GRAPH-R1: Incentivizing the Zero-Shot Graph Learning Capability in LLMs via Explicit Reasoning

Anonymous ACL submission

Abstract

Generalizing to unseen graph tasks without task-specific supervision remains challenging. Graph Neural Networks (GNNs) are limited by fixed label spaces, while Large Language Models (LLMs) lack structural inductive biases. Recent advances in Large Reasoning Models (LRMs) provide a zero-shot alternative via explicit, long chain-of-thought reasoning. Inspired by this, we propose a GNN-free approach that reformulates graph tasks—node classification, link prediction, and graph classification—as textual reasoning problems solved by LRMs. We introduce the first datasets with detailed reasoning traces for these tasks and develop GRAPH-R1, a reinforcement learning framework that leverages task-specific rethink templates to guide reasoning over linearized graphs. Experiments demonstrate that GRAPH-R1 outperforms state-of-the-art baselines in zero-shot settings, producing interpretable and effective predictions. Our work highlights the promise of explicit reasoning for graph learning and provides new resources for future research. Codes are available at https://anonymous.4open.science/r/emnlp_submission-FDF0.

1 Introduction

Zero-shot learning in graph machine learning aims to solve tasks in unseen label spaces or domains without any task-specific supervision. While graph neural networks (GNNs) perform well when ample labeled data are available, their generalization ability sharply deteriorates under distribution shifts or in new label spaces—unless expensive fine-tuning is applied (Ju et al., 2023). Prompt-based GNN variants (Liu et al., 2023; Sun et al., 2023), inspired by advances in natural language processing (NLP), offer partial mitigation; however, their fixed, task-specific output heads still hinder true zero-shot generalization.

Large language models (LLMs) offer a complementary and promising alternative. A straightforward approach flattens the graph into a textual sequence and feeds it to an LLM (Chen et al., 2024c; Guo et al., 2023; Wang et al., 2023; Liu and Wu, 2023). However, this often yields suboptimal results due to the lack of structural inductive bias essential for effective graph reasoning (Huang et al., 2024). Recent efforts have sought to more tightly integrate GNNs with LLMs. One line of work retains the GNN as the predictor while using the LLM to generate auxiliary signals, such as synthetic labels or node descriptions (Ye et al., 2024; Yu et al., 2025; Xia et al., 2024; Chen et al., 2024d). Yet, these methods still rely on rigid GNN heads and require retraining for each task. Another approach delegates prediction to the LLM while incorporating structural signals from a frozen GNN via cross-modal projection (Tang et al., 2024; He et al., 2025; Wang et al., 2024a). Unfortunately, the separation of training between components results in weak task conditioning and limited transferability. More tightly coupled methods—such as GOFA (Kong et al., 2024)—inject GNN features directly into the LLM token stream at inference time. While this improves zero-shot accuracy, it introduces substantial computational overhead and still struggles with generalization across tasks and domains.

From *graph structure* to text-based *reason-then-predict*. Recent advances in Large Reasoning Models (LRMs) (e.g., DEEPSEEK-R1 (DeepSeek-AI et al., 2025)) renew our interest in the graph-to-text paradigm, driven by their ability to generate explicit reasoning processes. These models can potentially compensate for the lack of hand-crafted structural priors and offer an interpretable, zero-shot-capable alternative for graph learning. Crucially, many canonical graph tasks—such as link prediction, edge classification and node or graph-level classification—can be naturally refor-

mulated as short sequences of relational reasoning steps, once the graph is linearized into text. Prompting a reasoning-capable LLM to generate such chains of deduction effectively replaces the opaque feedforward process of a GNN with a transparent *reason-then-predict* pipeline. This shift offers two key advantages: improved generalization under distribution shift (since the model must justify each step rather than memorize patterns), and human-interpretable rationales for every prediction. Progress in this direction, however, is currently bottle-necked by the lack of (i) a benchmark that evaluates both reasoning and prediction across a wide range of graph tasks, and (ii) a model that fully commits to this pure, GNN-free paradigm.

Our solution. To tackle the above challenges, we present the first dataset that simultaneously spans node classification, link prediction, and graph classification—each annotated with explicit chains of thought. Leveraging this resource, we develop GRAPH-R1—a reinforcement-learning-based, purely graph-to-text framework that promotes LLM reasoning for cross-task, cross-domain zero-shot prediction without any GNN component. We further design a *rethink reasoning template* specialized for graph prediction. Comprehensive experiments show that GRAPH-R1 matches or exceeds strong baselines, particularly in challenging transfer scenarios, highlighting the value of explicit reasoning for graph learning. Our contributions are threefold:

- We construct the first reasoning dataset tailored for graph machine learning tasks, with detailed reasoning traces.
- We propose GRAPH-R1, a reasoning-augmented LLM that improves generalization and transferability via reinforcement learning with task-specific rethink templates.
- Extensive experiments show state-of-the-art zero-shot performance across diverse graph tasks, demonstrating the impact of explicit reasoning in LLM-based graph prediction.

2 Methods

We present GRAPH-R1, a *graph-to-text* framework for cross-domain and cross-task graph reasoning. The method is built on three principal modules: (i) reasoning-centric graph dataset construction, (ii) reasoning knowledge acquisition via full-parameter fine-tuning, and (iii) reinforcement learning–based reasoning enhancement.

DeepSeek-R1 node classification reasoning

Reasoning Traces:

Okay, Let's try to figure out how to classify `node3`. So, the task is to determine the correct category for the essay represented by...

First, I need to look at `node3`'s own description. It says: "Practical lattice-coded PNC...

Second, the connections suggest that `node3` is part of a network of papers related to physical-layer network coding...

Wait, but the connections listed are `node3-node4`, `node3-node5`...

Therefore, based on the above reasoning, the primary category for `node3` is likely **cs.NI**.

`Answer:` **cs.NI**
`Brief_reasoning:` `node3` is best categorized under `cs.NI` due to its focus on practical lattice-coded physical-layer network coding, with no meaningful structural or semantic connections to other domains.

Figure 1: An illustration of explicit reasoning traces produced by DEEPSEEK-R1 for node classification.

2.1 Notations

Let $\tau \in \mathcal{T}$ denote a graph-learning task type (e.g., node classification, link prediction). A graph is written as $G = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X})$, where $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$ and $\mathcal{E} = \{e_1, \ldots, e_{|\mathcal{E}|}\}$ are the node and edge sets, respectively. The adjacency matrix $\mathbf{A}^G \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ satisfies $A^G_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$. Textual attributes are stored in $\mathbf{X}$: $\mathbf{x}(v_i)$ denotes the textual description of node v_i, and $\mathbf{x}(e_{ij})$ denotes that of edge e_{ij}, which connects nodes v_i and v_j. For a graph–task pair (G, τ) we design a prompt template $P_{G,\tau}$ that linearises G and specifies the task requirements.

2.2 Graph-Reasoning Data Curation

To investigate *reason-then-predict* graph learning, we construct the first dataset featuring explicit, detailed reasoning traces across multiple graph tasks.

Dataset and task selection. We sample 11 representative datasets from five domains—*citation networks, e-commerce, social media, molecular graphs*, and *knowledge graphs*. Together they cover node, edge, and graph-level tasks (node classification, link prediction, graph classification, edge classification), ensuring broad coverage for evaluating graph reasoning.

Graph-to-text augmentation. Unlike prior work that tokenizes structural features using GNN encoders, we revisit the pure graph-to-text paradigm. Taking node-level tasks as an example, for a target node v_i, we extract its h-hop subgraph and describe all node features $T_i = \{\mathbf{x}(v_j) \mid j \in \mathcal{N}(i) \cup \{i\}\}$, and edge relations $E_i = \{\mathbf{x}(e_{jk}) \mid v_j, v_k \in \mathcal{N}(i) \cup \{i\}\}$ within the subgraph using natural language, where $\mathcal{N}(i)$ is the neighborhood of v_i. To maintain input tractability for large graphs with verbose node texts (e.g., citation networks with titles and abstracts), we apply DEEPSEEK-V3 for automatic summarization. Prompt templates are provided in Appendix B.

Reasoning-trace extraction. A distinctive feature of our dataset construction is the inclusion of explicit reasoning traces for each answer. Specifically, each subgraph query Q_i consists of node features T_i, edge relations E_i, and a prompt template $P_{G,\tau}$ tailored to the graph structure G and task type τ, serving as input to the LLM. We then input Q_i into DEEPSEEK-R1 to generate an explicit reasoning trace R_i and a final prediction Y_i, as illustrated in Figure 1. Formally, this process can be represented as:

$$Q_i \rightarrow (Y_i, R_i).$$

Quality control. We apply a three-stage filtering process:

1. **Information sufficiency**: remove isolated nodes and trivial subgraphs.
2. **Answer validity**: discard samples where the predicted answer Y_i mismatches the gold label or contains sensitive content.
3. **Rationale coherence**: retain only rationales that exhibit reasonable length and logical consistency.

The final corpus contains 10,000 graph reasoning examples across multiple domains and tasks, each paired with an explicit chain-of-thought explanation.

2.3 Graph-R1

Building on the graph–reasoning corpus described above, we develop GRAPH-R1, an LLM-based framework for solving graph machine learning tasks through explicit reasoning. Training proceeds in two stages: (1) joint instruction tuning across multiple tasks and domains, and (2) reinforcement learning to refine reasoning quality. To support smaller LLM backbones, we introduce a *rethink* template that encourages deeper semantic and structural analysis, leading to more robust and interpretable multi-step deductions. This pipeline enables GRAPH-R1 to advance zero-shot graph reasoning with large language models.

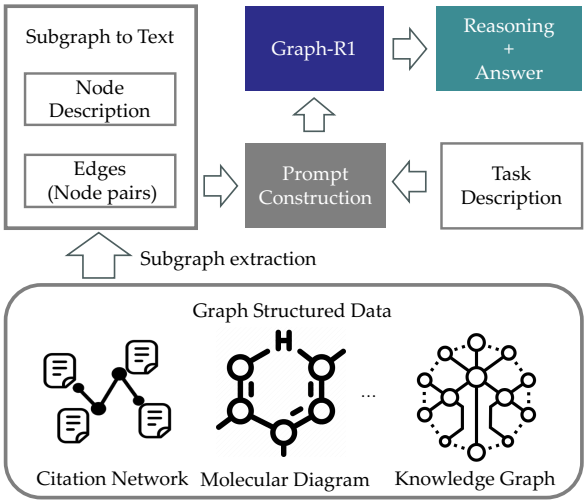

Figure 2: GRAPH-R1 framework. Graphs are linearized into a graph description language, and a task-aware prompt guides the LLM to produce explicit reasoning and the final answer.

2.3.1 Reasoning Knowledge Learning via Full-Parameter Fine-Tuning

In Phase 1, we perform joint instruction tuning across node-, edge-, and graph-level tasks from diverse domains, transferring the general reasoning capabilities of DEEPSEEK-R1 to the graph setting and leveraging multi-task synergies.

We adopt full-parameter supervised fine-tuning using the standard language modeling loss. Given a graph query Q_i—comprising textual node and edge features $\{T_i, E_i\}$ and a prompt $P_{G,\tau}$—the model is trained to generate both the reasoning trace R_i and the final answer Y_i:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log p_\theta\big(R_i, Y_i \mid Q_i\big), \qquad (1)$$

where N is the number of training examples and θ denotes the model parameters. The model thus learns to map graph-structured prompts to coherent reasoning traces and accurate solutions. Exposure to a broad range of tasks enhances generalization and promotes transferable reasoning abilities. Detailed training configurations are provided in Appendix C.

Rethink Prompt Template

Question: (will be dynamically filled)
You must conduct reasoning inside <think>...</think>. Inside it, you should include:
- Include topological analysis in <structure>...</structure>
- Include semantic interpretation in <semantic>...</semantic>
- Provide three candidate answers in <comprehensive>...</comprehensive>
- Re-evaluate each candidate in <rethink>...</rethink>

Your response must follow this format:
<think>
<structure>Structure analysis here</structure>
<semantic>Semantic analysis here</semantic>
<comprehensive>List candidate answers and brief reasoning</comprehensive>
<rethink>Re-evaluate each candidate in depth</rethink>
Final reasoning and answer
</think>
Answer: your_answer
Brief_reasoning: your_brief_reasoning

Figure 3: Rethink Prompt Template. This structure-aware reasoning format is used during both training and inference.

2.3.2 Reinforcement-Learning-Based Reasoning Enhancement

In Phase 2, we refine the instruction-tuned model using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a reinforcement learning method that jointly incentivizes answer correctness and the logical coherence of the reasoning trace—thereby enhancing generalization across graph tasks.

GRPO fine-tunes the supervised model using a reward signal that balances reasoning quality and prediction accuracy. Its training objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim P(Q)} \frac{1}{g} \sum_{i=1}^{g} \Big[\min(\rho_i A_i, \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon) A_i) - \beta\, \text{KL}(\pi_\theta \| \pi_{\text{ref}}) \Big]. \tag{2a}$$

$$\rho_i = \frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)}. \tag{2b}$$

$$\text{KL}(\pi_\theta \| \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - \log \frac{\pi_{\text{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - 1. \tag{2c}$$

Here, π_θ and $\pi_{\theta_{\text{old}}}$ denote the current and previous policies, respectively; q and o_i represent the sampled question and its i-th response; and g is the group size. The hyperparameters ϵ and β control the clipping threshold and KL divergence penalty, respectively. The group-wise advantage is computed as:

$$A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^{g})}{\text{std}(\{r_j\}_{j=1}^{g})},$$

where $\{r_j\}$ are the rewards for the g responses to the same question.

Rethink Template. Conventional prompting typically restricts reasoning to a single <think> block—effective for mathematical problems, but suboptimal for graph tasks where both structural and semantic information are critical, and labels can be ambiguous. To address this, we propose a *rethink* reasoning template specifically designed for graph prediction tasks (see Figure 3).

Our revised template introduces a structured, multi-phase reasoning process:

- <structure>: encourages explicit topological analysis;
- <semantic>: focuses on the interpretation of node/edge attributes;
- <comprehensive>: elicits multiple candidate answers to expose alternative hypotheses;
- <rethink>: revisits each candidate to encourage comparative and bidirectional evaluation.

This structure-aware prompting scheme enables tighter integration of topology and semantics, significantly improving RL performance on node classification, link prediction tasks and etc.

Reward Modeling. The reward function serves as the core training signal in reinforcement learning. Under the *standard* <think> template, we adopt a simple reward scheme:

$$R = \begin{cases} 1 & \text{if the answer is correct,} \\ 0.01 & \text{if the output is merely well-formatted,} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

While sufficient for toy mathematical tasks, this coarse-grained feedback overlooks the rich interme-

diate reasoning required for graph-based problems. To address this, we design a more fine-grained reward for the *rethink* template, which evaluates both the reasoning trace and the final answer. During the initial reasoning phase, the model lists multiple candidate answers; partial credit is assigned if the gold label appears among them:

$$R = \begin{cases} 1 & \text{if the final answer is correct,} \\ 0.3 & \text{if the correct answer appears in } \texttt{<rethink>}, \\ 0.01 & \text{if only the format is correct,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Coupled with GRPO, this refined reward enables the model to learn richer and more reliable reasoning paths, leading to state-of-the-art zero-shot performance across all evaluated graph domains and tasks.

3 Experiments

We begin by introducing the datasets used to train and evaluate GRAPH-R1 (§3.1), followed by the baselines and experimental setup (§3.2). We then present a comprehensive suite of experiments to assess the effectiveness and generalization of our method, focusing on the following questions: **RQ1:** Does GRAPH-R1 enable critical applications of general graph models, such as zero-shot learning? **RQ2:** Can it generalize to unseen tasks and domains, including cross-task transfer? **RQ3:** How do instruction tuning and the *rethink* template contribute to generalization? **RQ4:** How does GRAPH-R1 compare to large reasoning models on graph tasks?

3.1 Datasets

We evaluate GRAPH-R1 on five benchmark datasets:

- Cora — citation network with node and link prediction tasks (Wen and Fang, 2023).
- Products — e-commerce graph for node classification (He et al., 2024a).
- WikiCS — Wikipedia graph with node classification (Mernyei and Cangea, 2020).
- FB15K237 — knowledge graph for link prediction (Liu et al., 2024).
- Expla-Graph — synthetic graph reasoning benchmark (He et al., 2024b).

All tasks are aligned with the evaluation protocol of GOFA (Kong et al., 2024). To test cross-domain and cross-task generalization, we additionally evaluate on three unseen graph regression datasets—ESOL (Withnall et al., 2018), Lipo (Wu et al., 2017), and Freesolv (Casasnovas et al., 2014)—which are not seen during either fine-tuning or reinforcement learning. This ensures a strict zero-shot cross-task setting. Dataset statistics and task details are provided in Appendix A.

3.2 Experimental Setup

Baselines. We compare against two groups of baselines:

- *General-purpose LLMs*: LLaMA 2-7B (Touvron et al., 2023), Mistral-7B (Jiang et al., 2023), and DeepSeek-R1-distilled-Qwen2.5-14B (DeepSeek-AI et al., 2025);
- *Graph models leveraging LLMs*: OFA (Liu et al., 2024), GraphGPT (Tang et al., 2024), UniGraph (He et al., 2025), ZeroG (Li et al., 2024a), LLaGA (Chen et al., 2024b), and GOFA (Kong et al., 2024).

These baselines represent the current state-of-the-art in both general LLM and graph-specific LLM paradigms, providing a rigorous comparison for our proposed approach.

Implementation. We instantiate GRAPH-R1 with DeepSeek-R1-distilled-Qwen2.5 models (14B). The model is first instruction-tuned and then further optimized with GRPO-based reinforcement learning on our graph reasoning dataset. All methods, including baselines, are evaluated under consistent zero-shot conditions and identical hardware. Hyperparameters are tuned based on validation performance. Full training details, data splits, and evaluation metrics are available in Appendix C.

3.3 Cross-Dataset Zero-Shot Generalization (RQ1)

To address RQ1, we run strict zero-shot evaluations on the GOFA-aligned benchmarks listed in §3.1. Table 1 yields the following observations. Generic LLMs such as Llama2-7B and Mistral-7B rely mainly on textual cues. They are competitive on node-classification datasets, where semantics dominate, but drop sharply on link-prediction tasks that require relational reasoning. LLM-as-predictor models (GOFA, UniGraph) consistently surpass GNN-based hybrids (OFA, ZeroG). Encoding graph structure into the LLM token stream or feature space markedly improves cross-domain robustness.

Task	Cora–Node		WikiCS		Products			Expla-Graph	Cora–Link	FB15K237
Way / Type	7	2	10	5	47	10	5	2	2	10
Llama2-7B	47.92	73.45	40.10	58.77	27.65	58.71	64.33	57.76	48.15	48.32
Mistral-7B	60.54	88.39	63.63	71.90	43.99	70.16	74.94	68.77	49.43	62.48
OFA	28.65	56.92	21.20	35.15	19.37	30.43	39.31	51.36	52.22	–
GraphGPT	44.65	–	–	–	18.84	–	–	–	50.74	–
UniGraph	69.53	**89.74**	43.45	60.23	38.45	66.07	75.73	–	–	–
ZeroG	64.21	87.83	31.26	48.25	31.24	51.24	71.29	–	–	–
LLaGA	51.85	62.73	–	–	23.10	34.15	39.72	–	**88.09**	–
GOFA-T	70.81	85.73	71.17	80.93	54.60	79.33	87.13	79.49	85.10	73.59
GOFA-F	69.41	87.52	68.84	80.52	56.13	80.03	88.34	71.34	86.31	**80.69**
Graph-R1	**71.53**	89.08	**78.68**	**86.89**	**66.59**	**85.72**	**91.78**	**89.71**	86.31	75.17

Table 1: Zero-shot accuracy (%) across datasets. Best in **bold**; second best underlined.

Model	MAE ↓		
	ESOL	Lipo	FreeSolv
LLaGA	7.39	15.55	51.72
GOFA	4.93	**1.36**	14.98
GRAPH-R1	**1.72**	1.55	**11.59**

Table 2: Zero-shot graph regression results (lower MAE is better). Best in **bold**; second best underlined.

GRAPH-R1 attains the best accuracy on eight of ten settings and the second best on the rest, without any GNN encoder. Its graph-to-text reformulation plus reinforcement-learned reasoning allows the model to fuse topology and semantics purely in natural-language form, setting a new state of the art for zero-shot graph prediction.

3.4 Cross-Task Zero-Shot Generalization (RQ2)

To evaluate the model's generalization ability in the zero-shot cross-task setting, we conduct a test in which the model is trained solely on classification-style tasks—node, edge, graph classification or link prediction—and is evaluated on *unseen* graph regression tasks. We compare GRAPH-R1 with two representative LLM-based baselines, LLaGA and GOFA, both evaluated under the same zero-shot setting without access to regression training data. Results are shown in Table 2.

GRAPH-R1 achieves the best performance on ESOL and FreeSolv and ranks second on Lipo, outperforming all baselines without any task-specific tuning. These results highlight its strong cross-task generalization—crucial for real-world deployment where labeled data are often scarce or unavailable.

3.5 Ablation Study (RQ3)

To address **RQ3**, we ablate two key components of GRAPH-R1: instruction tuning and reinforcement learning (RL) with the *rethink* template. Specifically, we compare four variants: (i) *init* (the initial model without task-specific training), (ii) *w/o RL* (instruction-tuned without RL), (iii) *normal* (RL with the standard template), and (iv) the full *Graph-R1* (RL with the rethink template).

Effect of Instruction Tuning. As shown in Figure 4, instruction tuning alone consistently outperforms the initial model across all datasets. This demonstrates effective knowledge transfer and the distillation of graph reasoning capabilities from DeepSeek-R1 to our model, significantly enhancing its graph-specific inference performance.

Effect of RL with the Rethink Template. To better illustrate the effect of reinforcement learning, we compare the *normal* and *Graph-R1* variants using the *w/o RL* variant as the baseline. Results are presented in Figure 5, where the y-axis denotes normalized performance (i.e., the ratio of the performance to the w/o RL baseline). Applying RL with the standard template improves performance primarily on text- and logic-oriented tasks (e.g., Cora–Node, Expla-Graph), but leads to degradation on structure-heavy tasks such as Cora–Link

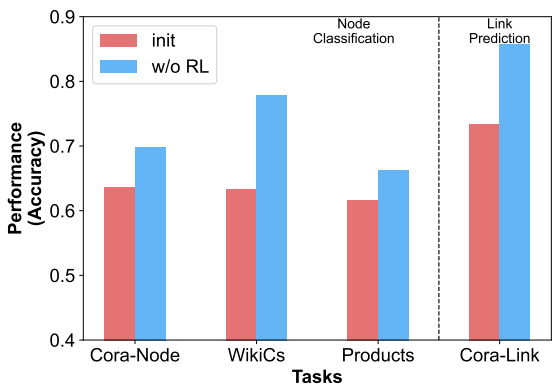

Figure 4: Comparison the results of initial model and instruction-tuned model.

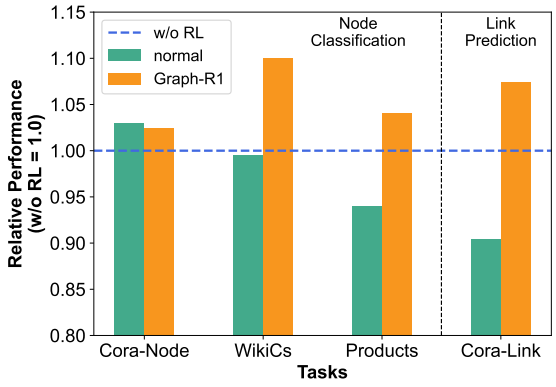

Figure 5: Comparison of RL performance with the standard template ("normal") and the rethink template ("rethink"), using the w/o RL result as the baseline (indicated by the blue dashed line at 1.0). All values are normalized with respect to the w/o RL baseline for each dataset.

and FB15K237, suggesting limited gains in structural reasoning. In contrast, RL with the rethink template yields consistent improvements across all tasks, underscoring its importance in enhancing both semantic and structural understanding, and thereby significantly boosting generalization.

3.6 Comparison with Large Reasoning Models (RQ4)

Answering **RQ4** is crucial for understanding both the necessity of our two-phase training strategy and the effectiveness of GRAPH-R1 as a general-purpose graph model. We compare GRAPH-R1 with Large Reasoning Models (LRMs), including DeepSeek-R1 (671B) and its 14B distilled variant. All models are evaluated using identical input formats to ensure a fair comparison. Due to the high computational cost of DeepSeek-R1, we randomly sample 1,000 examples from the evaluation dataset introduced in Section 3.1 for testing. The results on these samples are presented in Table 3.

GRAPH-R1 achieves substantial improvements over the 14B distilled model across all evaluated tasks. Moreover, in several cases, it matches or even surpasses the performance of DeepSeek-R1 (671B). These results provide strong evidence that our two-phase training strategy significantly enhances reasoning capabilities on graph tasks.

3.7 Case Studies of GRAPH-R1 Reasoning

To demonstrate the interpretability and reasoning capabilities of GRAPH-R1, we present two illustrative examples from its inference process on distinct graph tasks: node classification and link prediction. Due to space constraints, the full case details are provided in Table 10 in Appendix D. These examples highlight key aspects of the model's reasoning process, including its ability to integrate structural and semantic information, comprehensively evaluate candidate options, and effectively verify hypotheses.

In node classification, the model showcased its ability to comprehensively evaluate multiple candidate categories by combining structural and semantic analyses, prioritizing the most relevant category, and systematically re-evaluating each candidate to confirm its conclusion. For link prediction, GRAPH-R1 excelled in hypothesis testing during the rethink phase, where it formulated and rigorously tested assumptions about potential connections, ultimately rejecting unsupported hypotheses with clear reasoning.

4 Related Work

4.1 Pre-training and Fine-tuning for Graphs

The success of foundation models has inspired graph researchers to adopt a *pre-train–then-fine-tune* paradigm. Early efforts focused on self-supervised learning for graphs, where models such as GraphMAE (Hou et al., 2022, 2023), GraphCL (Ying et al., 2021), DGI (Velickovic et al., 2019), GCC (Qiu et al., 2020), and GCA (Zhu et al., 2020) are pre-trained on large-scale graph corpora and then fine-tuned for downstream tasks. More recent approaches explore *graph prompting*, where general-purpose pre-trained GNNs are adapted via textual or task-oriented prompts—for example, All-in-One (Sun et al., 2023) and GraphPrompt (Liu et al., 2023). However, these methods remain con-

Task	Cora-Node	WikiCS	Products	ExplaGraphs	Cora-Link	FB15K237
Way/Type	7	10	47	2	2	10
DeepSeek-R1-Distill-Qwen-14B	60.67	69.33	57.33	81.33	72.00	34.00
DeepSeek-R1-671B	68.67	76.00	**69.33**	**92.00**	68.00	**84.67**
Graph-R1	**72.67**	**78.67**	65.33	88.67	**86.67**	72.00

Table 3: Comparison between Graph-R1 and Large Reasoning Models (LRMs). Best in **bold**; second best underlined.

strained by the inherent architectural limitations of GNNs. As a result, their transferability is often limited to in-domain tasks and typically requires task-specific fine-tuning or additional parameters for optimal performance.

4.2 LLMs for Graph Learning

Graph-to-Text. Several studies transform subgraphs into natural-language prompts for LLMs (Chen et al., 2024c; Liu and Wu, 2023; Wang et al., 2023). However, subsequent analyses have found that ignoring structural information significantly degrades performance (Huang et al., 2024).

LLMs as Feature Enhancers. A common strategy is to leverage LLMs to embed heterogeneous node and edge attributes into a unified semantic space (Ye et al., 2024; Yu et al., 2025; Chen et al., 2024d). For instance, OFA (Liu et al., 2024) verbalises graph metadata and encodes it into dense language embeddings that augment the graph with enriched features. ZeroG (Li et al., 2024a) and OpenGraph (Xia et al., 2024) adopt similar approaches. Nonetheless, these methods often depend on downstream predictors—typically graph neural networks (GNNs)—and are thus limited in the range of tasks they can effectively support.

LLM as Unified Predictor. An emerging line of research treats the LLM itself as the task head, bypassing traditional graph-specific predictors. GraphGPT (Tang et al., 2024) and GOFA (Kong et al., 2024) align graph embeddings with the LLM embedding space and apply instruction tuning for downstream adaptation. UniGraph (He et al., 2025) and TEA-GLM (Wang et al., 2024a) introduce lightweight projection modules to enable zero-shot generalisation, while LLaGA (Chen et al., 2024b) tokenises entire graphs directly for LLM-based inference. While these approaches are promising, embedding alignment can incur information loss, and relying solely on answer-only decoding often under-utilises the full reasoning capabilities of LLMs.

4.3 Reasoning on Graphs

Recent work has begun to assess the reasoning ability of LLMs on graph-structured problems. GPT4Graph (Guo et al., 2023) evaluates GPT-4 on algorithmic tasks such as connectivity and max-flow, revealing encouraging results but limited scalability. NLGraph (Wang et al., 2023) proposes a broad benchmark, showing that while LLMs manage simple instances, they struggle with structural complexity; instruction tuning offers only marginal improvements. GraphWiz (Chen et al., 2024a) focuses on algorithmic reasoning (e.g., shortest paths), but omits standard learning tasks. InstructGraph (Wang et al., 2024b) enhances supervised learning with natural-language instructions, yet falls short on cross-task generalisation. We introduce the first LLM-based framework to jointly integrate reinforcement learning and explicit reasoning, aiming to generalise across diverse graph tasks.

5 Conclusion

We presented GRAPH-R1, a GNN-free paradigm that formulates graph learning tasks—such as node classification, link prediction, and graph classification—as textual reasoning problems solvable by Large Reasoning Models (LRMs). To support this, we introduced the first reasoning dataset for graph machine learning, featuring detailed reasoning traces. Guided by task-specific *rethink* templates, GRAPH-R1 enables LRMs to reason over linearized graph structures. Extensive experiments show that GRAPH-R1 outperforms strong baselines in zero-shot settings while producing interpretable predictions that expose its reasoning process. Our results highlight the promise of explicit reasoning for graph learning and open new directions at the intersection of graph learning and LRMs.

Limitations

While GRAPH-R1 shows strong zero-shot generalization and produces inherently interpretable reasoning across diverse graph tasks, it faces challenges when scaling to very large graphs. Current Large Reasoning Models (LRMs) have input length constraints, and linearizing large or complex graphs may exceed their context window. Future work may explore more efficient encoding methods to improve scalability.

References

Rodrigo Casasnovas, Joaquin Ortega-Castro, Juan Frau, Josefa Donoso, and Francisco Muñoz. 2014. Theoretical pka calculations with continuum model solvents, alternative protocols to thermodynamic cycles. *International Journal of Quantum Chemistry*, 114(20):1350–1363.

Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. 2024a. Graphwiz: An instruction-following language model for graph computational problems. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 353–364.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. 2024b. Llaga: Large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 7809–7823, Vienna, Austria. JMLR.org.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. 2024c. Exploring the potential of large language models (llms)in learning on graphs. *SIGKDD Explor. Newsl.*, 25(2):42–61.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2024d. Label-free node classification on graphs with large language models (llms). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Anna Gaulton, Louisa J. Bellis, A. Patrícia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. 2012. Chembl: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(Database-Issue):1100–1107.

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

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024a. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024b. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. 2025. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, pages 448–459, New York, NY, USA. Association for Computing Machinery.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. 2023. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 737–746.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. Graphmae: Self-supervised masked graph autoencoders. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 594–604.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*.

Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. 2024. Can llms effectively leverage graph structural information through prompts, and why? *Preprint*, arXiv:2309.16595.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mingxuan Ju, Tong Zhao, Qianlong Wen, Wenhao Yu, Neil Shah, Yanfang Ye, and Chuxu Zhang. 2023. Multi-task self-supervised graph neural networks enable stronger task generalization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan Zhang. 2024. Gofa: A generative one-for-all model for joint graph language modeling. In *The Thirteenth International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. 2024a. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 1725–1735, New York, NY, USA. Association for Computing Machinery.

Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Wai Kin (Victor) Chan, and Jia Li. 2024b. Glbench: A comprehensive benchmark for graph with large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Chang Liu and Bo Wu. 2023. Evaluating large language models on graphs: Performance insights and comparative analysis. *Preprint*, arXiv:2308.11224.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 417–428.

Péter Mernyei and Catalina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *CoRR*, abs/2007.02901.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. 2025. Tinyzero. https://github.com/Jiayi-Pan/TinyZero. Accessed: 2025-01-24.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1150–1160.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *Preprint*, arXiv:2402.03300.

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pages 2120–2131, New York, NY, USA. Association for Computing Machinery.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pages 491–500, New York, NY, USA. Association for Computing Machinery.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024a. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36:30840–30861.

Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian J. McAuley. 2024b. Instructgraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and Virtual Meeting, August 11-16, 2024*, pages 13492–13510.

Zhihao Wen and Yuan Fang. 2023. 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, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 506–516.

Michael Withnall, Hongming Chen, and Igor V. Tetko. 2018. Matched molecular pair analysis on large melting point datasets: A big data perspective. *ChemMedChem*, 13(6):599–606.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. 2017. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564.

Lianghao Xia, Ben Kao, and Chao Huang. 2024. Opengraph: Towards open graph foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2365–2379, Miami, Florida, USA. Association for Computational Linguistics.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, Lichao Sun, Xing Xie, and Senzhang Wang. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 40–48.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2024. Language is all a graph needs. *Preprint*, arXiv:2308.07134.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual*, pages 28877–28888.

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. 2025. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 13087–13095.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *CoRR*, abs/2403.13372.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *Preprint*, arXiv:2006.04131.

A Details of Datasets

All of the public datasets used in our paper were previously published, covering a multitude of domains. We report the detailed statistics for each dataset in Table 4. The detailed descriptions of these datasets are listed in the following:

Arxiv Arxiv (Hu et al., 2020) is a large-scale citation graph derived from arXiv Computer Science papers. Each node corresponds to a paper and edges represent citation links between papers. The task is to classify each paper into one of 40 arXiv subcategories, such as "cs.LG" or "cs.AI". This dataset serves as a representative benchmark for large-scale node classification.

Citeseer The Citeseer (Yang et al., 2016) dataset is a citation network comprising research papers and their citation relationships within the computer science domain. Each node represents a research paper, and each edge signifies a citation relationship between two papers.

Cora The Cora (Wen and Fang, 2023) dataset is a citation graph where each node corresponds to a research paper, and each edge represents a citation link between papers. The dataset focuses on papers within the machine learning domain and includes 70 fine-grained categories, making the classification task particularly difficult.

Pubmed Pubmed (He et al., 2024a) is a citation network of biomedical research papers from the PubMed database. Each node is a paper and edges correspond to citation links. The classification task involves assigning each paper to one of three disease-related categories.

Children The Children (Yan et al., 2023) dataset is a co-purchased or co-viewed product graph focused on children's books. Nodes correspond to individual books, and edges connect books that were frequently browsed or bought together. Each node is associated with textual information including the book's title and descriptive metadata.

Computer The Computer (Yan et al., 2023) dataset is co-purchased or co-viewed product graph, where each node represents a product in the computer category, and edges indicate that two products were frequently co-purchased or co-viewed by users. The textual content associated with each node consists of user-generated reviews for the corresponding product.

Photo The Photo (Yan et al., 2023) dataset is an e-commerce product graph where nodes represent photographic products, and edges indicate that two items were either co-purchased or co-viewed by users. The textual content of each node consists of user reviews associated with the corresponding product.

Products The Photo (He et al., 2024a) dataset is an e-commerce product graph where nodes represent Amazon products, and edges indicate that two items were either co-purchased or co-viewed by users. The textual content of each node consists of user reviews associated with the corresponding product.

Sports The Sports (Yan et al., 2023) dataset is a co-purchased or co-viewed product graph in the sports domain. Nodes represent sports-related products, and edges indicate that two items were often purchased or viewed together. The associated text for each node consists of the product's title.

FB15K237 FB15K237 (Liu et al., 2024) is a large-scale knowledge graph where each node represents an entity (e.g., a person, location, or object) and each edge corresponds to a relational triple connecting two entities. Textual content for nodes is constructed from entity names and relation descriptions.

WN18RR WN18RR (Liu et al., 2024) is another knowledge graph extracted from WordNet. It contains 40,943 nodes and 93,003 relations where each node is an English word and each edge represents the relation between two words.

WikiCS WikiCS (Mernyei and Cangea, 2020) is a web link network constructed from English Wikipedia articles related to computer science. Nodes are individual articles, and directed edges represent hyperlinks between them. The node text is the full content of each article.

CHEMBL ChEMBL (Gaulton et al., 2012) is a molecular graph dataset where each graph corresponds to a chemical compound. Nodes represent atoms, and edges denote chemical bonds. The textual information for each molecule is given by its SMILES (Simplified Molecular Input Line Entry System) representation.

BBBP The BBBP (Wu et al., 2017) dataset comes from a study focused on modeling and predicting the permeability of the blood-brain barrier.

Domain	Dataset	Avg.#Nodes	AVG.#Edges	#Classes	#Graphs
Social Network	Instagram	11339	155349	2	1
Web Link	WikiCS	11701	216123	10	1
Logical Graph	Expla_Graph	5.17	4.25	-	2766
Knowledge Graph	FB15K237	14541	310116	237	1
	WN18RR	40943	93003	11	1
Citation	Arxiv	169343	1166243	40	1
	Citeseer	3186	8554	6	1
	Cora	2708	10556	7	1
	Pubmed	19717	88648	3	1
E-commerce	Children	76875	1554578	24	1
	Computer	87229	721081	10	1
	Photo	48362	500939	12	1
	Products	54025	144638	47	1
	Sports	173055	1773500	13	1
Molecular	CHEMBL	25.87	55.92	1048	23874346
	BBBP	24.06	51.91	2	2039
	ESOL	13.29	27.35	-	1128
	Freesolv	8.72	16.76	-	642
	HIV	25.51	54.94	2	41127
	Lipo	27.04	59	-	4200
	PCBA	25.97	56.20	128	34017170

Table 4: Datasets Statistics (the "-" means that it is not appropriate to use the number of classes description. This is because Esol, Freesolv, Lipo is regression tasks, Expla_graph is a Q-A task).

The BBBP dataset contains binary labels indicating whether a compound can penetrate the blood-brain barrier (BBB) or not.

ESOL The ESOL (Withnall et al., 2018) dataset contains water-solubility data for chemical compounds. Each molecule is modeled as a graph, with node and edge structures corresponding to atoms and bonds. SMILES strings serve as the textual representation.

Freesolv Freesolv (Casasnovas et al., 2014) consists of molecular graphs used for estimating hydration free energy. Each molecule is modeled by a graph of atoms and bonds. The SMILES representation is used as the text-based molecular description.

HIV The HIV (Wu et al., 2017) dataset consists of molecular graphs representing candidate compounds for HIV treatment. Nodes denote atoms and edges are chemical bonds. Each molecule is described by its SMILES string.

Lipo Lipo (Wu et al., 2017) is a molecular dataset focused on lipophilicity prediction. Each molecule is represented 524 as a graph with atoms as nodes and bonds as edges. The SMILES string encodes each molecule's 525 structure in text form.

PCBA PCBA (Wu et al., 2017) is a large-scale molecular dataset for virtual screening. Each graph is a molecule, modeled by atoms and bonds, with SMILES strings representing the underlying chemical structure.

Expla_Graph Expla_Graphs (He et al., 2024b) is a graph question answering dataset on commonsense concepts. Each graph in Expla_Graphs contains commonsense concepts connected by its relation.

Instagram Instagram (Li et al., 2024b) is a social graph in which each node represents a user, and edges denote social connections such as following relationships. The textual content associated with each node is extracted from users' self-introductions or profile descriptions.

B Prompt Template

For each specific task type τ, we design an appropriate prompt template to guide the model in understanding and solving the corresponding graph reasoning task. Our prompt templates are systematically constructed and consist of three main components: task-specific and dataset-related prompt prefix and question template, and a format-constrained instruction template. The instruction template is further categorized into two variants: *normal* and *rethink*.

The prompt prefix provides necessary background information, context, and relevant details about the graph structure and node attributes. The question template then formulates the concrete prediction or reasoning objective for the current instance. The instruction template standardizes the output format, ensuring consistency and clarity in model responses.

A comprehensive overview of all prompt templates used for different task types is provided as follows: Table 6 shows prompt templates of node/link classification, Table 7 shows prompt templates of graph classification, Table 8 shows prompt templates of link prediction, and Table 9 shows prompt templates of graph regression.

In addition, we present a dedicated prompt template for summarizing node descriptions within graph reasoning tasks. This template is designed to effectively capture and condense the essential attributes and contextual information of individual nodes, facilitating more accurate and interpretable reasoning by the model. The detailed design of the node summary prompt template is provided in Table 5.

C Details of Implementation

Datasets For the construction of our graph reasoning dataset, we initially collected 348,000 instances from 11 diverse graph datasets: *Arxiv*, *Citeseer*, *Pubmed*, *Instagram*, *Children*, *Computer*, *Photo*, *Sports*, *Chemblpre*, *Chempcba*, and *Wn18rr*. Following the data filtering procedures described in Section 2.2, we curated a high-quality subset comprising 10,000 instances, which serves as the training datasets for instruction fine-tuning and reinforcement learning. For evaluation, we adopt the datasets reported in the GOFA paper, and construct evaluation datasets by inserting into our prompt templates to ensure consistency and comparability.

Baselines Details For all baseline methods, we report the results as provided in the GOFA paper. Since our evaluation datasets are constructed to be consistent with those reported in GOFA, the results are directly comparable and ensure a fair

Template Name	Content
Summary	summary each node's content in no more than 25 words. Your response should strictly be in forms as follows: nodex:<your summary> eg: node1:optimality of myopic sensing in multi channel opportunistic access {node descriptions}

Table 5: Prompt Templates in summarizing node descriptions

Template Name	Content
Prompt Prefix	Classify the <target: essay / book / electronic product / user / product / fitness-related item / wikipedia page> represented by node <node_id> using its subgraph data (text attributes and connections) as follows: Node description: <node description> Connection relationship among the nodes: <connection>
Question Format	Consider both semantic and structural information. Select strictly from: {labels}. Respond only with the category name and briefly summarize the reasoning process.
Normal Instruction	Your reasoning and response should be streamlined and restricted to within 2048 tokens. Your response should be in forms as follows: Answer: your_answer (e.g., {sample_answer}) Brief_reasoning: your_brief_reasoning
Rethink Instruction	You must conduct reasoning inside <think>...</think>. Inside <think>...</think>, you should include: - Structure information: <structure>...</structure> - Semantic similarities: <semantic>...</semantic> After structure and semantic analysis, you msut provide {candidate} candidate answers with brief reasoning inside <comprehensive>...</comprehensive>. Then, you must conduct re-reasoning inside <rethink>...</rethink>. In this section, you should detailed consider each of your candidate answers as if they were the correct answer and evaluate their feasibility. After re-reasoning, you must conduct your final answer based on your above analysis. Finally, besides your reasoning, give your final response. Your full response must follow this format: <think> <structure>Here show your structure analysis</structure> <semantic>Here show your semantic analysis</semantic> <comprehensive>Here show your comprehensive reasoning and list your candidate answers</comprehensive> <rethink>Here ongoing re-reasoning with each of your candidate answers inversely</rethink> Here show your final reasoning and answers </think> Answer: your_answer (e.g., {sample_answer}) Brief_reasoning: your_brief_reasoning

Table 6: Prompt Templates of Node/Link Classification

Template Name	Content
Prompt Prefix	Determine whether the chemical compound represented by the following molecular graph (nodes with atomic features and bond relationships) is predicted to exhibit activity (effectiveness) in each of the provided bioassays. Bioassays descriptions: <bioassays_descriptions> Node description: <node description> Connection relationship among the nodes: <connection>
Question Format	Your response must include: A sequence of strict 'Yes' or 'No' answers for each property in order, separated by spaces (e.g., {sample_answer}), and a concise explanation for your choices, referencing important structural features and the biological assay context.
Normal Instruction	Your reasoning and response should be streamlined and restricted to within 2048 tokens. Your response should be in forms as follows: Answer: your_answer (e.g., {sample_answer}) Brief_reasoning: your_brief_reasoning
Rethink Instruction	- Same reasoning and output format as Node/Link Classification. - Only the task context differs; follow the steps and output structure above.

Table 7: Prompt Templates of Graph Classification

Template Name	Content
Prompt Prefix	Classify the relationship between two <target: essays / books / electronic products / products / fitness-related items / wikipedia pages/entities> denoted as node <node_id> and node <node_id>, using the union of their corresponding subgraph (text attributes and connections) as follows: Node description: <node description> Connection relationship among the nodes: <connection>
Question Format	Consider: semantic and structural information. In your reasoning process provide the predicted connection bond value of the two target nodes between 0 and 1, set the threshold to 0.5. Based on your predicted connection bond value select strictly from: 'Yes, they have {target: citation/co-purchased or co-viewed} relationships' or 'No, they do not have {target: citation/co-purchased or co-viewed} relationships'. Respond only with the choice content and briefly summarize the reasoning process.
Normal Instruction	Provide an estimated connection bond value (ranging from 0 to 1). A higher value indicates a stronger likelihood of a relationship. Consider multiple factors, such as: - Structural information: Evaluate the direct and indirect connections between the two target nodes through their neighbors. - Semantic similarities: Analyze the relevance or similarity in meaning between the two target nodes. - Comprehensive information: If there exist two nodes that are semantically similar to each other, and these two nodes are respectively connected to the two target nodes, this can indirectly indicate the strength of the connection between the target nodes. Your reasoning and response should be streamlined and restricted to within 2048 tokens. Your response should follow this format: Answer: your_answer Brief_reasoning: your_brief_reasoning Bond_value: your_predicted_bond_value
Rethink Instruction	Provide an estimated connection bond value (ranging from 0 to 1). A higher value indicates a stronger likelihood of a relationship. Consider multiple factors, such as: - Structural information: Evaluate the direct and indirect connections between the two target nodes through their neighbors. - Semantic similarities: Analyze the relevance or similarity in meaning between the two target nodes. - Comprehensive information: If there exist two nodes that are semantically similar to each other, and these two nodes are respectively connected to the two target nodes, this can indirectly indicate the strength of the connection between the target nodes. If you can identify direct or indirect connections based on structural information, set the bond strength to 1 and specify the path(s) of connection in your reasoning. If no such connections can be identified, evaluate the bond strength based on the semantic similarity between the target node and its neighboring nodes semantics. You must conduct reasoning inside <think>...</think>. Inside <think>...</think>, you should include: - Structure information within <structure>...</structure> - Semantic similarities within <semantic>...</semantic> After structure and semantic analysis, provide comprehensive information inside <comprehensive>...</comprehensive> Then, you must conduct re-reasoning inside <rethink>...</rethink> . In this section, you should detailed consider each of the two given answers as if they were the correct answer and evaluate their feasibility. After re-reasoning, you must conduct your final answer based on your above analysis. Finally, besides your reasoning, give your final response. Your full response must follow this format: <think> <structure>Here show your structure analysis</structure> <semantic>Here show your semantic analysis</semantic> <comprehensive>Here show your comprehensive reasoning</comprehensive> <rethink>Here ongoing re-reasoning with each of the two candidate answers inversely</rethink> Here show your final reasoning and answers </think> Answer: your_answer Brief_reasoning: your_brief_reasoning Bond_value: your_predicted_bond_value

Table 8: Prompt Templates of Link Prediction

Template Name	Content
Prompt Prefix	Calculate the chemical relevant properties using the given molecular graph (nodes with atomic features and bond relationships) as the following calculation requirements. Calculation requirements: {description} Calculate the {target} of this molecule. Node description: <node description> Connection relationship among the nodes: <connection>
Question Format	Your Response Must Include: A numerical answer, and the mathematical solution process, referencing important structural features and the biological assay context.
Normal Instruction	You are a chemistry expert assistant specialized in molecular graph regression tasks. Given a molecular graph with atomic features and bond relationships, you are asked to approximate the target value using the formula mentioned in calculation requirement Your task is to: - Analyze the molecular structure based on the provided nodes and edges. - Identify key chemical features that influence the target value (e.g., number and position of Cl atoms, ring systems, stereochemistry, hydrogen bonding capability). - Estimate the target value based on the formula. - Provide a final numeric prediction rounded to two decimal places. Please adjust the units of your final result so that the numerical value falls within the range of -30 to 30. Round the result to two decimal places. Respond strictly in the following format: Answer: your_answer (keep two decimal places, e.g., {sample_answer}) Brief_reasoning: your_brief_reasoning
Rethink Instruction	You are a chemistry expert assistant specialized in molecular graph regression tasks. Given a molecular graph with atomic features and bond relationships, you are asked to approximate the target value using the formula mentioned in calculation requirements. Your task is to: - Analyze the molecular structure based on the provided nodes and edges. - Identify key chemical features that influence the target value (e.g., number and position of Cl atoms, ring systems, stereochemistry, hydrogen bonding capability). - Estimate the target value based on the formula. - Provide a final numeric prediction rounded to two decimal places. You must conduct reasoning inside <think>...</think>. Inside<think>...</think>, you should include: - Structure information within <structure>...</structure> - Semantic similarities within <semantic>...</semantic> After structure and semantic analysis, you must provide the range of target with brief reasoning inside <comprehensive>...</comprehensive> Then, you must conduct re-reasoning inside <rethink>...</rethink>. In this section, you should detailed consider your target range as if it were the correct range and evaluate its feasibility. After re-reasoning, you must conduct your final answer based on your above analysis. Finally, besides your reasoning, give your final response. Please adjust the units of your final result so that the numerical value falls within the range of -30 to 30. Round the result to two decimal places. Your full response must follow this format: <think> <structure>Here show your structure analysis</structure> <semantic>Here show your semantic analysis</semantic> <comprehensive>Here show your comprehensive reasoning</comprehensive> <rethink>Here ongoing re-reasoning with each of the two candidate answers inversely</rethink> Here show your final reasoning and answers </think> Answer: your_answer (keep two decimal places, e.g., {sample_answer}) Brief_reasoning: your_brief_reasoning

Table 9: Prompt Templates of Graph Regression

evaluation.

Details of GRAPH-R1 Graph-R1 is developed based on DeepSeek-R1-distilled-Qwen2.5-14B. We employ a two-stage training pipeline: supervised instruction fine-tuning followed by reinforcement learning with the rethink template. In supervised instruction fine-tuning, We utilize the LLaMA-Factory toolkit(Zheng et al., 2024) to perform full-parameter supervised fine-tuning. The learning rate is set to 1e-5, and all other hyperparameters follow their default settings. For the reinforcement learning stage, we adopt the GRPO training strategy implemented via the Tiny-Zero toolkit(Pan et al., 2025). For GRPO, wo use a batch size of 64, set learning rate to 1e-6, and apply the number of sampled outputs per question of 5. All other parameters are set to their default values. For inference, we utilize the VLLM (Kwon et al., 2023) framework to enable efficient batched decoding. All experiments are conducted on a cluster equipped with 8 NVIDIA A100 GPUs.

D some Cases from Graph-R1

We provide some interesting cases from GRAPH-R1's responses to illustrate the interpretability and principles behind its reasoning, as shown in Table 10.

Task Type	Structure	Semantic	Comprehensive	Rethink
Node Classification	Central node (node0) connects to multiple nodes in a distributed system context. **Details:** Node0 connects to node1, node3, node6, and node7. Node7 links to node2, node3, node4, and node0. Other connections include node5 to node6 and node4 to node7.	Focuses on distributed systems, load balancing, and consensus algorithms. **Details:** Keywords like "distributed load balancing," "plurality consensus," "majority dynamics," and "multi-agent systems" dominate. Node4 mentions distributed learning dynamics, linking to machine learning.	Identifies three candidates based on structure and semantics. **Details:** 1. cs.DC (Distributed Computing): Core focus on distributed algorithms, load balancing, and protocols. 2. cs.MA (Multiagent Systems): Multi-agent dynamics mentioned in node2 and node7. 3. cs.LG (Machine Learning): Distributed learning dynamics in node4.	Evaluates the feasibility of each candidate. **Details:** cs.DC: Most fitting due to focus on distributed systems and algorithms. cs.MA: Relevant but secondary to distributed systems. cs.LG: Specific to node4, less central than distributed computing
Link Prediction	Evaluates direct and indirect connections between Node0 and Node1. **Details** Node0 connects to nodes related to hashing methods (node2, node3) and genome processing (node7). Node1 connects to nodes focused on backscatter communication systems (node4, node8, node9). No direct or indirect links between the two target nodes.	Analyzes the thematic relevance between Node0 and Node1. **Details:** Node0 focuses on OLAP for text analysis in humanities research, while Node1 is about MIMO SWIPT power optimization in wireless communication. No overlapping keywords or shared technical domains.	Considers intermediary nodes and shared themes to predict connection strength. **Details:** Node0's connections (node2, node3) focus on hashing methods and algorithms. Node1's connections (node4, node8) focus on communication systems and throughput optimization. No shared intermediary nodes or overlapping application areas.	Tests hypotheses assuming citation relationships exist. **Details:** If OLAP techniques were used in MIMO systems optimization, it would bridge the gap, but no evidence supports this. Alternatively, if MIMO SWIPT used OLAP for data analysis, the connection would still require shared terminology, which is absent. The domains remain distinct—humanities research vs. wireless communication engineering.

Table 10: Case studies for different graph tasks analyzed from structural, semantic, comprehensive, and rethink perspectives.