JOSEIT JOURNAL OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY
http://jurnal.iaii.or.id/index.php/JOSEIT

Efficient Chain-of-Thought Guided Instruction Tuning for Graph Representation Learning

Wang Yiming[1], Li Qian[2*], Du Yuntao[23], Cui Lizhen[12], Yan Zhongmin[1]
[1]School of Software, Shandong University, Jinan 250101, China
[2]Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University, Jinan 250101, China
[3]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

[*]*Corresponding author: wangy@sdu.edu.cn*

Abstract. Large language models (LLMs) reason well over natural language but do not naturally understand graph structure, and directly applying them to graph data runs into two persistent obstacles: a wide gap between the graph and text modalities, and the difficulty of encoding structural information in a way an LLM can use. Existing graph-instruction-tuning methods typically close this gap with simple linear projections and question-answer pairs that carry no explicit reasoning trace, which leaves cross-modal alignment shallow and drives up fine-tuning cost. This paper proposes GraphCoT, a graph representation learning method built on efficient chain-of-thought (CoT) fine-tuning. Its core is a high-quality CoT distillation mechanism in which a strong teacher model generates instruction data containing explicit intermediate reasoning paths, which explicitly teaches a student model to reason in multiple steps from only a small amount of training data. GraphCoT further introduces a graph-text alignment module that maps graph representations into the language embedding space, and a lightweight two-stage training strategy that aligns modalities before task-specific fine-tuning, which keeps training cost low while preserving both reasoning quality and transferability. Node classification and link prediction experiments on multiple benchmark datasets show that GraphCoT outperforms mainstream graph learning methods in accuracy and generalization, confirming the effectiveness of the CoT distillation mechanism and the alignment strategy for graph representation learning.

Keywords: *graph representation learning; chain-of-thought; graph encoder; cross-modal alignment; instruction tuning*

Cite as: Wang Yiming, LI Qian, Du Yuntao, Cui Lizhen, & Yan Zhongmin. (2024). Efficient Chain-of-Thought Guided Instruction Tuning for Graph Representation Learning. Journal of Systems Engineering and Information Technology (JOSEIT), 3(3), 7-13. https://doi.org/10.29207/joseit.v3i3.8453

Received by the Editor: 1-11-2024
Final Revision**:** 20-12-2024
Published**:** 30-12-2024

This is an open-access article under the CC BY-4.0 license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A graph is an abstract mathematical model that describes entities and the relations between them; the basic structure formed by vertices and edges gives it a natural ability to express the interaction patterns of complex systems, and graphs are widely used in social network analysis [1], drug discovery [2], and recommender systems [3]. Beyond explicitly modeling heterogeneous relations, this structural representation has become an important methodological foundation for complex relational reasoning and dynamic pattern mining. In recent years, graph neural networks (GNNs) have made substantial progress on tasks such as node classification and link prediction [4] by mining the node-attribute features and topological information embedded in a graph.

At the same time, large language models (LLMs) have shown outstanding language understanding and reasoning-generation ability in natural language processing (NLP). The chain-of-thought (CoT) reasoning paradigm in particular, which guides a model through explicit intermediate steps to solve complex problems, has markedly strengthened LLM reasoning. Several studies have tried to bring the strong semantic modeling capacity of LLMs into graph learning [5–6], gradually forming a graph-language cross-modal modeling paradigm.

InstructGLM [7] describes graph structure in natural language and combines it with task-specific fine-tuning to improve LLM performance on graph tasks, demonstrating generative graph-reasoning ability, but its strong task dependence limits generality and makes it difficult to transfer to other graph tasks. GraphInstruct [8] builds a benchmark spanning 21 classic graph-reasoning tasks and proposes GraphLM and GraphLM+, which improve graph understanding and reasoning through efficient instruction tuning. InstructGraph [9] introduces a structured, "code-like" formatter to encode graph data uniformly, and combines instruction tuning with preference alignment to improve accuracy and stability on graph-reasoning and graph-generation tasks. GraphGPT [10] adopts a two-stage graph-instruction-tuning paradigm that uses structure injection and a linear alignment module so the LLM can understand and interpret graph-structural components, extending its graph-perception ability.

Despite this progress, efficiently integrating LLMs with graph learning still faces several challenges: (1) Insufficient graph-text alignment: there is no effective cross-modal semantic alignment mechanism between graph structure and text representation, and most existing methods rely on simple linear mappings that struggle to capture

complex nonlinear cross-modal interactions. (2) Weak model reasoning ability: the training data used by existing models is mostly direct question-answer pairs without a reasoning chain, leaving the language model's reasoning ability underdeveloped. (3) No fine-grained tuning mechanism: constrained by the sheer scale of LLMs, some methods rely on full-parameter fine-tuning or automated instruction construction, which drives up cost and weakens generalization.

To address these challenges, this paper proposes GraphCoT, a graph representation learning method based on efficient chain-of-thought fine-tuning, aimed at strengthening an LLM's structural understanding and reasoning transparency on graph tasks. GraphCoT adopts a joint data-structure optimization strategy: on the data side, it constructs high-quality, graph-task-oriented CoT instruction data for efficient fine-tuning with a small amount of data; on the structure side, it designs a graph-text alignment module that aligns graph representations with the LLM's text semantic space. The main contributions of this paper are as follows:

1. A CoT-based instruction distillation strategy for graph tasks: high-quality CoT instruction data automatically generated by a strong LLM is used to guide a student model to explicitly learn multi-step reasoning paths, achieving efficient fine-tuning on a small amount of data while preserving quality by reducing the number of CoT instructions needed.
2. A graph-text modality alignment strategy: a projector from graph structure to language semantic space is designed to strengthen the LLM's understanding of graph-structural semantics.
3. Extensive experimental validation and ablation analysis: on node classification and link prediction, the two basic graph tasks, comparisons against multiple state-of-the-art baselines confirm GraphCoT's clear advantage in accuracy and generalization, and ablation experiments further demonstrate the effectiveness and necessity of each module design.

2. Related Work

2.1 Large Language Models

Large language models have shown excellent performance across a wide range of NLP tasks [11–12] on the strength of their reasoning and problem-solving ability, exhibiting striking capabilities such as language understanding, instruction following, and logical reasoning. Their powerful generative mechanism lets LLMs make explicit the knowledge implicitly learned during large-scale black-box training, and apply it broadly to downstream tasks. To further strengthen generation, researchers have proposed techniques such as in-context reasoning [13] and CoT [14] to guide models toward more logically structured output.

As pre-training scale and corpus quality continue to improve, different LLMs show performance differences in text generation, language understanding, and complex logical reasoning. Closed-source models such as ChatGPT [15] and Claude [16] show greater stability and robustness on complex reasoning and generation tasks; open-source models such as Llama [17–18], DeepSeek [19], and Qwen [20] combine strong performance with notable progress in controllability and extensibility. Despite these breakthroughs, LLMs' reliance on large-scale corpus training also brings high training cost, weak interpretability, and limited performance in multimodal settings.

2.2 Chain-of-Thought Distillation

CoT is a technique that improves an LLM's complex-reasoning ability through a sequence of intermediate reasoning steps [21]. By explicitly asking a model in the prompt to "think step by step," CoT decomposes a complex problem into several simpler sub-problems or reasoning steps. Knowledge distillation [22] is a model-compression paradigm that has been widely applied across many fields. In recent years, CoT distillation has increasingly become an important route for transferring an LLM's chain-style reasoning ability to a student model. Its core idea is to use the reasoning process generated by the LLM as an additional supervisory signal, guiding the student model to learn both the explanatory logic and the final prediction label [22–23].

Some studies approach the problem from an information-theoretic angle to optimize conflicts between multi-task objectives [24]; others, such as KPOD (keypoint-based progressive CoT distillation) [25], target issues like uneven token representation and a fixed learning order by introducing a key-token weighting mechanism together with a progressive distillation strategy to improve training effectiveness. To sharpen the student model's attention to the reasoning path, some work introduces a counterfactual-reasoning objective to safeguard generation faithfulness [26]. On expanding the volume of training data, SCOTT (self-consistent chain-of-thought distillation) [27] shows that multiple reasoning chains can effectively improve model performance; in addition, iteratively constructing reasoning paths in structured forms such as chains, trees, and graphs has also been shown to further improve the effectiveness of knowledge transfer [28].

2.3 Cross-Modal Understanding

Beyond language modeling alone, substantial research has aligned LLMs with multiple modalities, driving the development of multimodal LLMs [29–30]. Work that aligns graph structure with LLMs currently relies mainly on structure-aware prompt design: by converting graph structure into natural-language prompts, it draws on an LLM's semantic understanding to model entity representations and their relations in a knowledge graph, achieving

structured knowledge injection fused with language-model reasoning [31–33]. Filter-then-Generate (FtG) [34] further introduces graph-structure-based instruction tuning that couples structure and text information to optimize representation alignment.

A separate class of methods, such as LLM-GNN [6], has the LLM process graph nodes directly to obtain high-quality annotations that strengthen a GNN's performance on graph-reasoning tasks. LinguGKD (linguistic graph knowledge distillation) [35] proposes a multi-scale contrastive distillation strategy that aligns LLM and GNN feature representations at multiple granularities, from the node level to the graph level. GHGRL (generalized heterogeneous graph representation learning) [36] considers an LLM's language-understanding ability together with heterogeneous-graph modeling needs, and designs an LLM-driven graph representation learning module to align heterogeneous-graph representations. In addition, MSG-LLM (graph-enhanced LLM with multi-scale retrieval) [37] bridges LLMs and graphs by constructing similar graph structures and semantic interaction mechanisms across scales.

3. Method

This paper proposes a graph representation learning method based on efficient chain-of-thought fine-tuning, achieving graph-model optimization guided by a small amount of high-quality training data. The overall GraphCoT framework is shown in Figure 1. The method contains two core technical modules: (1) a CoT-based graph-task instruction generation mechanism for constructing high-quality training samples with clear reasoning paths, and (2) a non-linear projector for modality alignment, which effectively fuses graph-structure representation with the language embedding space.

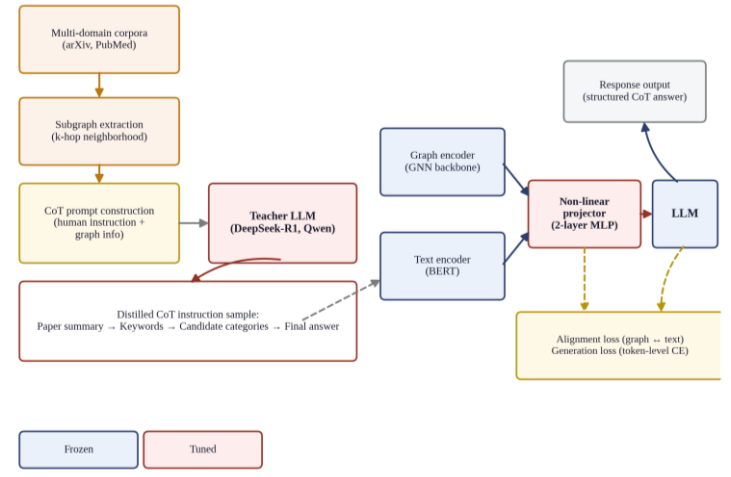


Figure 1. Overall framework of GraphCoT

3.1 Chain-of-Thought Instruction Distillation

To strengthen an LLM's multi-step reasoning ability on graph tasks, this paper introduces a CoT distillation paradigm. This paradigm uses a high-performance teacher model (e.g., DeepSeek-R1, Qwen) to generate structured instruction samples — containing both an explicit reasoning process and a final conclusion — from standard graph tasks, and uses these samples to supervise training of the student model. For the two basic graph tasks of node classification and link prediction, this paper designs differentiated instruction-generation strategies.

For node classification, given a teacher LLM with strong reasoning ability, the instruction design incorporates a manually specified thinking-flow template that guides the model to reason step by step according to preset stages before producing an answer. As the example CoT instruction template below shows, this kind of template effectively induces the teacher model to generate a structured, chain-of-thought-style response, from which a predefined CoT reasoning path can be extracted so the student can effectively learn the teacher's reasoning ability. The goal of CoT distillation is for the student model to acquire the teacher LLM's reasoning transparency and deep semantic modeling capacity by learning this chain-style reasoning structure.

After obtaining an initial batch of samples, GraphCoT introduces an error-feedback mechanism to improve data quality and the robustness of the distilled data: the LLM's preliminary predictions are verified, and any sample predicted incorrectly is fed back into the teacher model together with its ground-truth label, prompting the teacher LLM to produce a detailed explanation consistent with the correct answer and thereby generate a high-quality instruction sample. This mechanism ensures that every distilled instruction sample carries both an accurate conclusion and a clear reasoning process. For link prediction, unlike the classification task, GraphCoT instead uses a staged dialogue template built on self-reflection and error correction. This template uses a two-turn dialogue structure: in the first turn, the LLM is asked the original human-instruction question directly, and its answer is

judged for correctness. This kind of instruction sample not only markedly improves the accuracy of the generated results, but also lets the student model learn the teacher model's reflection-and-correction ability during distillation, strengthening its own cognitive ability and reasoning robustness.

3.2 Graph-Text Semantic Alignment

A natural representational gap exists between graph structure and natural language. How to transfer an LLM's understanding ability to the semantic modeling of graph-structural elements is a significant challenge. GraphCoT proposes a graph-text alignment module that uses a multilayer perceptron (MLP) to achieve modality alignment between graph structure and natural language. For a given graph input G = (E, R, A) and its corresponding set of text descriptions {Ti}, on the text side a pretrained language model such as BERT (bidirectional encoder representations from transformers) [38] is used to encode the text with context awareness, obtaining a text representation vector HT.

On the graph side, a flexible graph-structure encoder module is designed, using a pretrained GNN backbone and its message-passing mechanism to hierarchically aggregate each node and its neighborhood topology in the graph. At each message-passing step, the graph encoder aggregates neighborhood information for the central node:

$$H^{(l+1)} \leftarrow Aggregation\ (Message(H^{l}, \tilde{A}))\quad (1)$$

where Hl denotes the node representation at layer l, and Ã denotes the graph adjacency matrix with added self-loop connections. This layer-wise aggregation process yields the graph-structure representation vector HG.

To streamline the overall tuning process, GraphCoT uses a two-layer MLP projector to build a correspondence between graph tokens and language tokens, mapping the learned graph-structure representation into the graph-token representation space the LLM is skilled at handling. Specifically, for an input feature $H \in R^{dxN}$, the first layer applies a linear transformation followed by the QuickGELU activation function:

$$X = W_1H + B_1,\ \ Z = QuickGELU(X)\quad (2)$$

where $W1 \in R^{d'xd}$ is a weight matrix that projects the input into the hidden space, $B1 \in R^{d'xN}$ is the corresponding bias matrix, and QuickGELU(·) denotes an activation function applied after a single fully connected layer that preserves the smooth nonlinearity of GELU (Gaussian error linear unit) while reducing approximation complexity, benefiting gradient propagation and training convergence.

The second layer is a linear mapping layer that maps the hidden representation back to a target dimension aligned with the language representation space:

$$Y = W_2Z + B_2\quad (3)$$

where $W2 \in R^{dxd'}$ is a weight matrix and $B2 \in R^{dxN}$ is the corresponding bias matrix. During training, the model primarily optimizes the parameters of this projector module to achieve efficient, stable graph-text alignment.

3.3 Lightweight Two-Stage Training Strategy

To avoid the high training overhead caused by frequently updating the LLM and the graph encoder, GraphCoT adopts a lightweight two-stage optimization strategy: throughout training, the backbone model parameters stay frozen, and only the graph-text alignment projector module is updated. The first stage is structural alignment training. To fully capture the contextual information within the graph structure and align graph tokens with their corresponding text semantics, before CoT instruction fine-tuning GraphCoT first performs a graph-matching task [10]: by learning to align graph structure with shuffled node-text pairs, it establishes an initial mapping between graph tokens and semantic information. This stage mainly initializes the parameters of the graph-text alignment module, reducing the burden on the subsequent task-tuning stage.

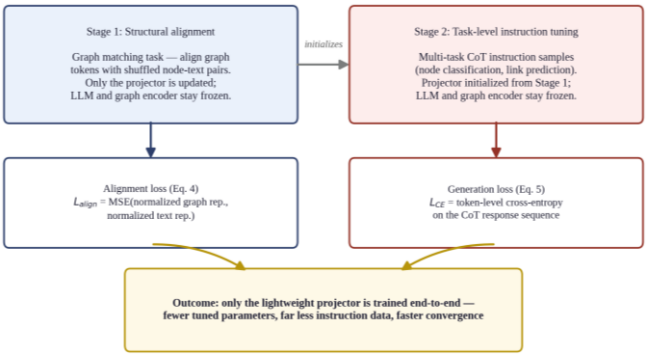


Figure 2. Lightweight two-stage training pipeline

The second stage is task-level instruction fine-tuning. Using the constructed multi-task CoT instruction samples, the model is further optimized to strengthen its understanding and explanation of graph structure. In this

stage, the LLM and graph-encoder parameters remain frozen, and only the projector module is updated. This strategy effectively reduces the number of trained parameters, achieving better results while tuning a minimal parameter set. In particular, thanks to the high-quality CoT instruction distillation strategy, the required volume of training data is further reduced, giving a lightweight and efficient fine-tuning strategy. To balance alignment effectiveness and generation performance, a two-stage composite loss is used. The alignment loss is intended to map graph-structural feature representations into a semantic space consistent with the corresponding language-model hidden state, ensuring that graph information can be effectively received and used by the language model. Mean squared error (MSE) is used as the alignment objective:

$$\hat{h}_i = h_i / \|h_i\|^2, \quad \hat{t}_i = t_i / \|t_i\|^2 \quad (4a)$$

$$L_{align} = (1/N) \Sigma_{i=1}^{N} \|\hat{h}_i - \hat{t}_i\|^2_2 \quad (4b)$$

where $\hat{h}i$ and $\hat{t}i$ denote the L2-normalized graph-structure and text representation vectors for the i-th sample, respectively, and N is the number of samples. The generation loss is introduced once graph-text alignment is complete, during the task-fine-tuning stage, as a token-level cross-entropy loss that provides fine-grained optimization of the LLM's output:

$$L_{CE} = -\Sigma_{t=1}^{T} \ln p(y_t \mid y_{<t}, \text{context}) \quad (5)$$

where yt denotes the t-th target token and context denotes the model input.

4. Results and Discussion

The experiments use three public text-attributed graph datasets: OGB-arxiv, PubMed, and Cora. OGB-arxiv [39] is a directed citation network of computer-science papers on arXiv, where edges denote citation relations; it contains approximately 169,343 nodes and 1,166,243 citation edges, with each node representing a paper manually labeled into one of 40 research areas. PubMed [40] consists of 19,717 diabetes-related medical publications split into three classes — "experimental diabetes," "type 1 diabetes," and "type 2 diabetes" — with a corresponding citation network of 44,338 citation edges. An expanded version of Cora [41] contains 25,120 mutually citing research papers spanning 70 fine-grained categories.

4.1 Evaluation Protocol

For data splitting, OGB-arxiv uses the public split [39], with a train–validation–test ratio of 6:2:3; PubMed and Cora are both split 3:1:1 [40, 42]. Experiments are conducted on two basic graph tasks: node classification and link prediction. Node classification assigns each node to a research-topic category; link prediction determines whether an edge exists between a pair of nodes. Model performance is evaluated using accuracy (Acc) and macro-averaged F1 (Macro-F1) for node classification, and area under the ROC curve (AUC) — a common metric for binary-classification performance — together with average precision (AP) for link prediction.

4.2 Baseline Methods

To ensure a comprehensive evaluation, this paper compares against state-of-the-art models as baselines, grouped into seven categories: (1) an MLP for node representation learning; (2) common graph neural encoders, including GraphSAGE (graph sample and aggregate) [43], GCN (graph convolutional network) [41], GAT (graph attention network) [44], and RevGNN (reversible GNN) [45]; (3) the self-supervised graph-learning method DGI (deep graph infomax) [46]; (4) knowledge-distillation-enhanced GNN methods, such as GKD (geometric knowledge distillation) [47] and GLNN (graph-less neural network) [48]; (5) strong graph-transformer methods, NodeFormer [49] and DIFFormer (diffusion-based transformers) [50]; (6) partially open-source LLMs used as baselines, including Baichuan-7B, vicuna-7B-v1.1, and vicuna-7B-v1.5; and (7) GraphGPT [10]. Except for GraphGPT, all baseline results are taken from the GraphGPT paper or the best results reported in the papers that originally introduced them. The GraphGPT results are those of the GraphGPT-7B-mix-all checkpoint the authors released on Hugging Face.

4.3 Implementation Details

The experiments are implemented with PyTorch and the Transformers library, using vicuna-7B-v1.5-16K as the base model, AdamW as the optimizer, an initial learning rate of 2E-3, and a linear warm-up ratio of 3E-2, on two H800 GPUs with a maximum input length of 4,096 tokens. Training proceeds in two stages: in the self-supervised alignment stage, only the graph-text alignment projector is fine-tuned, for 3 epochs; in the task instruction-tuning stage, the LLM and graph encoder are frozen, the projector is initialized from the first stage, and training runs for 2 epochs. Apart from the number of training epochs, the remaining hyperparameters are kept identical across both stages.

4.4 Overall Performance

To compare overall performance, this paper runs supervised experiments on both node classification and link prediction. The supervised setting means the model is trained on a given dataset and evaluated on that same dataset. Table 1 summarizes overall node-classification performance on the Arxiv and PubMed datasets.

Table 1. Results of Baseline Models and GraphCoT on Node Classification Tasks (%)

Model	Arxiv Acc	Arxiv Macro-F1	PubMed Acc	PubMed Macro-F1
MLP*	51.79	25.36	67.55	—
GraphSAGE*	54.80	32.90	75.39	—
GCN*	52.67	32.02	79.00	—
GAT*	53.32	31.18	79.00	—
RevGNN*	54.74	32.40	—	—
DGI*	50.59	27.87	76.80	—
GKD*	55.70	15.95	86.24	—
GLNN*	60.88	37.57	75.42	—
NodeFormer*	59.22	33.28	81.50	—
DIFFormer*	59.86	33.55	81.80	—
Baichuan-7B*	9.46	3.63	—	—
vicuna-7B-v1.1*	26.57	13.75	—	—
vicuna-7B-v1.5*	49.62	18.53	—	—
GraphGPT-7B	68.45	14.53	74.16	71.37
GraphCoT (ours)	72.73	29.46	86.41	82.64

The node-classification results show that, in the supervised setting, the proposed method outperforms all state-of-the-art baselines on both datasets on the primary Acc metric, improving on the strongest baseline, GraphGPT-7B, by 4.28 percentage points on Arxiv and 12.25 percentage points on PubMed — evidence of stronger overall modeling ability on graph tasks. The Macro-F1 results show a more nuanced picture: in the supervised setting, GNN-based graph-learning models and their variants generally achieve lower accuracy but relatively balanced performance across classes in multi-class prediction (e.g., GLNN reaches a higher Macro-F1 than GraphCoT on Arxiv despite substantially lower Acc), while methods that directly use an open-source LLM perform poorly on both metrics. GraphCoT's design lets the LLM understand the structural properties inherent to a knowledge graph, which raises accuracy and strengthens performance on multi-class problems overall, achieving the best Macro-F1 of any LLM-based method and the best Acc across all baselines.

Beyond its strong node-classification performance, GraphCoT also shows a clear advantage on link prediction. As Table 2 shows, on the PubMed link-prediction setting, GraphCoT substantially outperforms existing baselines on both AUC and AP, reaching 82.24% and 79.52% respectively, far exceeding GraphGPT-7B — further confirming GraphCoT's advantage in graph-structure modeling and reasoning, and its generality and effectiveness.

Table 2. Results of Baseline Models and GraphCoT on Link Prediction Task (%)

Model	PubMed AUC	PubMed AP
MLP*	55.83	58.33
GraphSAGE*	50.41	58.13
GAT*	56.06	63.73
RevGNN*	45.38	50.83
Node2Vec*	65.35	68.85
GraphGPT-7B	64.51	58.86
GraphCoT (ours)	82.24	79.52

Table 3 compares the volume of instruction-tuning data GraphCoT and GraphGPT-7B use across the three datasets under the supervised setting. On Arxiv, GraphGPT fine-tunes on 94,441 instruction samples, while GraphCoT uses only 2,000 — about 2.1% of that volume; on the PubMed node-classification (NC) and link-prediction (LP) subtasks, GraphCoT uses only 700 and 1,000 instruction samples respectively, far below the 23,660 and 50,000 GraphGPT uses. Despite this substantially smaller data scale, GraphCoT achieves better accuracy and generalization on both node classification and link prediction, as shown in Tables 1 and 2 above.

Table 3. Analysis of Instruction Data Volume (Number of Instruction Samples)

Model	Arxiv (NC)	PubMed (NC) / (LP)
GraphGPT-7B	94,441	23,660 / 50,000
GraphCoT (ours)	2,000	700 / 1,000

Figure 3 visualizes this gap on a logarithmic scale. This result confirms that a high-quality CoT distillation mechanism can effectively raise data efficiency, and, paired with a lightweight fine-tuning strategy built around the graph-text alignment structure, lets GraphCoT achieve a genuine "less-is-more" performance breakthrough.

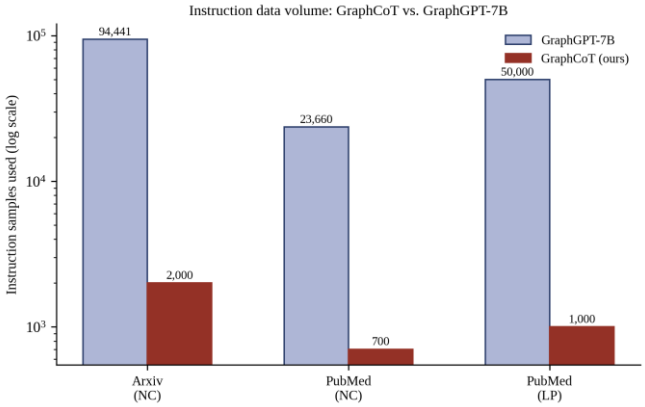


Figure 3. Instruction data volume used by GraphCoT versus GraphGPT-7B across the three benchmark settings (log scale).

4.5 Generalization Analysis

To probe GraphCoT's ability to generalize across domains, a zero-shot experiment is conducted in which the model is trained on one set of datasets and tested on another without additional training. Specifically, the model is trained on Arxiv + PubMed and tested on the independent Cora dataset, examining its ability to adapt to an entirely unseen distribution. The results are shown in Table 4.

Table 4. Node Classification Results of Baseline Models and GraphCoT under Zero-Shot Task Setting, Trained on (Arxiv+PubMed) and Tested on Cora (%)

Model	Cora Acc	Cora Macro-F1
MLP*	2.20	0.06
GraphSAGE*	1.32	0.29
GCN*	1.87	0.32
GAT*	1.61	0.57
RevGNN*	2.17	0.16
DGI*	2.05	0.11
GKD*	4.06	0.37
GLNN*	1.82	0.92
NodeFormer*	1.44	0.53
DIFFormer*	1.00	0.07
Baichuan-7B*	4.05	4.05
vicuna-7B-v1.1*	10.90	9.70
vicuna-7B-v1.5*	14.89	12.13
GraphGPT-7B	15.01	14.51
GraphCoT (ours)	39.52	32.87

As Table 4 shows, GraphCoT reaches 39.52% node-classification accuracy and a Macro-F1 of 32.87% on Cora, clearly ahead of every method compared and best on both metrics, confirming strong generalization in a zero-shot graph-task setting. By contrast, traditional GNN models such as GCN, GAT, and RevGNN achieve only around 1%–2% accuracy on Cora with very low Macro-F1 scores, showing severe performance degradation under an unseen data distribution. Structure-enhanced methods such as GLNN and GKD improve slightly but remain clearly behind overall. Models based on pretrained language backbones (the vicuna series) or graph-text fusion models such as GraphGPT-7B also improve, but still fall short of GraphCoT. These results show that relying solely on a language model or solely on graph-structural information is insufficient for full generalization on graph tasks, whereas the high-quality CoT distillation mechanism and structural-alignment strategy GraphCoT introduces have good transferability and practical potential.

4.6 Ablation Study

To verify the effectiveness of each module design, an ablation analysis is conducted on the node-classification task. In the notation used, "C" denotes the proposed CoT instruction template design, "R" denotes the reflection-and-correction mechanism applied after the initial CoT instruction data is generated, and "P" corresponds to the two-layer MLP projector strategy that achieves graph-text alignment. The overall trend confirms that GraphCoT's combination of the chain-style reasoning ability drawn from a strong LLM with its graph-structure understanding

module gives it a clear advantage in both accuracy and generalization, under both supervised and zero-shot settings.

Removing the graph-text alignment module (−P) causes a clear drop in performance, especially in class-balance behavior across multiple categories, confirming the projector's key role in aligning the graph semantic space. This effect is weaker on simpler tasks (such as the three-way classification on PubMed) and considerably more pronounced on the more challenging zero-shot task, indicating that this structure helps the model capture deeper semantic features and improves its understanding of graph-structural information and its cross-domain transfer ability. It is worth noting that, on the Arxiv dataset, using the variant with the projector, reflection-and-correction, and CoT template all removed (−P−R−C) as a reference point, restoring the CoT template (−P−R) lowers Acc slightly while Macro-F1 rises slightly. These two changes are not contradictory: Acc is more sensitive to classes with a larger sample share, while Macro-F1 more fully reflects the model's discriminative ability on minority classes.

This result therefore shows that the CoT template does not bring a uniform gain on every sample — the intermediate reasoning it generates may introduce extra uncertainty on some samples, lowering Acc, while at the same time improving recognition across different classes, especially minority ones, making overall predictions more balanced. After further adding the reflection-and-correction mechanism (−P), both Acc and Macro-F1 improve, showing that the reflection mechanism can check and correct unreliable intermediate reasoning, easing the decision noise CoT can introduce. Together, these results reflect the complementary relationship between CoT reasoning and reasoning-quality control.

5. Conclusion

Motivated by the complex reasoning demands of graph tasks, this paper proposes GraphCoT, an efficient chain-of-thought fine-tuning method for graph tasks aimed at strengthening a large language model's ability to understand graph structure and reason in multiple steps. By combining a high-quality chain-of-thought distillation mechanism with a lightweight graph-text alignment strategy, GraphCoT achieves efficient fine-tuning on an extremely small volume of instruction data, and shows strong performance across multiple graph-task benchmark datasets. The experiments confirm that the synergy between explicit reasoning supervision and structural-semantic fusion substantially improves graph representation learning. GraphCoT also demonstrates a feasible path for transferring an LLM's general reasoning ability into graph-learning settings, offering a new idea and methodological foundation for building graph-task models that are interpretable, transferable, and scalable.

References

[1] XU X, ZHANG Y F, ZHOU F, et al. Improving multimodal social media popularity prediction via selective retrieval knowledge augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, 39(1): 932-940.
[2] GILMER J, SCHOENHOLZ S S, RILEY P F, et al. Neural message passing for quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning*, 2017: 1263-1272.
[3] WANG S Y, SUI Y D, WANG C, et al. Unleashing the power of knowledge graph for recommendation via invariant learning. *Proceedings of the ACM Web Conference 2024*. New York: ACM, 2024: 3745-3755.
[4] CHEN Y L, WU T X, LIU Y C, et al. Uncertain knowledge graph completion with rule mining. *Proceedings of the 21st International Conference on Web Information Systems and Applications*. Singapore: Springer, 2024: 100-112.
[5] SUN S Y, REN Y X, CHEN J H, et al. Large language models as topological structure enhancers for text-attributed graphs. *Proceedings of the 30th International Conference on Database Systems for Advanced Applications*. Singapore: Springer, 2026: 106-122.
[6] CHEN Z K, MAO H T, WEN H Z, et al. Label-free node classification on graphs with large language models (LLMs). arXiv:2310.04668, 2023.
[7] YE R S, ZHANG C Q, WANG R H, et al. Language is all a graph needs. *Findings of the Association for Computational Linguistics: EACL 2024*. Stroudsburg: ACL, 2024: 1955-1973.
[8] LUO Z H, SONG X R, HUANG H, et al. GraphInstruct: empowering large language models with graph understanding and reasoning capability. arXiv:2403.04483, 2024.
[9] WANG J N, WU J D, HOU Y P, et al. InstructGraph: boosting large language models via graph-centric instruction tuning and preference alignment. *Findings of the Association for Computational Linguistics: ACL 2024*. Stroudsburg: ACL, 2024: 13492-13510.
[10] TANG J B, YANG Y H, WEI W, et al. GraphGPT: graph instruction tuning for large language models. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2024: 491-500.
[11] KOJIMA T, GU S S, REID M, et al. Large language models are zero-shot reasoners. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022: 22199-22213.
[12] WEI J, TAY Y, BOMMASANI R, et al. Emergent abilities of large language models. arXiv:2206.07682, 2022.
[13] MIN S, LYU X X, HOLTZMAN A, et al. Rethinking the role of demonstrations: what makes in-context learning work? *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg: ACL, 2022: 11048-11064.
[14] KUMAR S, MENDKE S, RAHMAN K L A, et al. Enhancing zero-shot chain of thought prompting via uncertainty-guided strategy selection. *Proceedings of the 31st International Conference on Computational Linguistics*. Stroudsburg: ACL, 2025: 2003-2025.
[15] OUYANG L, WU J, XU J, et al. Training language models to follow instructions with human feedback. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022: 27730-27744.
[16] BAI Y T, KADAVATH S, KUNDU S, et al. Constitutional AI: harmlessness from AI feedback. arXiv:2212.08073, 2022.
[17] TOUVRON H, LAVRIL T, IZACARD G, et al. LLaMA: open and efficient foundation language models. arXiv:2302.13971, 2023.
[18] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: open foundation and fine-tuned chat models. arXiv:2307.09288, 2023.
[19] DEEPSEEK-AI, LIU A X, FENG B, et al. DeepSeek-V3 technical report. arXiv:2412.19437, 2024.
[20] XU J, GUO Z F, HE J Z, et al. Qwen2.5-Omni technical report. arXiv:2503.20215, 2025.

[21] WEI J, WANG X Z, SCHUURMANS D, et al. Chain-of-thought prompting elicits reasoning in large language models. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022: 24824-24837.
[22] MAGISTER L C, MALLINSON J, ADAMEK J, et al. Teaching small language models to reason. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: ACL, 2023: 1773-1781.
[23] HSIEH C Y, LI C L, YEH C K, et al. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. *Findings of the Association for Computational Linguistics: ACL 2023*. Stroudsburg: ACL, 2023: 8003-8017.
[24] CHEN X, HUANG H X, GAO Y J, et al. Learning to maximize mutual information for chain-of-thought distillation. *Findings of the Association for Computational Linguistics: ACL 2024*. Stroudsburg: ACL, 2024: 6857-6868.
[25] FENG K T, LI C S, ZHANG X L, et al. Keypoint-based progressive chain-of-thought distillation for LLMs. *Proceedings of the 41st International Conference on Machine Learning*, 2024: 13241-13255.
[26] WANG P F, WANG Z Y, LI Z, et al. SCOTT: self-consistent chain-of-thought distillation. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: ACL, 2023: 5546-5558.
[27] LI L H, HESSEL J, YU Y, et al. Symbolic chain-of-thought distillation: small models can also "think" step-by-step. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: ACL, 2023: 2665-2679.
[28] ZHUANG X W, ZHU Z H, WANG Z C, et al. UniCoTT: a unified framework for structural chain-of-thought distillation. *Proceedings of the 13th International Conference on Learning Representations*, 2025.
[29] LEE Y J, LI C Y, LIU H T, et al. Visual instruction tuning. *Advances in Neural Information Processing Systems 36*, 2023: 34892-34916.
[30] ZHU D Y, CHEN J, SHEN X Q, et al. MiniGPT-4: enhancing vision-language understanding with advanced large language models. arXiv:2304.10592, 2023.
[31] SHU D, CHEN T L, JIN M Y, et al. Knowledge graph large language model (KG-LLM) for link prediction. *Proceedings of the 22nd Asian Conference on Machine Learning*, 2025: 143-158.
[32] MA J, GAO Z T, CHAI Q, et al. Debate on graph: a flexible and reliable reasoning framework for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, 39(23): 24768-24776.
[33] MA S J, XU C J, JIANG X H, et al. Think-on-graph 2.0: deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. arXiv:2407.10805, 2024.
[34] LIU B, ZHANG J H, LIN F Q, et al. Filter-then-generate: large language models with structure-text adapter for knowledge graph completion. *Proceedings of the 31st International Conference on Computational Linguistics*. Stroudsburg: ACL, 2025: 11181-11195.
[35] HU S X, ZOU G B, YANG S, et al. Large language model meets graph neural network in knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, 39(16): 17295-17304.
[36] GAO H, ZHANG C H, WU F G, et al. Bootstrapping heterogeneous graph representation learning via large language models: a generalized approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, 39(16): 16717-16726.
[37] DING J, ZHENG Z, LIN B, et al. MSG-LLM: a multi-scale interactive framework for graph-enhanced large language models. *Proceedings of the 31st International Conference on Computational Linguistics*. Stroudsburg: ACL, 2025: 9687-9700.
[38] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg: ACL, 2019: 4171-4186.
[39] HU W, FEY M, ZITNIK M, et al. Open graph benchmark: datasets for machine learning on graphs. *Advances in Neural Information Processing Systems 33*, 2020: 22118-22133.
[40] HE X X, BRESSON X, LAURENT T, et al. Explanations as features: LLM-based features for text-attributed graphs. arXiv:2305.19523, 2023.
[41] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks. *Proceedings of the 5th International Conference on Learning Representations*, 2017.
[42] WEN Z H, FANG Y. Augmenting low-resource text classification with graph-grounded pre-training and prompting. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2023: 506-516.
[43] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017: 1025-1035.
[44] VELICKOVIC P, CUCURULL G, CASANOVA A, et al. Graph attention networks. *Proceedings of the 6th International Conference on Learning Representations*, 2018.
[45] LI G, MÜLLER M, GHANEM B, et al. Training graph neural networks with 1000 layers. *Proceedings of the 38th International Conference on Machine Learning*, 2021: 6437-6449.
[46] VELICKOVIC P, FEDUS W, HAMILTON W L, et al. Deep graph infomax. *Proceedings of the 7th International Conference on Learning Representations*, 2019.
[47] WU Q T, YAN J C, YANG C X. Geometric knowledge distillation: topology compression for graph neural networks. *Advances in Neural Information Processing Systems 35*, 2022: 29761-29775.
[48] ZHANG S, LIU Y, SUN Y, et al. Graph-less neural networks: teaching old MLPs new tricks via distillation. *Proceedings of the 10th International Conference on Learning Representations*, 2022.
[49] LI Z N, WIPF D P, WU Q T, et al. NodeFormer: a scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems 35*, 2022: 27387-27401.
[50] WU Q T, YANG C X, ZHAO W T, et al. DIFFormer: scalable (graph) transformers induced by energy constrained diffusion. arXiv:2301.09474, 2023.