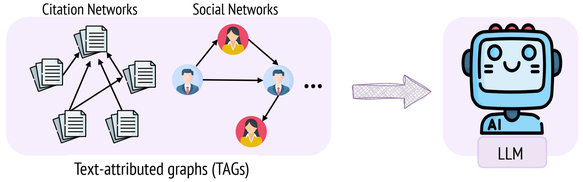
GNN-as-*Judge*: Unleashing the Power of LLMs for Graph Learning with GNN Feedback

Ruiyao Xu, Kaize Ding

Background & Motivation

Introductoin

- LLMs are powerful for text-attributed graph (TAG) learning, leveraging superior semantic understanding to achieve strong node classification performance.
- LLM-as-Enhancers use LLMs to generate enriched node features or embeddings for downstream GNN learning.
- LLM-as-Predictors directly use LLMs to predict node classes as a text generation task, and are more flexible as they enable zero-shot transfer across datasets and label spaces.



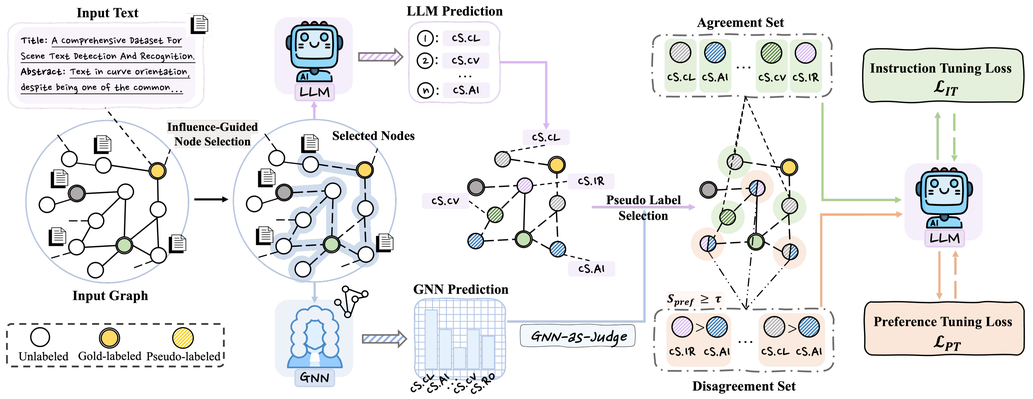
Problem

- LLM-as-Predictors require sufficient labeled nodes for fine-tuning — but real-world graphs are sparsely labeled.
- Under few-shot settings, LLMs lack structural inductive bias, leading to unreliable pseudo-labels and poor generalization.
- Challenge ①: How to generate reliable pseudo-labels beyond LLM's own knowledge on graphs?
- Challenge ②: How to mitigate label noise when fine-tuning LLMs with pseudo-labeled data?

Contributions

- We study LLM-as-Predictors for few-shot semi-supervised node classification on TAGs, a fundamental yet underexplored problem.
- We propose GNN-as-Judge, a novel framework that positions GNNs as judges to select reliable pseudo-labels by exploiting agreement and disagreement between GNNs and LLMs.
- We develop a weakly-supervised fine-tuning algorithm combining instruction tuning on agreement nodes and preference tuning on disagreement nodes, achieving state-of-the-art performance across multiple TAG benchmarks.

Method



Influence-Guided Node Selection for Pseudo Labeling

- Calculate estimated influence score for unlabeled nodes

$$\mathcal{IS}(v_j) = \max_{v_i \in \mathcal{V}_{\text{train}}} \frac{|\mathcal{P}^*_{v_i,v_j}|}{(D^*_{GM})^{h^*}}$$

Nodes closer to labeled nodes with shorter paths and lower-degree intermediaries receive stronger label signals

- Select Top-K most informative unlabeled nodes for pseudo labelling

$$\mathcal{V}_{\text{sel}} = \text{TopK}\left(\{\mathcal{IS}(v_j)\}_{v_j \in \mathcal{V}_{\text{unlab}}}, K\right)$$

Agree/Disagree Node Set Selection with GNN Feedback

- Agreement Set $\mathcal{V}_{\text{agreed}}$:

$$\hat{y}^{\text{GNN}} = \hat{y}^{\text{LLM}}$$

Theorem 2: Intuitively, two models with different inductive biases are less likely to make the same mistake

- Disagreement Set $\mathcal{V}_{\text{disagreed}}$:

$$\hat{y}^{\text{GNN}} \neq \hat{y}^{\text{LLM}}$$

- Preference Score:

$$S_{\text{pref}}(v_i) = P_{\text{GNN}}(\hat{y}^{\text{GNN}} \mid v_i) - P_{\text{GNN}}(\hat{y}^{\text{LLM}} \mid v_i)$$

Select disagreed nodes with $S_{\text{pref}} \geq \tau$

Weakly-Supervised Fine-tuning for Graphs

- Agreement nodes use Instruction-Tuning loss:

$$\mathcal{L}_{\text{IT}}(\theta; x_i, y_i) = -\log p_\theta(y_i|x_i)$$

- Disagreement nodes use Preference-Tuning loss:

$$\mathcal{L}_{\text{PT}}(\theta; x_i, y_{w,i}, y_{l,i}) = -\log \sigma(g_\theta(x_i, y_{w,i}, y_{l,i}))$$

- Unified Objective:

$$\mathcal{L}(\theta) = \mathbb{E}[\mathcal{L}_{\text{IT}}] + \lambda\, \mathbb{E}[\mathcal{L}_{\text{PT}}]$$

IT consolidates correct predictions while PT gets learning signal from noisy labels with GNN feedback

Experiments

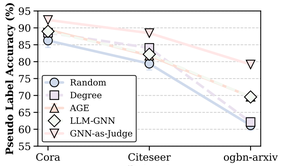
Main Results

Shot	Method	Cora	Citeseer	Pubmed	ogbn-arxiv	ogbn-products
3-shot	GCN	69.45±2.34	63.12±1.02	65.23±1.67	38.33±2.41	59.19±0.79
	SGC	67.21±1.25	63.07±0.95	65.34±1.18	39.16±1.49	57.42±1.12
	Zero-Shot	65.54±0.26	58.17±0.64	74.51±0.39	50.18±1.44	75.48±1.54
	Graph-CoT	63.02±0.77	47.23±1.06	86.22±2.47	49.67±1.23	74.15±1.83
	w. Neighbor	68.72±1.56	54.93±1.28	74.98±3.16	49.28±2.09	76.88±1.07
	GLEM	67.81±0.92	53.09±1.39	63.85±1.02	36.37±1.96	52.46±1.93
	TAPE	73.71±1.86	64.96±0.36	71.33±0.87	48.25±0.77	69.64±1.15
	LLM-GNN	73.85±0.74	61.67±0.58	66.01±0.63	42.36±1.72	55.46±0.37
	LLaGA	54.79±0.91	32.93±1.24	43.96±1.74	29.73±2.82	30.67±1.54
	GraphGPT	57.77±1.62	52.34±1.06	57.51±1.89	31.26±2.73	40.83±2.97
	GNN-as-Judge	**77.89±1.28**	**73.59±0.64**	**87.12±0.89**	**62.21±1.45**	**81.02±1.23**
5-shot	GCN	73.69±1.03	63.24±0.87	70.58±0.49	45.67±0.41	68.26±1.54
	SGC	72.48±0.35	62.08±0.77	70.74±0.94	49.89±1.23	66.78±1.56
	Zero-Shot	65.54±0.26	58.17±0.64	74.51±0.39	50.18±1.44	75.48±1.54
	Graph-CoT	63.02±0.77	47.23±1.06	86.22±2.47	49.67±1.23	74.15±1.83
	w. Neighbor	68.72±1.56	54.93±1.28	74.98±3.16	49.28±2.09	76.88±1.07
	GLEM	74.69±0.46	61.71±0.58	73.29±1.44	39.19±1.57	56.87±1.70
	TAPE	74.28±0.81	67.73±0.48	75.02±0.83	55.22±0.48	77.44±1.32
	LLM-GNN	75.61±1.04	62.37±1.35	74.33±0.95	45.74±1.66	64.01±0.39
	LLaGA	62.88±2.19	43.71±4.36	58.63±1.05	33.74±2.45	37.29±3.12
	GraphGPT	60.17±1.44	51.83±2.24	57.39±3.67	36.25±1.87	44.78±2.34
	GNN-as-Judge	**79.54±0.39**	**74.39±1.63**	**87.49±1.23**	**66.76±0.83**	**81.93±2.21**
10-shot	GCN	78.22±0.89	68.38±1.49	75.33±0.94	50.95±1.77	69.65±0.89
	SGC	78.49±0.37	67.44±0.60	74.98±1.91	51.89±1.23	67.91±0.48
	Zero-Shot	65.54±0.26	58.17±0.64	74.51±0.39	50.18±1.44	75.48±1.54
	Graph-CoT	63.02±0.77	47.23±1.06	86.22±2.47	49.67±1.23	74.15±1.83
	w. Neighbor	68.72±1.56	54.93±1.28	74.98±3.16	49.28±2.09	76.88±1.07
	GLEM	78.11±0.73	66.83±0.61	74.17±2.39	47.73±1.09	60.22±1.89
	TAPE	79.33±0.57	69.39±0.65	77.18±1.06	60.37±0.92	79.53±0.63
	LLM-GNN	79.39±1.26	66.28±0.94	76.82±0.57	52.74±0.48	66.98±0.39
	LLaGA	69.25±0.97	51.22±1.43	67.29±2.26	45.35±1.74	40.55±1.63
	GraphGPT	61.58±0.77	55.40±3.16	71.33±2.81	48.67±1.89	51.46±1.05
	GNN-as-Judge	**80.71±0.83**	**74.62±1.35**	**90.17±1.69**	**67.88±1.03**	**82.48±1.56**

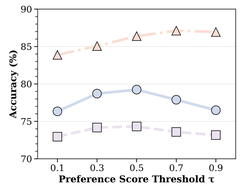
Zero-Shot Node Classification

Train → Test	Model	Accuracy
ogbn-arxiv → Cora	LLAGA	16.24±0.95
	GraphGPT	6.29±0.73
	GNN-as-Judge	**68.27±0.91**
ogbn-arxiv → Citeseer	LLAGA	14.72±1.12
	GraphGPT	5.37±0.84
	GNN-as-Judge	**56.67±0.89**
ogbn-arxiv → Pubmed	LLAGA	30.52±1.18
	GraphGPT	10.54±1.05
	GNN-as-Judge	**83.41±0.76**

Pseudo-Label Selection



Sensitivity Analysis



Diversity Analysis