongly the model's own scoring agrees with the answer it just produced, following the small-loss principle of Co-Teaching [39]. An LLM that hesitates on some token of its generated answer, or a GNN that poorly fits its own predicted class, is more likely to be wrong than a model that scores its own answer with high confidence.

For each node $v \in \mathcal{B}_t$ we compute a self-confidence signal,

$$\ell_v^G \;=\; \ell_{\text{CE}}\big(f_{\text{GNN}}(v),\, \hat{y}_v^G\big) \quad \text{(GNN)}, \qquad d_v \;=\; \min_j \log p_{\text{LLM}}(w_j \mid w_{<j}) \quad \text{(LLM)},$$

where low $\ell_v^G$ means the GNN strongly fits the label it just produced and high $d_v$ means no token along the LLM's produced answer was uncertain. We keep the most confident top-$R(t)$ fraction. The selection ratio $R(t) \in (0, 1]$ is annealed linearly from $R_{\min}$ at $t{=}1$ to $R_{\max}$ at $t{=}T$,

$$R(t) \;=\; R_{\min} + (R_{\max} - R_{\min}) \cdot \frac{t-1}{T-1}, \tag{2}$$

so early rounds (when both models are still weak) exchange few but reliable pseudo-labels, while later rounds exchange more. The two paired sets $\mathcal{S}_t^{G\to L} = \{(v,\, \hat{y}_v^G)\}$ and $\mathcal{S}_t^{L\to G} = \{(v,\, \hat{y}_v^L)\}$ resulting from this exchange feed the model updates that follow.

### 4.2 Updating the LLM and GNN

**Intuition.** The exchanged pseudo-labels are the peer's best guesses on the receiver's weak spots, so each model now has access to supervision it could not have produced alone. We update both models on the same training data structure, anchors plus peer-supplied pseudo-labels, but with model-specific losses. Both updates start from the previous round's checkpoint, so each model carries forward the gains from earlier rounds.

**LLM SFT.** At round $t$, the LLM $f_{\text{LLM}}$ is fine-tuned on the labeled anchors $\mathcal{V}_{\text{train}}$ together with the GNN-selected pseudo-label set $\mathcal{S}_t^{G\to L}$, starting from the previous round's adapter. Let $\hat{y}$ denote the supervision target: the ground-truth label $y_v$ when $v \in \mathcal{V}_{\text{train}}$, and the GNN's prediction $\hat{y}_v^G$ when $v \in \mathcal{S}_t^{G\to L}$. Let $\hat{y}_{<i}$ denote its first $i-1$ tokens. We minimise the standard token-level cross-entropy

$$\mathcal{L}_{\text{LLM}}^{(t)} \;=\; -\sum_{(v,\hat{y})\,\in\,\mathcal{V}_{\text{train}}\,\cup\,\mathcal{S}_t^{G\to L}} \;\sum_{i=1}^{|\hat{y}|} \log p_{\text{LLM}}\big(\hat{y}_i \mid \mathcal{P}(v),\, \hat{y}_{<i}\big), \tag{3}$$

with $p_{\text{LLM}}$ the LLM's next-token distribution and $\mathcal{P}(v)$ the prompt defined in §3.

**GNN training.** The GNN is trained on the few ground-truth anchors and the larger LLM-selected pseudo-label set. Naively concatenating the two would let the larger pseudo-label set drown out the anchor signal, so we average each loss within its own set and combine them with a round-dependent weight. This gives a convex combination of an anchor loss and a pseudo-label loss:

$$\mathcal{L}_{\text{GNN}}^{(t)} \;=\; (1{-}\alpha_t)\cdot \underbrace{\frac{1}{|\mathcal{V}_{\text{train}}|}\sum_{(v,y)\in\mathcal{V}_{\text{train}}} \ell_{\text{CE}}(f_{\text{GNN}}(v), y)}_{\text{anchor loss}} \;+\; \alpha_t \cdot \underbrace{\frac{1}{|\mathcal{S}_t^{L\to G}|}\sum_{(v,\hat{y}_v^L)\in\mathcal{S}_t^{L\to G}} \ell_{\text{CE}}(f_{\text{GNN}}(v), \hat{y}_v^L)}_{\text{pseudo-label loss}}. \tag{4}$$

The mixing weight $\alpha_t$ is annealed linearly from $\alpha_0$ at round 1 to $\alpha_{\max}$ at round $T$. Early rounds emphasise the anchor signal. Later rounds, when the LLM produces cleaner pseudo-labels, weight those more heavily, while the within-set averaging prevents the larger pseudo-label set from washing out the anchors.

### 4.3 Reinforcing Pseudo-Label Quality with RPL-PO

We incorporate RPL-PO to mine an additional supervisory signal from the cross-round trajectory itself: the same LLM's two answers on the same node, before and after one round of teaching, form a free preference pair without any label or external judge.

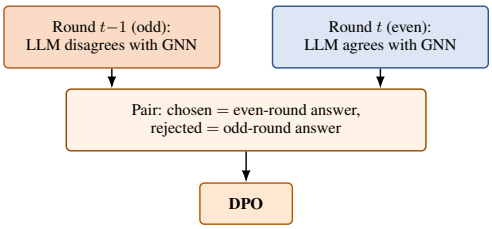


Figure 3: RPL-PO zoom-in: every two rounds, the LLM's even-round answer (peer-confirmed) is preferred.

**Intuition.** Round-by-round co-teaching gives us something single-round pipelines do not have: *the same LLM's two answers on the same node, before and after one round of teaching*. When the second answer is endorsed by the now-stronger GNN while the first was not, the LLM has visibly self-corrected on that node, and we want training to make this correction stick. We harvest this signal as *Round-based Pseudo-Label Preference Optimization* (RPL-PO), illustrated in Fig. 3: the later, peer-endorsed answer is preferred over the earlier, peer-rejected answer, and the LLM is updated by DPO [35] on these pairs. As argued in §1, this lets us mine supervision from the trajectory itself, without designating either model as the golden teacher.

Consecutive rounds $(2k-1, 2k)$ are seeded to draw the *same* batch, so for every node $v$ we have an odd-round LLM prediction $\hat{y}_v^{L,\text{odd}}$ and an even-round prediction $\hat{y}_v^{L,\text{even}}$ on the same input. We keep $v$ in the preference set when both conditions hold:

$$\underbrace{\hat{y}_v^{L,\text{odd}} \neq \hat{y}_v^{G,\text{odd}}}_{\text{(i) odd round disagrees}} \quad \text{and} \quad \underbrace{\hat{y}_v^{L,\text{even}} = \hat{y}_v^{G,\text{even}} \text{ and } \hat{y}_v^{L,\text{even}} \neq \hat{y}_v^{L,\text{odd}}}_{\text{(ii) LLM changes to agree}}. \tag{5}$$

The second condition ensures the pair represents genuine self-correction by the LLM rather than the GNN drifting onto an unchanged LLM. The LLM updated its answer, and the now-stronger GNN endorses the new one. We set

$$\text{chosen} \;=\; \hat{y}_v^{L,\text{even}}, \qquad \text{rejected} \;=\; \hat{y}_v^{L,\text{odd}}, \tag{6}$$

and apply standard DPO [35] with the SFT checkpoint as the reference. The contrast is across *time* (same model, two training stages) and grounded by cross-model consensus, providing supervision that requires no human annotation and no external reward model.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** For our main results, we evaluate on six text-attributed graphs spanning citation networks (Cora [2], Citeseer [3], PubMed [3], ogbn-arxiv [1]), Wikipedia hyperlinks (WikiCS [66]), and an Amazon product co-purchase subset (ogbn-products [1]). The five datasets other than WikiCS overlap with the benchmarks of Xu and Ding [34]. We additionally include WikiCS, on which their numbers are not available, and we re-run the published implementations of GNN-as-Judge and the recent LLM-as-Predictor baselines (LLM-GNN, LLaGA, GraphGPT) to fill the WikiCS column. Full per-dataset statistics (#nodes, #edges, #features, #classes) and the $k$-shot training / validation / test splits we use are listed in Appendix E.

**Baselines.** We compare against methods from three categories. (1) *Classical GNN models*: GCN [20], GAT [67], and GraphSAGE [68]. (2) *LLM-as-Predictors*: Zero-shot, Graph-CoT [69], and neighbor-augmented prompting. (3) *LLM-Graph methods*: GLEM [13], TAPE [4], LLM-GNN [8], LLaGA [11], GraphGPT [9], and GNN-as-Judge [34].

**Implementation details.** Full implementation details are deferred to Appendix F. For GLEM [13], TAPE [4], LLM-GNN [8], LLaGA [11], GraphGPT [9], and GNN-as-Judge [34], we verified that the results from [34] are reproducible under matched splits, and report their numbers on Cora, Citeseer, PubMed, ogbn-arxiv, and ogbn-products. WikiCS is not covered by [34] and is re-implemented by us. Other remaining baselines are implemented by us.

Table 1: Node classification accuracy (%) on six benchmarks under 3/5/10-shot settings. **Bold**: best. Our result is selected by the validation set at each shot budget. GAJ denotes GNN-as-Judge; its multi-round variant matches the LLM update-round budget.

| Method | Cora | Citeseer | PubMed | WikiCS | ogbn-arxiv | ogbn-products |
|---|---|---|---|---|---|---|
| *3-shot (3 labels per class)* | | | | | | |
| GCN | 70.72±3.06 | 55.14±3.82 | 67.72±5.19 | 60.33±4.59 | 39.83±1.48 | 60.14±0.41 |
| GAT | 70.74±2.12 | 60.46±4.46 | 66.50±5.09 | 59.65±5.21 | 32.43±1.56 | 57.14±1.23 |
| SAGE | 68.89±2.87 | 56.74±3.22 | 66.10±5.56 | 58.55±4.41 | 34.69±2.47 | 58.85±0.88 |
| Zero-Shot | 66.15±0.36 | 58.56±0.84 | 74.78±0.65 | 67.65±0.75 | 50.35±1.66 | 75.88±1.05 |
| Graph-CoT | 63.23±0.88 | 48.56±2.77 | 86.90±2.60 | 74.50±1.15 | 49.45±1.05 | 74.56±1.95 |
| w. Neighbor | 68.86±1.78 | 54.05±1.55 | 75.26±2.56 | 72.05±2.54 | 49.50±1.86 | 76.95±1.55 |
| GLEM | 67.81±0.92 | 53.09±1.39 | 63.85±1.02 | 71.56±1.56 | 36.37±1.96 | 52.46±1.93 |
| TAPE | 73.71±1.86 | 64.96±0.36 | 71.33±0.87 | 73.56±2.55 | 48.25±0.77 | 69.64±1.15 |
| LLM-GNN | 73.85±0.74 | 61.67±0.58 | 66.01±0.63 | 72.15±2.56 | 42.36±1.72 | 55.46±0.37 |
| LLaGA | 54.79±0.91 | 32.93±1.24 | 43.96±1.74 | 71.56±0.56 | 29.73±2.82 | 30.67±1.54 |
| GraphGPT | 57.77±1.62 | 52.34±1.06 | 57.51±1.89 | 72.26±1.68 | 31.26±2.73 | 40.83±2.97 |
| GNN-as-Judge | 77.89±1.28 | 73.59±0.64 | 87.12±0.89 | 67.50±1.02 | 62.21±1.45 | 81.02±1.23 |
| GAJ (multi-round) | 79.12±1.33 | 73.13±1.87 | 89.88±1.65 | 69.35±1.41 | 62.58±1.89 | 81.56±2.54 |
| Ours | **85.46**±0.98 | **76.94**±0.85 | **91.05**±1.11 | **74.66**±1.05 | **69.20**±1.54 | **82.01**±1.31 |
| *5-shot (5 labels per class)* | | | | | | |
| GCN | 75.99±1.95 | 62.30±3.50 | 74.05±1.78 | 66.14±2.76 | 44.96±1.33 | 68.33±0.22 |
| GAT | 74.70±2.35 | 65.56±2.12 | 72.32±1.52 | 65.32±1.58 | 39.27±1.19 | 66.80±0.45 |
| SAGE | 74.05±2.78 | 64.11±2.73 | 71.85±2.13 | 64.39±2.89 | 39.75±3.23 | 67.55±0.67 |
| Zero-Shot | 66.15±0.36 | 58.56±0.84 | 74.78±0.65 | 67.65±0.75 | 50.35±1.66 | 75.88±1.05 |
| Graph-CoT | 63.23±0.88 | 48.56±2.77 | 86.90±2.60 | 74.50±1.15 | 49.45±1.05 | 74.56±1.95 |
| w. Neighbor | 68.86±1.78 | 54.05±1.55 | 75.26±2.56 | 72.05±2.54 | 49.50±1.86 | 76.95±1.55 |
| GLEM | 74.69±0.46 | 61.71±0.58 | 73.29±1.44 | 75.56±1.84 | 39.19±1.57 | 56.87±1.70 |
| TAPE | 74.28±0.81 | 67.73±0.48 | 75.02±0.83 | 76.15±2.11 | 55.22±0.48 | 77.44±1.32 |
| LLM-GNN | 75.61±1.04 | 62.37±1.35 | 74.33±0.95 | 76.32±1.23 | 45.74±1.66 | 64.01±0.39 |
| LLaGA | 62.88±2.19 | 43.71±4.36 | 58.63±1.05 | 74.11±2.31 | 33.74±2.45 | 37.29±3.12 |
| GraphGPT | 60.17±1.44 | 51.83±2.24 | 57.39±3.67 | 75.16±1.22 | 36.25±1.87 | 44.78±2.34 |
| GNN-as-Judge | 79.54±0.39 | 74.39±1.63 | 87.49±1.23 | 78.10±1.23 | 66.76±0.83 | 81.93±2.21 |
| GAJ (multi-round) | 82.56±0.65 | 75.11±1.28 | 88.61±1.48 | 78.15±1.56 | 68.75±0.99 | 82.48±2.15 |
| Ours | **86.01**±1.59 | **77.12**±1.14 | **91.04**±1.69 | **78.28**±0.85 | **69.97**±1.84 | **83.15**±2.25 |
| *10-shot (10 labels per class)* | | | | | | |
| GCN | 79.40±0.98 | 67.55±2.57 | 76.24±2.88 | 71.66±1.27 | 50.29±1.50 | 70.24±0.24 |
| GAT | 77.41±1.51 | 67.84±1.60 | 74.87±3.04 | 70.43±0.74 | 45.49±0.87 | 67.97±0.65 |
| SAGE | 78.24±1.11 | 67.73±2.30 | 74.88±1.96 | 71.26±0.93 | 45.24±1.33 | 68.99±0.41 |
| Zero-Shot | 66.15±0.36 | 58.56±0.84 | 74.78±0.65 | 67.65±0.75 | 50.35±1.66 | 75.88±1.05 |
| Graph-CoT | 63.23±0.88 | 48.56±2.77 | 86.90±2.60 | 74.50±1.15 | 49.45±1.05 | 74.56±1.95 |
| w. Neighbor | 68.86±1.78 | 54.05±1.55 | 75.26±2.56 | 72.05±2.54 | 49.50±1.86 | 76.95±1.55 |
| GLEM | 78.11±0.73 | 66.83±0.61 | 74.17±2.39 | 76.51±1.55 | 47.73±1.09 | 60.22±1.89 |
| TAPE | 79.33±0.57 | 69.39±0.65 | 77.18±1.06 | 76.64±1.03 | 60.37±0.92 | 79.53±0.63 |
| LLM-GNN | 79.39±1.26 | 66.28±0.94 | 76.82±0.57 | 76.82±1.05 | 52.74±0.48 | 66.98±0.39 |
| LLaGA | 69.25±0.97 | 51.22±1.43 | 67.29±2.26 | 74.34±1.89 | 45.35±1.74 | 40.55±1.63 |
| GraphGPT | 61.58±0.77 | 55.40±3.16 | 71.33±2.81 | 75.80±1.63 | 48.67±1.89 | 51.46±1.05 |
| GNN-as-Judge | 80.71±0.83 | 74.62±1.35 | 90.17±1.69 | 78.30±1.34 | 67.88±1.03 | 82.48±1.56 |
| GAJ (multi-round) | 83.15±0.89 | 75.31±1.65 | 90.50±1.85 | 78.41±1.55 | 67.35±1.99 | 82.87±2.85 |
| Ours | **85.95**±1.62 | **78.02**±1.25 | **92.42**±0.68 | **78.56**±2.08 | **71.02**±1.57 | **83.41**±0.84 |

**More Settings and Datasets.** Appendices J–L present additional experiments with higher label budgets and compute-matched controls, as well as evaluations on STRING-db and amazon-ratings.

### 5.2 Few-Shot Semi-Supervised Node Classification

Table 1 presents the main results. We highlight three key observations.

**Obs 1. LLM-GNN Co-Teaching achieves the best accuracy on all six benchmarks across all three label budgets.** The average absolute gain over single-pass GNN-as-Judge (the strongest prior method on average) is $5.00\%$ at 3-shot, with the largest improvement of $7.57\%$ on Cora. We also include two multi-round control baselines with the same number of LLM update rounds as our method (Appendix K). The gains stay consistently positive as the label budget grows, indicating that the trajectory-mined supervision in RPL-PO is most useful when labels are scarce.

**Obs 2. The improvement scales with task difficulty.** On the large, fine-grained benchmarks (ogbn-arxiv with 40 classes) and on the topologically heterogeneous WikiCS, classical GNNs and recent LLM-as-Predictor baselines both deteriorate sharply — LLaGA and GraphGPT drop to roughly 30% accuracy on 3-shot ogbn-arxiv — revealing that pure structural inductive bias and pure semantic prompting both struggle when the label budget is small and the class space is large. LLM-GNN Co-Teaching narrows the gap to the supervised regime that prior methods could not close.

**Obs 3. Bidirectional co-teaching beats every form of unidirectional supervision.** LLM-as-Enhancer methods (TAPE, GLEM, LLM-GNN) and LLM-as-Predictor methods (LLaGA, GraphGPT) treat one model's outputs as fixed supervision; GNN-as-Judge keeps the supervision flow unidirectional but inverts who judges whom. LLM-GNN Co-Teaching is the only entry where neither model is fixed as teacher, and it dominates all of these on every (dataset, shot) cell. Removing the golden-teacher constraint thus pays off across the spectrum from frozen-feature transfer to single-round agreement filtering, which we further dissect through ablations and error-structure analysis below.

### 5.3 Ablation Study

We ablate the key components of LLM-GNN Co-Teaching under the 3-shot setting across all six datasets (Table 2). *Co-teaching structure*: removing bidirectional teaching (teach-once, GNN frozen after Round 0) lowers mean accuracy on all six datasets, while removing RPL-PO isolates the additional gain attributable to trajectory-based preference optimization on top of SFT. *Selection mechanism*: replacing the linearly annealed $R(t)$ with a fixed ratio (0.5 or 0.2) underperforms on most datasets, and agreement-based selection (keeping only nodes where GNN and LLM agree) lags small-loss ranking on most datasets, indicating that confidence-based ranking provides value beyond simple agreement filtering. *Training configuration*: removing neighbor information from the prompt drops accuracy on every dataset, showing that neighbor context complements the node's own text.

Table 2: Ablation study (3-shot). We report LLM test accuracy (%) selected by the validation set. Ablation rows are mean±std over 3 seeds. Each row removes or modifies one component from the full method.

| Variant | Cora | Citeseer | PubMed | WikiCS | ogbn-arxiv | ogbn-products |
|---|---|---|---|---|---|---|
| **Full Model** | **85.46**±0.98 | **76.94**±0.85 | **91.05**±1.11 | **74.66**±1.05 | **69.20**±1.54 | **82.01**±1.31 |
| *Co-teaching structure* | | | | | | |
| w/o bidirectional | 78.21±0.32 | 74.12±0.45 | 86.23±0.62 | 69.21±0.45 | 65.11±0.68 | 79.18±0.75 |
| w/o RPL-PO | 82.21±0.45 | 75.11±0.44 | 89.02±0.52 | 70.22±0.88 | 66.72±0.74 | 79.72±0.68 |
| *Selection mechanism* | | | | | | |
| Fixed $R$=0.5 | 83.12±0.71 | 76.12±0.72 | 87.31±0.59 | 74.30±0.45 | 67.01±0.62 | 81.75±0.52 |
| Fixed $R$=0.2 | 81.88±0.81 | 75.38±0.51 | 85.12±0.65 | 72.66±0.72 | 67.07±0.36 | 80.65±0.42 |
| Agreement selection | 82.24±0.64 | 72.01±0.58 | 90.84±0.67 | 72.23±0.65 | 68.14±0.61 | 81.98±0.57 |
| *Training configuration* | | | | | | |
| w/o neighbor info | 84.77±0.33 | 76.67±0.35 | 90.65±0.57 | 73.01±0.81 | 68.52±0.75 | 80.66±0.56 |

### 5.4 Cross-Dataset Zero-Shot Transfer

We evaluate the zero-shot generalization of LLM-GNN Co-Teaching by training on ogbn-arxiv and evaluating on Cora, Citeseer, and PubMed *without any fine-tuning on the target dataset*. Unlike GNNs, which require task-specific classification heads, LLMs trained via co-teaching can transfer across label sets because the learned capability is text-based classification, not tied to a fixed output space. Table 3 shows that LLM-GNN Co-Teaching achieves strong zero-shot transfer, outperforming prior LLM-graph methods. This suggests that iterative co-teaching improves the LLM's general graph reasoning ability, not just its performance on the training distribution.

Table 3: Zero-shot cross-dataset accuracy.

| Source / Target | Model | Acc. |
|---|---|---|
| arxiv / Cora | LLaGA | 16.24±0.95 |
| | GraphGPT | 6.29±0.73 |
| | GNN-as-Judge | 68.27±0.91 |
| | **Ours** | **68.85**±0.31 |
| arxiv / Citeseer | LLaGA | 14.72±1.12 |
| | GraphGPT | 5.37±0.84 |
| | GNN-as-Judge | 56.67±0.89 |
| | **Ours** | **58.95**±0.23 |
| arxiv / PubMed | LLaGA | 30.52±1.18 |
| | GraphGPT | 10.54±1.05 |
| | GNN-as-Judge | 83.41±0.76 |
| | **Ours** | **85.03**±0.23 |

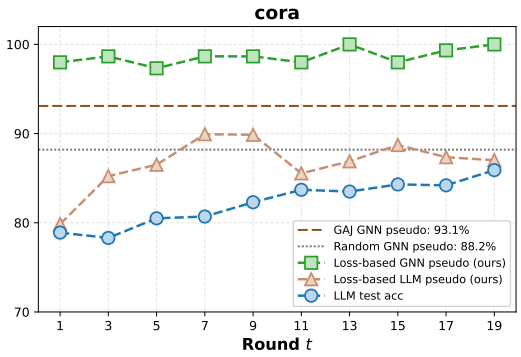


Figure 4: Per-round signals on cora.

### 5.5 Over Round Pseudo Label Quality

For each round we record the accuracy of the small-loss-selected GNN and LLM pseudo-label streams that each peer feeds the other, together with downstream LLM test accuracy. On cora, the GNN stream stays at 97–100%, above GAJ's GNN-pseudo quality (93.1%) and uniform Random selection (88.2%); the LLM stream rises from 80% at R1 to ∼87%, exhibiting an upward trajectory that fixed-rule baselines cannot. The clean pseudo-labels translate into accuracy gains: the LLM climbs from 78.9% (R1) to 85.9% (R19). The same pattern holds on arxiv, pubmed, and wikics (App. C).

### 5.6 Error Structure Analysis

To examine *where* co-teaching helps, we analyze how errors are structured along node degree on ogbn-arxiv (3-shot, 1,000 test nodes); degree is one representative axis, not a complete characterization. Figure 5(a) plots the smoothed per-degree error rate of the no-teaching baseline. Panels (b)–(d) plot the *error-fraction density*: a Gaussian KDE on the degrees of misclassified nodes, scaled by $n_{\text{err}}/N_{\text{total}}$ so the area under each curve equals the model's overall error rate. The three densities compare *No teaching*, *GNN-as-Judge* [34] (its full SFT+ORPO pipeline, using GAJ checkpoints), and LLM-GNN Co-Teaching; (b) and (d) share node IDs from the same arxiv run.

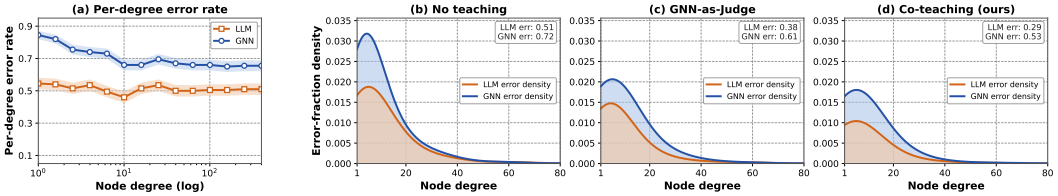


Figure 5: Error structure on ogbn-arxiv 3-shot. **(a)** Per-degree error rate at the no-teaching baseline: **LLM** stays nearly flat at ≈ 0.5, while **GNN** spikes to 0.85 at degree 1 and decays to 0.65 at high degree. **(b)–(d)** Error-fraction density across stages.

Panel (a) reveals two complementary failure modes: the LLM error rate varies little with degree, despite using neighbor text in its prompt, while the GNN fails on cold nodes ($P(\text{error} \mid \text{deg}{=}1) \approx 0.85$ vs. 0.65 at high degree). GAJ's single-pass pipeline (c) updates the LLM while keeping the GNN fixed, leaving the GNN's errors concentrated on low-degree nodes. Co-teaching (d) attains the smallest error mass under both curves and roughly halves the GNN low-degree error-density peak, suggesting that repeated rounds transfer the LLM's degree-robust signal to the GNN where neighbors are sparse, while the LLM benefits from a progressively cleaner GNN-selected pseudo-label set.

### 5.7 Time Analysis, Robustness Check, and Sensitivity Check

For space, the **time analysis** (App. A, G), **robustness check** (App. H), and **sensitivity check** (App. I) are deferred to the appendix.

## 6 Conclusion

We proposed LLM-GNN Co-Teaching, a co-teaching framework that abandons the golden-teacher assumption and enables mutual improvement between GNNs and LLMs for graph learning. By combining co-teaching with RPL-PO from the training trajectory, LLM-GNN Co-Teaching establishes a new state of the art on six graph benchmarks, improving over GNN-as-Judge by up to 7.57% under 3-shot supervision, with consistent gains under 5-shot supervision and in zero-shot cross-dataset transfer. More broadly, our work demonstrates that bidirectional co-teaching, in which LLM and GNN teach each other, is an effective approach to LLM graph learning. We believe this co-teaching paradigm may extend to other tasks, where GNN and LLM provide complementary views of the data. Limitations of the present study are discussed in Appendix A.

## Funding and Competing Interests

**Funding.** This research was a HKUST-WeBank Joint Lab Project and supported by the funding from WeBank (grant number WEB24EG01-M). We also express our gratitude to WeBank for providing access toresearch facilities and resources.

**Competing interests.** The authors declare no competing interests.

## References

[1] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[2] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

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

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

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

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

[7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] 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.

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

[10] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs. In *Findings of the association for computational linguistics: EACL 2024*, pages 1955–1973, 2024.

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

[12] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*, 2023.

[13] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*, 2022.

[14] Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. Instructgraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13492–13510, 2024.

[15] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 295–304, 2020.

[16] Kaize Ding, Jianling Wang, James Caverlee, and Huan Liu. Meta propagation networks for graph few-shot semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 6524–6531, 2022.

[17] Song Wang, Chen Chen, and Jundong Li. Graph few-shot learning with task-specific structures. *Advances in Neural Information Processing Systems*, 35:38925–38936, 2022.

[18] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. *Advances in neural information processing systems*, 33:5862–5874, 2020.

[19] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5892–5899, 2020.

[20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[21] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. Pmlr, 2019.

[22] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[23] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. pmlr, 2018.

[24] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[25] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

[26] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can llms effectively leverage graph structural information through prompts, and why? *arXiv preprint arXiv:2309.16595*, 2023.

[27] Xixi Wu, Yifei Shen, Fangzhou Ge, Caihua Shan, Yizhu Jiao, Xiangguo Sun, and Hong Cheng. When do llms help with node classification? a comprehensive analysis. *arXiv preprint arXiv:2502.00829*, 2025.

[28] Xinnan Dai, Haohao Qu, Yifei Shen, Bohang Zhang, Qihao Wen, Wenqi Fan, Dongsheng Li, Jiliang Tang, and Caihua Shan. How do large language models understand graph patterns? a benchmark for graph pattern comprehension. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.

[30] Yichuan Li, Kaize Ding, and Kyumin Lee. Grenade: Graph-centric language model for self-supervised representation learning on text-attributed graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2745–2757, 2023.

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

[32] 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, 2024.

[33] Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. Let's ask gnn: Empowering large language model for graph in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1396–1409, 2024.

[34] Ruiyao Xu and Kaize Ding. Gnn-as-judge: Unleashing the power of llms for graph learning with gnn feedback. *arXiv preprint arXiv:2604.08553*, 2026.

[35] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

[37] Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 13087–13095, 2025.

[38] 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*, pages 2526–2537, 2025.

[39] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[40] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International conference on machine learning*, pages 7164–7173. PMLR, 2019.

[41] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[42] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152, 2018.

[43] Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. Self-paced co-training. In *International Conference on Machine Learning*, pages 2275–2284. PMLR, 2017.

[44] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari. Learning with noisy labels. In *NeurIPS*, 2013.

[45] Xian-Jin Gui, Wei Wang, and Zhang-Hao Tian. Towards understanding deep learning from noisy labels with small-loss criterion. *arXiv preprint arXiv:2106.09291*, 2021.

[46] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International conference on machine learning*, pages 1062–1070. PMLR, 2019.

[47] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020.

[48] Junyu Luo, Xiao Luo, Kaize Ding, Jingyang Yuan, Zhiping Xiao, and Ming Zhang. Robustft: Robust supervised fine-tuning for large language models under noisy response. *arXiv preprint arXiv:2412.14922*, 2024.

[49] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

[50] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

[51] Subhabrata Mukherjee and Ahmed Awadallah. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems*, 33:21199–21212, 2020.

[52] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

[53] Hongwei Wang and Jure Leskovec. Unifying graph convolutional neural networks and label propagation. *arXiv preprint arXiv:2002.06755*, 2020.

[54] Hongrui Liu, Binbin Hu, Xiao Wang, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. Confidence may cheat: Self-training on graph neural networks under distribution shift. In *Proceedings of the ACM web conference 2022*, pages 1248–1258, 2022.

[55] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085*, 2017.

[56] 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.

[57] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[58] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

[59] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[60] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

[61] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning*, 2024.

[62] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[63] Siyan Zhao, John Dang, and Aditya Grover. Group preference optimization: Few-shot alignment of large language models. *arXiv preprint arXiv:2310.11523*, 2023.

[64] Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9954–9972, 2024.

[65] Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, 2024.

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

[67] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

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

[69] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[70] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[71] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[72] Taiyan Zhang, Renchi Yang, Yurui Lai, Mingyu Yan, Xiaochun Ye, and Dongrui Fan. Leveraging large language models for effective label-free node classification in text-attributed graphs. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pages 698–708, 2025.

[73] Xing Wei, Chunchun Chen, Rui Fan, Xiaofeng Cao, Sourav Medya, and Wei Ye. Preference-driven knowledge distillation for few-shot node classification. *Advances in Neural Information Processing Systems*, 38:103297–103327, 2026.

[74] Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 747–758, 2024.

# A Limitations

**Time complexity.** LLM-GNN Co-Teaching increases time complexity over a single-shot LLM-on-graph pipeline by a factor of $T$, the number of co-teaching rounds, because each round repeats vLLM inference on the unlabeled batch, an LLM SFT pass on the anchors plus the GNN-selected pseudo-labels, and a GNN training pass on the full graph. Every even round additionally runs a DPO pass on the temporal preference pairs from rounds $t-1$ and $t$.

**Coverage of data domains.** We evaluate on six text-attributed graphs from three domains: citation networks (Cora, Citeseer, PubMed, ogbn-arxiv), Wikipedia hyperlinks (WikiCS), and Amazon co-purchase (ogbn-products). Additional benchmarks include a protein-association graph and amazon-ratings. The method has not been validated on graph-only molecular or financial datasets. The assumption that GNN structural inductive bias and LLM semantic reasoning offer complementary signal is most clearly supported in our setting and may need re-examination in domains where one of the two views is degraded. Extending the evaluation to such domains is left for future work.

# B Broader Impacts

**Positive impacts.** LLM-GNN Co-Teaching targets the few-shot regime, where labeled data is scarce. This setting is common in domains where annotation is expensive or only domain experts can produce labels, including rare scientific subfields, niche legal or policy corpora, low-resource languages, and biomedical sub-disciplines. By turning an LLM and a GNN into mutual teachers, our method extracts more value from the few existing labels and from the unlabeled remainder of the graph. The preference signal comes entirely from the model trajectory, with no human annotators, no reward model, and no external judge, which lowers the practical barrier for groups with limited labeling budgets. The framework is also model-agnostic at the LLM side, so the same recipe applies to smaller open-weight LLMs and is therefore accessible to research groups without frontier-scale resources.

**Negative impacts.** The recipe inherits the well-known concerns of large language models. LLM-generated pseudo-labels can carry the social, demographic, or political biases of the pretraining corpus, and the iterative co-teaching loop can amplify any bias that the GNN does not correct. We mitigate this with the small-loss criterion, which keeps only the LLM's most confident predictions, but confident LLM mistakes are still possible and can persist across rounds. The compute and energy footprint per run is higher than a single-shot pipeline, as quantified in Appendix A, which has the usual implications for carbon cost. Finally, automated node classification at scale on social or behavioral graphs raises privacy concerns when the underlying data is not consented or the application is surveillance-oriented. Practitioners deploying this framework outside academic benchmarks should audit the pretrained LLM for domain bias, evaluate calibration on a held-out set, and follow the dataset licence and consent conditions.

# C Per-Round Pseudo-Label Quality

Figure 6 shows the same per-round signals as Figure 4 (§5.5) but for all four datasets (arxiv, cora, pubmed, wikics). The trends shown for Cora in Figure 4 hold across the additional datasets: top-10% pseudo-label streams climb across rounds and stay in the 80–99% band, while LLM test accuracy improves across rounds.

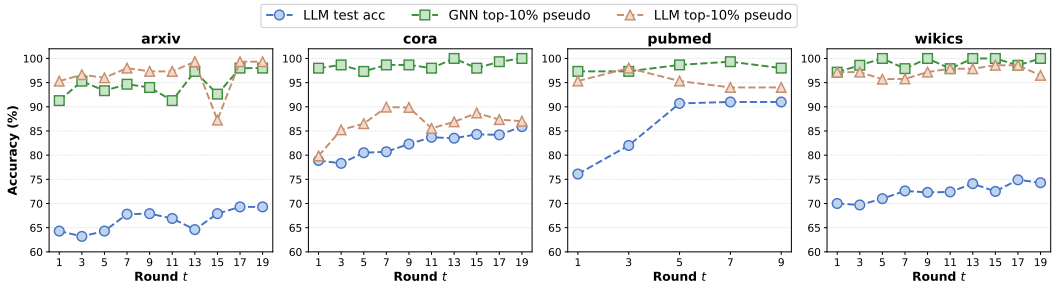


Figure 6: Per-round LLM test accuracy and top-10% GNN/LLM pseudo-label quality across all four 3-shot benchmarks (arxiv, cora, pubmed, wikics). **Blue** (circles): LLM test acc. **Green** (squares): GNN top-10% pseudo-label GT-acc. **Salmon** (triangles): LLM top-10% pseudo-label GT-acc. Pseudo-label quality at round $t+1$ is paired with the test accuracy at round $t$ since round-$t+1$'s pseudo-labels are produced by the round-$t$ models.

## D LLM SFT Prompt Specifications

The following boxes give the verbatim per-dataset prompt template appended to each node's text during LLM SFT. The user message passed to Llama-3-8B-Instruct is the concatenation of (i) the node's raw text (and, when using neighbor information, the 1-hop and 2-hop neighbor titles), (ii) the dataset-specific prompt below.

**cora (7 classes)**

```
Question: Which of the following sub-categories of AI does this paper
belong to? Here are the 7 categories: Rule_Learning, Neural_Networks,
Case_Based, Genetic_Algorithms, Theory, Reinforcement_Learning,
Probabilistic_Methods. Reply only one category that you think this paper
might belong to. Only reply the category phrase without any other
explanation words.

Answer:
```

**citeseer (6 classes)**

```
Question: Which of the following theme does this paper belong to?
Here are the 6 categories: Agents, ML (Machine Learning), IR
(Information Retrieval), DB (Databases), HCI (Human-Computer
Interaction), AI (Artificial Intelligence). Reply only one category
that you think this paper might belong to. Only reply the category
full name I give you without any other words.

Answer:
```

**pubmed (3 classes)**

```
Question: Which of the following topic does this scientific publication
talk about? Here are the 3 categories: Experimental, Diabetes Mellitus
Type 1, Diabetes Mellitus Type 2. Reply only one category that you
think this paper might belong to. Only reply the category name without
any other words.

Answer:
```

**wikics (10 classes)**

```
Question: Which of the following branch of Computer science does this
Wikipedia-based dataset belong to? Here are the 10 categories:
Computational Linguistics, Databases, Operating Systems, Computer
Architecture, Computer Security, Internet Protocols, Computer File
Systems, Distributed Computing Architecture, Web Technology,
Programming Language Topics. Reply only one category that you think
this paper might belong to. Only reply the category full name without
any other words.

Answer:
```

**ogbn-arxiv (40 classes)**

```
Question: Which of the following arXiv CS sub-categories does this dataset
belong to? Here are the 40 categories: 'arxiv cs na', 'arxiv cs mm',
'arxiv cs lo', 'arxiv cs cy', 'arxiv cs cr', 'arxiv cs dc', 'arxiv cs hc',
'arxiv cs ce', 'arxiv cs ni', 'arxiv cs cc', 'arxiv cs ai', 'arxiv cs ma',
'arxiv cs gl', 'arxiv cs ne', 'arxiv cs sc', 'arxiv cs ar', 'arxiv cs cv',
'arxiv cs gr', 'arxiv cs et', 'arxiv cs sy', 'arxiv cs cg', 'arxiv cs oh',
'arxiv cs pl', 'arxiv cs se', 'arxiv cs lg', 'arxiv cs sd', 'arxiv cs si',
'arxiv cs ro', 'arxiv cs it', 'arxiv cs pf', 'arxiv cs cl', 'arxiv cs ir',
'arxiv cs ms', 'arxiv cs fl', 'arxiv cs ds', 'arxiv cs os', 'arxiv cs gt',
'arxiv cs db', 'arxiv cs dl', 'arxiv cs dm'. Use the words in this part to
answer me, not the explanation part below.

Here are the explanation of each category: 'arxiv cs ai (Artificial
Intelligence)', 'arxiv cs ar (Hardware Architecture)', 'arxiv cs cc
(Computational Complexity)', 'arxiv cs ce (Computational Engineering, Finance,
and Science)', 'arxiv cs cg (Computational Geometry)', 'arxiv cs cl
(Computation and Language)', 'arxiv cs cr (Cryptography and Security)',
'arxiv cs cv (Computer Vision and Pattern Recognition)', 'arxiv cs cy
(Computers and Society)', 'arxiv cs db (Databases)', 'arxiv cs dc
(Distributed, Parallel, and Cluster Computing)', 'arxiv cs dl (Digital
Libraries)', 'arxiv cs dm (Discrete Mathematics)', 'arxiv cs ds (Data
Structures and Algorithms)', 'arxiv cs et (Emerging Technologies)',
'arxiv cs fl (Formal Languages and Automata Theory)', 'arxiv cs gl (General
Literature)', 'arxiv cs gr (Graphics)', 'arxiv cs gt (Computer Science and
Game Theory)', 'arxiv cs hc (Human-Computer Interaction)', 'arxiv cs ir
(Information Retrieval)', 'arxiv cs it (Information Theory)', 'arxiv cs lg
(Machine Learning)', 'arxiv cs lo (Logic in Computer Science)', 'arxiv cs ma
(Multiagent Systems)', 'arxiv cs mm (Multimedia)', 'arxiv cs ms (Mathematical
Software)', 'arxiv cs na (Numerical Analysis)', 'arxiv cs ne (Neural and
Evolutionary Computing)', 'arxiv cs ni (Networking and Internet
Architecture)', 'arxiv cs oh (Other Computer Science)', 'arxiv cs os
(Operating Systems)', 'arxiv cs pf (Performance)', 'arxiv cs pl (Programming
Languages)', 'arxiv cs ro (Robotics)', 'arxiv cs sc (Symbolic Computation)',
'arxiv cs sd (Sound)', 'arxiv cs se (Software Engineering)', 'arxiv cs si
(Social and Information Networks)', 'arxiv cs sy (Systems and Control)'.

Reply only one category that you think this paper might belong to. Only reply
the category name (not the explanation) I given without any other words.

Answer:
```

**ogbn-products (47 classes)**

```
Which of the following categories does this product belong to? There
are a total of 47 categories, including Home & Kitchen, Health &
Personal Care, Beauty, Sports & Outdoors, Books, Patio, Lawn & Garden,
Toys & Games, CDs & Vinyl, Cell Phones & Accessories, Grocery &
Gourmet Food, Arts, Crafts & Sewing, Clothing, Shoes & Jewelry,
Electronics, Movies & TV, Software, Video Games, Automotive, Pet
Supplies, Office Products, Industrial & Scientific, Musical
Instruments, Tools & Home Improvement, Magazine Subscriptions, Baby
Products, NAN, Appliances, Kitchen & Dining, Collectibles & Fine Art,
All Beauty, Luxury Beauty, Amazon Fashion, Computers, All Electronics,
Purchase Circles, MP3 Players & Accessories, Gift Cards, Office &
School Supplies, Home Improvement, Camera & Photo, GPS & Navigation,
Digital Music, Car Electronics, Baby, Kindle Store, Kindle Apps,
Furniture. Reply only one category that you think this product might
belong to. Only reply the category name I give of the category without
any other words and numbers.

Answer:
```

# E Datasets and Splits

We evaluate on six text-attributed graphs. The first five (Cora, Citeseer, PubMed, ogbn-arxiv, ogbn-products) match the benchmark of Xu and Ding [34]. We additionally include `WikiCS`, a Wikipedia-hyperlink graph where each node is an English Wikipedia article on computer science. Per-dataset statistics are listed in Table 4. The training / validation / test splits we use under each $k$-shot setting are given in Table 5. The test pool is the entire remainder of the graph.

Table 4: Dataset statistics, all extracted from our local files. For ogbn-products(subset) we use the curated subset of He et al. [4] (54,025 product nodes, the same subset reused by Xu and Ding [34]).

| Dataset | #Nodes | #Edges | #Features | #Classes | Domain |
|---|---|---|---|---|---|
| Cora | 2,708 | 10,858 | 1,433 | 7 | paper citation |
| Citeseer | 3,186 | 4,277 | 3,703 | 6 | paper citation |
| PubMed | 19,717 | 88,670 | 384 | 3 | biomedical citation |
| WikiCS | 11,701 | 431,726 | 300 | 10 | Wikipedia hyperlink |
| ogbn-arxiv | 169,343 | 1,166,243 | 128 | 40 | arXiv CS citation |
| ogbn-products(subset) | 54,025 | 72,319 | 100 | 47 | Amazon co-purchase |

**$k$-shot splits.** For Cora, Citeseer, PubMed, and WikiCS we sample $k$ labeled nodes per class as the training (anchor) set, fix $500$ random nodes as the validation set, and treat the remaining nodes as the test pool. Following the setup of GNN-as-Judge [34], we sample up to $k$ labeled nodes per class from the original training pool of ogbn-products(subset). Nine of the subset's $42$ observed classes have no nodes in this pool, so labeled nodes can be sampled from only $33$ classes. Classes with fewer than $k$ training candidates contribute all available nodes. This yields $92$, $147$, and $270$ training nodes at 3-, 5-, and 10-shot, respectively, with $500$ validation nodes and the remainder used as the test pool. For ogbn-arxiv we adopt the official OGB validation and test splits and re-sample $k$-shot training nodes per class from the OGB training pool, treating all other nodes (the unused OGB-train remainder + the OGB-test set) as the test pool.

**Per-dataset notes.**

- **Cora, Citeseer, PubMed.** Standard citation networks [2, 3] where each node is a paper and edges encode citations. Node text is paper title + abstract.
- **WikiCS.** A Wikipedia hyperlink graph filtered to computer-science articles, with $10$ subdomain classes (e.g. Computational Linguistics, Operating Systems, Web Technology). Each node's raw text is the lead paragraph of the article. WikiCS is *not* included in the GAJ benchmark. We re-run their published implementation to fill the WikiCS column of Table 1.
- **ogbn-arxiv.** An arXiv CS-paper citation network [1] with $40$ sub-categories. We use the OGB official splits ($90{,}941$ train / $29{,}799$ val / $48{,}603$ test). Our $k$-shot training set is sub-sampled from the OGB-train pool, and the remainder of the graph (unused OGB-train $\cup$ OGB-test) becomes the test pool.

Table 5: Train / validation / test sizes per dataset and $k$-shot setting. **Train** denotes the small labeled anchor set. **Val** $= 500$ for the small-graph datasets and the OGB official validation set (29,799) for ogbn-arxiv. **Test** is the remainder of the graph.

| | 3-shot | | | 5-shot | | | 10-shot | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | Train | Val | Test | Train | Val | Test | Train | Val | Test |
| Cora | 21 | 500 | 2,187 | 35 | 500 | 2,173 | 70 | 500 | 2,138 |
| Citeseer | 18 | 500 | 2,668 | 30 | 500 | 2,656 | 60 | 500 | 2,626 |
| PubMed | 9 | 500 | 19,208 | 15 | 500 | 19,202 | 30 | 500 | 19,187 |
| WikiCS | 30 | 500 | 11,171 | 50 | 500 | 11,151 | 100 | 500 | 11,101 |
| ogbn-arxiv | 120 | 29,799 | 139,424 | 200 | 29,799 | 139,344 | 400 | 29,799 | 139,144 |
| ogbn-products(subset) | 92 | 500 | 53,433 | 147 | 500 | 53,378 | 270 | 500 | 53,255 |

- **ogbn-products(subset).** A subset of the Amazon co-purchase graph from He et al. [4], filtered to 54,025 products across 42 categories, with text being the product title + description. We retain the original 47-class label space in LLM's prompt. We adopt the same subset and split convention as Xu and Ding [34].

# F Implementation Details and Hyperparameters

This appendix documents how every method in Table 1 is implemented and trained. Section F.1 covers the baseline methods (the same set used by Xu and Ding [34], with their numbers reproduced in our main table). Section F.2 describes the classical GNN backbones (GCN, GAT, SAGE) reported in our table. Section F.3 lists every hyperparameter used by LLM-GNN Co-Teaching.

## F.1 Baselines

**Classical GNN backbones (GCN, GAT, GraphSAGE).** All classical-GNN baselines are 2-layer message-passing networks with 64-dimensional hidden representations, trained with Adam (learning rate $10^{-2}$, weight decay $5 \times 10^{-4}$) for up to 200 epochs with early stopping at patience 100. Dropout is selected from $\{0.3, 0.5, 0.7\}$ on the validation set and batch normalization is inserted between the two layers. These choices match the setting in Xu and Ding [34] so the numbers transfer directly.

**LLM-as-Predictors.** Three prompting-only baselines all use Llama-3-8B-Instruct without any fine-tuning. *Zero-shot* prompts feed the node text and the candidate class names. *Graph Chain-of-Thought* [69] adds a step-by-step reasoning instruction. *Neighbor-Augmented Prompting* [8] additionally appends the texts of the node's local neighbors.

**LLM-Graph methods.**

- *GLEM* [13] alternates EM steps between an encoder LM and a GNN. Following the original setting, EM iterations $= 1$ and pseudo-label ratio $= 0.5$. The GNN is the same 2-layer 64-d architecture as the classical baselines. The LM is RoBERTa [70] with LoRA [71], batch size 32, pre-trained on each dataset before joint training.
- *TAPE* [4] uses Llama-3-8B-Instruct + LoRA (default settings) to generate per-node explanations that are appended to textual features. The GNN configuration is identical to the classical baselines.
- *LLM-GNN* [8] is adapted from its zero-shot original to our few-shot setting: the LLM (Llama-3-8B-Instruct) acts as an annotator on labeled data, the DA-AGE method from the original paper selects pseudo-labels, and the GNN is jointly trained on labeled and pseudo-labeled data.
- *LLaGA* [11] uses HO templates with hop count 4, RoBERTa [70] as the text encoder, a linear projection $\phi_\theta$ implemented as a 2-layer MLP with hidden dimension 2,048, batch size 64, learning rate $10^{-4}$, 10 epochs.
- *GraphGPT* [9] uses two instruction-tuning stages on dataset-specific graph-matching tasks. Self-supervised stage: lr $10^{-4}$, batch 16, 2 epochs. Task-specific stage: lr $10^{-4}$, batch 32, 10 epochs.
- *GNN-as-Judge* [34] is the closest competitor. The GNN is a 2-layer 64-d GCN with the same training setup as the classical baselines. The LLM is Llama-3-8B-Instruct + LoRA ($r$=8, $\alpha$=16, dropout 0.1, batch size 8). Instruction tuning runs for 10 epochs at lr $5 \times 10^{-6}$. The subsequent weakly-supervised fine-tuning runs for 8 epochs at lr $10^{-5}$ with the IT/PT mixing weight $\lambda = 0.1$. Top-$K$=1,500 influential nodes are selected for pseudo-labeling and the ORPO preference threshold $\tau$ is fixed at 0.7.

### F.2 Our GCN, GAT, and SAGE Backbones

The Classical-GNN rows of Table 1 use GCN, GAT, and GraphSAGE with identical optimization (Adam, lr $10^{-2}$, weight decay $5 \times 10^{-4}$), depth (2 layers), and stopping (early stop at patience 100, max 500 epochs) as the classical baselines above. Architecture-specific choices: GAT uses 4 attention heads in layer 1 and 1 head in layer 2, GraphSAGE uses mean aggregation, and dropout is fixed at 0.5 for all three. We re-train these from scratch under each $n$-shot setting and report mean and standard deviation over 5 random seeds.

### F.3 LLM-GNN Co-Teaching Hyperparameters

Table 6 lists every hyperparameter used to produce the LLM-GNN Co-Teaching results in Table 1. All six 3-shot primary runs (cora, citeseer, pubmed, wikics, ogbn-arxiv, ogbn-products) share these values. The only per-dataset variation is the dataset-specific prompt template (Appendix D). The identical setting was applied for the 5- and 10-shot rows.

Table 6: Full hyperparameter configuration for LLM-GNN Co-Teaching on the 3-shot benchmarks.

| **Stage** | **Hyperparameter** | **Value** |
|---|---|---|
| GNN initialization (Stage 2) | Architecture | GCN |
| | Hidden dimension | 64 |
| | Number of layers | 2 |
| | Dropout | 0.5 |
| | Learning rate | $1\times10^{-2}$ |
| | Epochs (early stop patience) | 500 (100) |
| LLM warm-up SFT (Stage 3) | Base model | Llama-3-8B-Instruct |
| | PEFT method | LoRA, $r$=8, $\alpha$=16 |
| | Learning rate | $5\times10^{-6}$ |
| | Epochs | 10 |
| | Per-device batch size | 4 (grad. accum. $= 1$) |
| | Anchor repeat $K$ | 3 |
| Per-round LLM SFT | Learning rate | $2\times10^{-5}$ |
| | Epochs | 2; 5; 10 |
| | Per-device batch size | 4 |
| | Anchor repeat $K$ | 3 |
| Per-round GNN training | Learning rate | $1\times10^{-3}$ |
| | Epochs | 200 |
| | Pseudo-label weight $\alpha$ | $0.3 \rightarrow 0.7$ (linear) |
| Co-teaching schedule | Number of rounds $T$ | 20 |
| | Unlabeled batch size $B$ | 1500 |
| | $R(t)$ min | linear, $R_{\min}$=0.05; 0.1; 0.2; 0.3; 0.4 |
| | $R(t)$ max | linear, $R_{\max}$=0.5; 0.6; 0.7; 0.8; 0.9; 1 |
| | Neighbor info in LLM prompt | 1-hop ($\leq$ 5 titles) + 2-hop ($\leq$ 5) |
| RPL-PO (even rounds) | Learning rate | $5\times10^{-6}$ |
| | Epochs | 1 |
| | $\beta$ | 0.1 |
| | Loss type | sigmoid (vanilla DPO) |
| Hardware / numerics | GPU | 1$\times$ NVIDIA RTX 5880 Ada (46 GB) |
| | Precision | bf16 |
| | Random seed | 42 |
| | Wall-clock per round | $\sim$25 min (arxiv), $\sim$5–10 min (cora) |

## G Time Analysis on ogbn-arxiv (3-shot)

To complement the time-complexity discussion in Appendix A, Figure 7 plots end-to-end wall-clock on a single NVIDIA A100-40GB against the corresponding 3-shot test accuracy on ogbn-arxiv. All methods are run under their declared training schedules; LLM-GNN Co-Teaching runs the default $T = 10$ co-teaching rounds plus the round-0 initialisation.

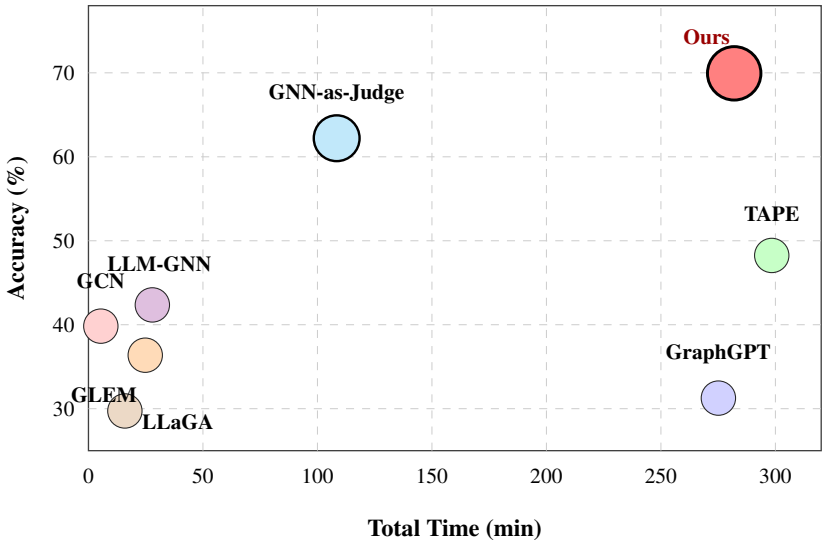


Figure 7: Training time versus ogbn-arxiv 3-shot accuracy on a single A100-40GB. **Ours** (red) sits in the same cost band as TAPE and GraphGPT but achieves the highest accuracy by a clear margin. **GNN-as-Judge** (cyan, thicker border) is the strongest baseline.

**Discussion.** Three patterns stand out. *Raw cost does not predict accuracy.* TAPE and GraphGPT each consume close to five hours and still trail GAJ on ogbn-arxiv. *LLM-GNN Co-Teaching achieves higher accuracy at a similar runtime to TAPE and GraphGPT.* Its wall-clock is within six percent of TAPE and GraphGPT, yet it achieves absolute accuracy gains of $21.69\%$ over TAPE, $38.68\%$ over GraphGPT, and $7.73\%$ over GAJ at roughly $2.6\times$ the GAJ runtime. *Methods with shorter runtimes achieve less than* $50\%$ *accuracy.* GCN, GLEM, LLaGA, and LLM-GNN finish in under half an hour and reach at most $42.36\%$ on this 40-class task, the regime where pure GNNs or single-shot LLM-on-graph methods cannot extract more from three labels per class. Iterative co-teaching trades extra compute for a markedly better point on the cost-accuracy curve.

## H Robustness to Backbone Choice

We swap the GNN and the LLM peers in turn while keeping every other hyperparameter fixed at the default in Appendix F and re-run the 3-shot pipeline across all six benchmarks. Three variants are compared with the Full Model (Llama-3-8B-Instruct + 2-layer GCN) in Table 7. **Vicuna-7B** substitutes a weaker LLM with no graph-aware pretraining. **GAT** and **GraphSAGE** substitute the GNN backbone, with all other hyperparameters left at their classical-GNN defaults from Appendix F.2.

Table 7: Robustness to backbone choice (3-shot, mean $\pm$ std over 3 seeds). To compare peak performance across backbones, we report the highest LLM test accuracy across rounds for each run. Full Model uses Llama-3-8B-Instruct and a 2-layer GCN. Each row swaps a single component. **Bold**: best.

| Variant | Cora | Citeseer | PubMed | WikiCS | ogbn-arxiv | ogbn-products |
|---|---|---|---|---|---|---|
| **Full Model** | $85.75 \pm 0.88$ | $77.12 \pm 0.97$ | $91.32 \pm 1.56$ | $74.80 \pm 0.95$ | $\mathbf{69.94 \pm 0.84}$ | $\mathbf{82.82 \pm 1.02}$ |
| *LLM swap* | | | | | | |
| Vicuna-7B | $78.56 \pm 1.56$ | $70.56 \pm 1.56$ | $85.56 \pm 1.33$ | $70.98 \pm 0.88$ | $65.56 \pm 1.23$ | $78.56 \pm 1.56$ |
| *GNN swap* | | | | | | |
| GAT | $\mathbf{86.12 \pm 0.98}$ | $77.05 \pm 1.23$ | $\mathbf{91.35 \pm 1.24}$ | $\mathbf{75.23 \pm 1.32}$ | $68.56 \pm 2.15$ | $78.88 \pm 1.32$ |
| GraphSAGE | $84.56 \pm 1.25$ | $\mathbf{77.32 \pm 1.12}$ | $91.12 \pm 1.05$ | $75.12 \pm 0.86$ | $68.45 \pm 1.55$ | $81.23 \pm 1.75$ |

**Findings.** *Robust to GNN specification.* Swapping the GCN backbone for GAT or GraphSAGE changes accuracy by less than $1.6\%$ in 11 of the 12 comparisons; the exception is GAT on ogbn-products, which reduces accuracy by $3.94\%$. The framework therefore does not rely on a specific message-passing architecture. *Suffers under a low-capability LLM.* Swapping Llama-3-8B-Instruct for Vicuna-7B drops accuracy by $3.8\%$ to $7.2\%$ across all six datasets. The LLM peer carries the bulk of the semantic signal, so a weaker base model bottlenecks

the co-teaching loop even when the GNN side is held fixed. LLM-GNN Co-Teaching is robust to the tested GNN backbones, while its performance depends on the LLM backbone.

# I Hyperparameter Sensitivity

We sweep two hyperparameters of LLM-GNN Co-Teaching and report 3-shot LLM test accuracy across all six benchmarks (Figure 8). The first sweep varies the unlabeled sample size $|\mathcal{B}_t|$ used for cross-inference at each round (default 1500). The second sweep varies the number of DPO epochs in RPL-PO (default 1). Each value is mean $\pm$ std over 3 seeds.

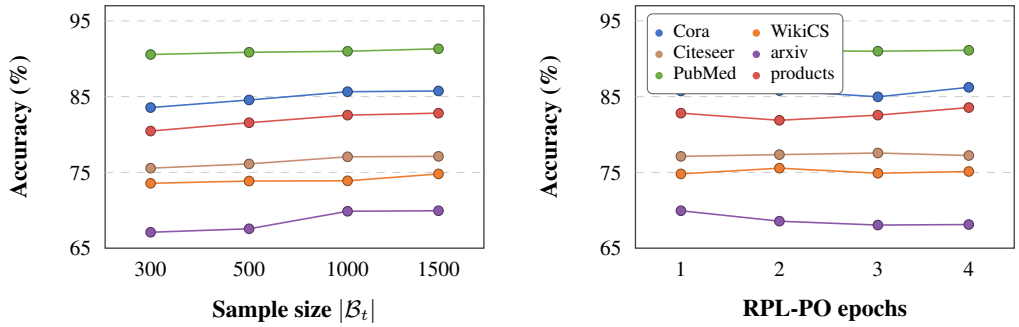


Figure 8: Sensitivity analysis of two LLM-GNN Co-Teaching hyperparameters across all six 3-shot benchmarks. **Left:** accuracy versus the unlabeled sample size $|\mathcal{B}_t|$ used for cross-inference per round; $|\mathcal{B}_t| = 1500$ is sufficient for every dataset and adding more nodes barely moves the accuracy. **Right:** accuracy versus the number of DPO epochs in RPL-PO; the curves are flat within one standard deviation, showing that the framework is not sensitive to this knob and a single epoch suffices.

## J Higher Label Budgets

We extend the ogbn-arxiv evaluation to 15, 20, 30, and 50 labeled nodes per class, including GLEM [13], Locle [72], PKD [73], GAugLLM [74], and GNN-as-Judge [34] among the baselines. Table 8 shows that LLM-GNN Co-Teaching improves from 71.02% at 10 shots to 74.30% at 50 shots and remains ahead of the strongest baseline at every budget. Zero-shot and Graph-CoT [69] do not use labeled examples, so their results are constant across budgets.

Table 8: ogbn-arxiv accuracy (mean±std, %) across label budgets.

| **Method** | **10-shot** | **15-shot** | **20-shot** | **30-shot** | **50-shot** |
|---|---|---|---|---|---|
| Zero-shot | 50.35±1.66 | 50.35±1.66 | 50.35±1.66 | 50.35±1.66 | 50.35±1.66 |
| Graph-CoT | 49.45±1.05 | 49.45±1.05 | 49.45±1.05 | 49.45±1.05 | 49.45±1.05 |
| GCN | 50.29±1.50 | 51.27±2.46 | 52.55±2.14 | 54.60±1.68 | 56.63±1.88 |
| GLEM, Llama-3-8B, multi-round | 56.98±2.95 | 58.87±2.56 | 62.36±2.14 | 64.84±2.68 | 67.47±2.51 |
| Locle | 56.75±1.32 | 59.54±1.56 | 59.88±1.68 | 60.50±1.86 | 62.45±2.65 |
| PKD | 43.67±1.39 | 44.20±2.65 | 46.30±2.98 | 44.08±1.89 | 45.55±2.34 |
| GAugLLM | 44.80±1.98 | 48.50±2.65 | 49.18±1.98 | 51.12±2.35 | 53.16±1.65 |
| LLaGA | 45.35±1.74 | 52.60±1.44 | 50.70±1.33 | 57.40±1.56 | 58.00±1.78 |
| GNN-as-Judge, single pass | 67.88±1.03 | 68.16±2.36 | 68.75±2.55 | 69.51±1.76 | 70.05±2.16 |
| GNN-as-Judge, multi-round | 67.35±1.99 | 69.22±1.28 | 68.92±2.38 | 70.71±2.16 | 71.56±1.86 |
| **Ours** | **71.02±1.57** | **73.00±1.86** | **73.90±1.68** | **74.10±2.65** | **74.30±2.86** |

## K Compute-Matched Comparisons

We compare against two controls with the same number of LLM update rounds as LLM-GNN Co-Teaching. Multi-round GNN-as-Judge [34] retains one-way supervision while increasing the LLM training budget. A Llama-3-8B-backed multi-round GLEM [13] updates both peers, testing bidirectional training without our small-loss exchange and RPL-PO. Table 9 reports the 3-shot results. LLM-GNN Co-Teaching remains ahead of both controls on all six datasets.

Table 9: Compute-matched 3-shot comparison (mean±std, %). Multi-round controls match our LLM update rounds; our checkpoint is selected by the validation set.

| **Method** | **Cora** | **Citeseer** | **PubMed** | **WikiCS** | **ogbn-arxiv** | **ogbn-products** |
|---|---|---|---|---|---|---|
| GNN-as-Judge, single pass | 77.89±1.28 | 73.59±0.64 | 87.12±0.89 | 67.50±1.02 | 62.21±1.45 | 81.02±1.23 |
| GNN-as-Judge, multi-round, compute-matched | 79.12±1.33 | 73.13±1.87 | 89.88±1.65 | 69.35±1.41 | 62.58±1.89 | 81.56±2.54 |
| GLEM, Llama-3-8B, multi-round | 68.51±1.56 | 54.86±1.78 | 65.88±1.41 | 73.14±1.06 | 38.52±2.74 | 54.88±2.61 |
| **Ours** | **85.46±0.98** | **76.94±0.85** | **91.05±1.11** | **74.66±1.05** | **69.20±1.54** | **82.01±1.31** |
| Improvement over multi-round GNN-as-Judge | 6.34 | 3.81 | 1.17 | 5.31 | 6.62 | 0.45 |

# L Additional Text-Attributed Graphs

We conduct evaluations on the STRING protein-association and Amazon Ratings datasets. Baselines include GAugLLM [74], PKD [73], GLEM [13], and GNN-as-Judge [34]. Table 10 reports 3-, 5-, and 10-shot accuracy. LLM-GNN Co-Teaching achieves absolute accuracy gains of 1.72–6.93% on STRING-db and 2.92–3.14% on amazon-ratings over the strongest listed baseline.

Table 10: Accuracy (%) on STRING-db and amazon-ratings. Our checkpoints are selected by the validation set.

| | **STRING-db** | | | **amazon-ratings** | | |
|---|---|---|---|---|---|---|
| **Method** | **3-shot** | **5-shot** | **10-shot** | **3-shot** | **5-shot** | **10-shot** |
| Graph-CoT | 39.92 | 39.98 | 39.93 | 27.63 | 27.63 | 27.66 |
| GCN | 37.37 | 44.42 | 51.95 | 30.05 | 31.49 | 31.26 |
| GAugLLM | 34.14 | 38.90 | 45.14 | 27.99 | 28.78 | 28.92 |
| PKD | 42.12 | 43.39 | 43.36 | 18.06 | 17.74 | 15.14 |
| GLEM, Llama-3-8B, multi-round | 49.71 | 51.72 | 53.62 | 36.49 | 37.26 | 38.66 |
| GNN-as-Judge, single pass | 42.33 | 47.36 | 48.47 | 19.35 | 31.62 | 28.85 |
| GNN-as-Judge, multi-round | 47.23 | 48.26 | 50.44 | 26.28 | 35.03 | 35.37 |
| **Ours** | **51.43** | **56.51** | **60.55** | **39.52** | **40.18** | **41.80** |

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and §1 state our two contributions (bidirectional co-teaching that drops the golden-teacher assumption, and RPL-PO). Each claimed gain is reported in Table 1 and verified with the per-round trajectories in §5.5 and the error analysis in §5.6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix A discusses the per-round LLM compute cost, the $\mathcal{O}(T)$ scaling of total cost, and the restricted set of data domains we evaluated (descriptive-text TAGs only).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper proposes an empirical method and reports empirical results. It does not state or prove formal theorems.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The full pipeline is described in §4, with hyperparameters in Appendix F, dataset and split details in Appendix E, and prompt templates in Appendix D. Source code is publicly available at the URL given in the abstract.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code for reproducing our experiments has been released in the repository linked in the abstract. All six datasets are public benchmarks.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: §5 states the few-shot protocol. Appendix E provides dataset and split details. Appendix F lists every hyperparameter used to produce Table 1, and Appendix D reproduces the prompt template per dataset.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Every cell of LLM-GNN Co-Teaching in Table 1 reports mean $\pm$ standard deviation over three seeds. The ablation Table 2 likewise reports mean $\pm$ std over three seeds for every variant.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources and runtime are described in Appendices A, F, and G.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics. All datasets used are public benchmarks released under their original licences, and the code is publicly released.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both directions are discussed in Appendix B. Positive impacts include lowering the labelling barrier in expert-annotation domains. Negative impacts include amplification of LLM-pretraining biases through iterative pseudo-labelling, energy footprint, and privacy concerns when applied to non-consented social graphs.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: We do not release any new high-risk asset. Training code is publicly released, and the fine-tuned LoRA adapter weights are available upon request. The base LLM (Llama-3-8B-Instruct) is loaded from its existing public release under Meta's licence.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We strictly comply with the licenses and terms of use of all resources used in this work.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Code for reproducing our experiments has been released in the repository linked in the abstract, together with a README documenting its structure and usage.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments use existing public benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB review is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: An LLM (Llama-3-8B-Instruct, fine-tuned with LoRA) is a core component of the proposed method. Its role as one of the two co-teaching peers, the prompt template, the SFT and DPO training objectives, and all hyperparameters are documented in §4 and Appendix F. We additionally used a writing-assistance LLM only for grammar and phrasing edits, which has no bearing on the methodology or experimental results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.