TRN-R1-Zero: Text-rich Network Reasoning via LLMs with Reinforcement Learning Only

Yilun Liu, Ruihong Qiu and Zi Huang
School of Electrical Engineering and Computer Science
The University of Queensland
Brisbane, Queensland, Australia
{yilun.liu, r.qiu, helen.huang}@uq.edu.au

Abstract

Zero-shot reasoning on text-rich networks (TRNs) remains a challenging frontier, as models must integrate textual semantics with relational structure without task-specific supervision. While graph neural networks rely on fixed label spaces and supervised objectives, recent large language model (LLM)-based approaches often overlook graph context or depend on distillation from larger models, limiting generalisation. We propose TRN-R1-Zero, a post-training framework for TRN reasoning trained solely via reinforcement learning. TRN-R1-Zero directly optimises base LLMs using a Neighbour-aware Group Relative Policy Optimisation objective that dynamically adjusts rewards based on a novel margin gain metric for the informativeness of neighbouring signals, effectively guiding the model toward relational reasoning. Unlike prior methods, TRN-R1-Zero requires no supervised fine-tuning or chain-of-thought data generated from large reasoning models. Extensive experiments across citation, hyperlink, social and co-purchase TRN benchmarks demonstrate the superiority and robustness of TRN-R1-Zero. Moreover, relying strictly on node-level training, TRN-R1-Zero achieves zero-shot inference on edge- and graph-level tasks, extending beyond cross-domain transfer. The codebase is publicly available at https://github.com/superallen13/TRN-R1-Zero.

1 Introduction

Text classification is a cornerstone task in natural language processing, underpinning applications from information retrieval to content recommendation. Yet, in real-world scenarios, texts seldom exist in isolation: scientific papers cite one another, Wikipedia pages are interlinked through hyperlinks, users in social networks follow each other, and e-commerce products often co-occur in purchases. These relational connections naturally form text-rich networks (TRNs), where nodes correspond to textual entities and edges capture their semantic or functional relations. As illustrated in Figure 1, TRNs from citation, hyperlink, social, and co-purchase domains exhibit rich relational structures that go beyond isolated document understanding (Tang et al., 2024c,b, 2026; Chen et al., 2025; Wu et al., 2024; Wang et al., 2025a; Liu et al., 2023, 2025b,a). Effectively reasoning over such TRNs, particularly in zero-shot classification settings without domain-specific supervision, is a crucial step toward more generalisable and context-aware language intelligence.

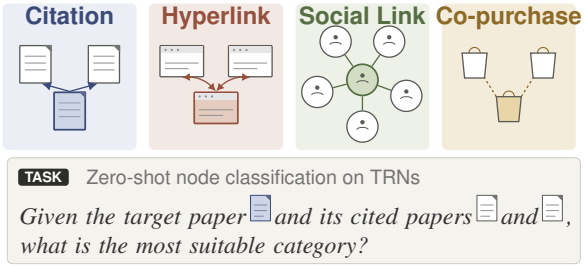


Figure 1: **Top**: Examples of text-rich networks (TRNs) from citation, hyperlink, social and co-purchase domains. **Bottom**: An example of reasoning-based user query over TRNs.

Existing large language model (LLM)-based methods for node classification on TRNs generally follow two paradigms. (1) Encoder-based approaches use LLMs as text encoders for both node and label descriptions (Li et al., 2024a; Fang et al., 2025; Wang et al., 2025b). The resulting embeddings are aggregated through structure-aware mechanisms across neighbouring nodes, and classification is performed via node-label similarity. However, these methods largely treat the LLM as a feature extractor rather than an explicit reasoner. (2) Generative approaches, on the other hand, reformulate node classification as a label-token generation task. To incorporate structural information, some employ soft-embedding tech-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18060–18072
July 2-7, 2026 ©2026 Association for Computational Linguistics

Model	LRM	CoT SFT	Reason.
GraphWiz	✓ (GPT-4)	✓ (GPT-4)	✓
Graph-NPH-R1	✗	✓ (QwQ-32B)	✓
Graph-R1	✓ (DeepSeek-V3)	✓ (DeepSeek-R1)	✓
TRN-R1-Zero	✗	✗	✓

Table 1: Comparison of reasoning training for LLMs on graph tasks. For graph theory problems, GraphWiz (Chen et al., 2024a) and Graph-NPH-R1 (Wang et al., 2025c) rely on mimicking the CoT process of larger LLMs. For text-based network problems, Graph-R1 (Wu et al., 2025b) and GraphWiz further depend on external LRMs to provide reasoning supervision. Our TRN-R1-Zero achieves reasoning ability without relying on LRMs or their generated CoT data.

niques that align graph encodings with the LLM's embedding space (Tang et al., 2024a; Wang et al., 2024; Chen et al., 2024b; Kong et al., 2024), while others use natural language descriptions of graph structures as inputs (Wang et al., 2023; Chen et al., 2024a; Huang et al., 2024; Li et al., 2024b; Wu et al., 2025a). Recent work on both text-based or non-text networks attempt to transfer reasoning abilities from large reasoning models (LRMs) by fine-tuning on chain-of-thought data (Chen et al., 2024a; Wang et al., 2025c; Wu et al., 2025b). Despite these advances, **existing paradigms struggle to directly elicit explicit reasoning within LLMs on TRNs**, while often relying on additional supervision or external reasoning resources.

To address these limitations, we propose TRN-R1-Zero, a reinforcement learning-based framework that enables explicit reasoning on TRNs without any supervised fine-tuning or external distillation. Unlike encoder-based methods that treat LLMs merely as feature extractors, or generative approaches that depend on pre-generated reasoning traces, TRN-R1-Zero learns to reason relationally through direct optimisation over the underlying graph context. We develop a novel Neighbour-aware Group Relative Policy Optimisation objective with a margin gain metric, which leverages local neighbourhood information as adaptive signals to guide reasoning training, allowing it to effectively infer structural and semantic relationships for node classifications on text-rich networks in unseen domains. This design activates the LLM's reasoning capability intrinsically, rather than relying on external supervision or task-specific data. A comparative summary of existing paradigms and TRN-R1-Zero is provided in Table 1. Our main contributions are:

1. An RL-only pipeline for zero-shot node classification on TRNs, without distillation, SFT, or external LRMs.

2. A neighbour-aware policy objective with a margin gain mechanism that explicitly encourages the use of relational context.

3. Extensive experiments on citation, hyperlink, social, and co-purchase TRNs demonstrate consistent zero-shot gains in cross-domain and cross-task settings over prior methods.

2 Related Work

Large Language Models for Node Classification. Existing approaches to zero-shot node classification on text-rich networks (TRNs) can be categorised into encoder-based and generative paradigms.

Encoder-based methods use language models (LMs) or LLMs primarily as text encoders, generating embeddings for nodes and labels that are subsequently aligned or aggregated by external algorithms. ZeroG (Li et al., 2024a) fine-tunes Sentence-BERT (Reimers and Gurevych, 2019) with LoRA (Hu et al., 2022) to effectively encode both node texts and label descriptions. UniGLM (Fang et al., 2025) fine-tunes BERT (Devlin et al., 2019) into a more generalised text encoder through contrastive learning, boosting the downstream graph model performance trained in this embedding space. TAPE (He et al., 2024a) fine-tunes DeBERTa (He et al., 2021) with explanations and predictions generated from an extra LLM. Nevertheless, these fine-tuned encoders exhibit limited generalisation ability because of the small model size and data scarcity of the tuning phase. LLM-BP (Wang et al., 2025b) employs LLM2Vec (BehnamGhader et al., 2024) as a text encoder and applies propagation-based techniques to integrate neighbour information. These methods, however, fail to exploit the explicit reasoning capabilities of LLMs.

Generative methods formulate node classification as a text generation task. GraphGPT (Tang et al., 2024a), GOFA (Kong et al., 2024), TEA-GLM (Wang et al., 2024), and LLaGA (Chen et al., 2024b) employ a learnable mapping model to project graph structures into the LLM's token embedding space, creating soft embeddings that enable the LLM to generate graph-aware representations after supervised fine-tuning. Alternatively,

other works describe graph information directly in natural language, enabling the LLM to understand over structural context without explicit graph encoders (Huang et al., 2024; Li et al., 2024b; Wu et al., 2025a).

Large Language Model Reasoning. LLMs trained with reinforcement learning have demonstrated impressive reasoning abilities and human-like performance across a range of tasks, including mathematics (Shao et al., 2024; Balunovic et al., 2025), task planning (Hu et al., 2024), code generation (Jimenez et al., 2024), and debugging (Zhong et al., 2024). Proximal Policy Optimisation (PPO)(Schulman et al., 2017) serves as the foundation for reasoning-oriented fine-tuning. The recent Group Relative Policy Optimisation (GRPO)(Shao et al., 2024; Guo et al., 2025) introduces a rule-based objective that enables reasoning skills without human-annotated supervision, while Dr.GRPO (Liu et al., 2025c) further enhances reward shaping and variance control during reasoning training.

The **reasoning ability of LLMs has recently been extended to structured data**. For general graphs without textual attributes, GraphWiz (Chen et al., 2024a) and Graph-NPH-R1 (Wang et al., 2025c) leverage large reasoning models (LRMs) to generate chain-of-thought (CoT)(Wei et al., 2022) data for reasoning over graph-theoretic problems such as shortest path and connectivity. Graph-R1(Wu et al., 2025b) further targets text-rich graphs, using LRMs to produce long CoT traces that supervise the fine-tuning of smaller models. In contrast, TRN-R1-Zero removes the need for distillation or externally generated CoT data from larger LRMs, directly eliciting reasoning ability within the base model itself.

3 Methodology: TRN-R1-Zero

To perform node classification with reasoning on text-rich networks, this section describes how TRN-R1-Zero achieves this capability through a novel optimisation paradigm, as illustrated in Figure 2.

3.1 Zero-shot Node Classification

Given a text-rich network $G = (V, E, Y)$, where $V = \{v_1, \ldots, v_{|V|}\}$ denotes the set of nodes with associated texts, $E \subseteq V \times V$ denotes the set of edges, and $Y = \{y_1, \ldots, y_{|Y|}\}$ denotes the set of label texts, the objective is to predict the label of a target node $v_i \in V$ without any supervision from the target network.

Classification as Token Generation. Given a large language model $\mathcal{M}_\theta$, the input comprises the text of the target node t_i, its neighbourhood $\mathcal{N}(v_i)$, and the candidate label texts Y. Each class $y \in Y$ is mapped to a discrete identifier token (e.g., "1", "2", "3"). Thus, node classification is reformulated as a next-token prediction task:

$$\hat{y}_i = \arg\max_{y \in Y} P_\theta\big(y \mid \mathcal{P}(t_i, \mathcal{N}(v_i), Y)\big), \quad (1)$$

where $\mathcal{P}(\cdot)$ denotes the constructed prompt that integrates node, neighbour, and label information.

3.2 Prompt with Neighbourhood Sampling

The input to the LLM is constructed by combining the target node text, sampled neighbour texts, and candidate label descriptions into an instruction-style prompt (see Box 1 below). In our prompt design, neighbourhood sampling serves a dual purpose: it controls the input length and acts as a form of data augmentation. For each target node v_i, multiple subgraphs are randomly sampled following a fixed width–depth strategy, where (i) **width** limits the number of included neighbours, and (ii) **depth** truncates the text of each neighbour. By repeatedly drawing diverse subsets of neighbours, the LLM is exposed to varied local contexts, effectively expanding the training corpus and mitigating overfitting in low-resource graph settings.

Box 1: Train Prompt for TRN-R1-Zero

```
# System Prompt
You are a helpful AI Assistant that provides
well-reasoned and detailed responses. You first
think about the reasoning process as an internal
monologue and then provide the user with the
answer. Respond in the following format:
<think>
...
</think>
<answer>
...
</answer>

# Graph Prompt
Target node: {target_node_text}
Neighbour nodes: {neighbor_node_text}

# Task Instruction
I provide the content of the target node and
its neighbour nodes.  Each node content is
{node_type}.  The relation between the target
node and its neighbour nodes is {relation}. The
{num_categories} categories are: {labels}.
Question: Based on the information of the target
and neighbour nodes, predict the category ID (0
to {max_id}) for the target node.
```

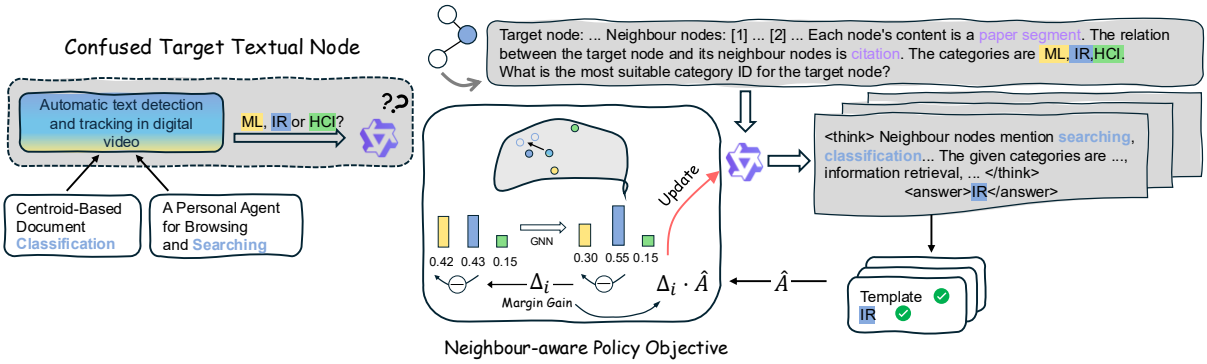


Figure 2: Overall training pipeline of TRN-R1-Zero, comprising three key components: graph sampling, prompt construction, and neighbour-aware policy objective.

3.3 Neighbour-aware Group Relative Policy Optimisation Objective

Reinforcement learning for LLM post-training builds upon the GRPO objective (Shao et al., 2024), a variant of PPO (Schulman et al., 2017) adapted for sequence-level rewards. Given a query q and an output sequence $o = (o_1, \ldots, o_{|o|})$, the objective is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[\sum_{t=1}^{|o|} \min\Big(r_t \hat{A}_t, \; \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\, \hat{A}_t \Big) \right]. \quad (2)$$

where $r_t = \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}$ is the token-level importance sampling ratio, and $\hat{A}_t$ is the advantage estimator.

$$\hat{A}_t = \frac{R_i - \bar{R}}{\text{std}(R)}. \quad (3)$$

A KL regularisation term

$$-\beta \cdot \text{KL}[\pi_\theta(\cdot \mid q, o_{<t}) \,\|\, \pi_{\text{ref}}(\cdot \mid q, o_{<t})] \quad (4)$$

is added to penalise deviation from a frozen reference policy π_{ref}, stabilising optimisation.

This formulation can hinder reward shaping, as the standard deviation term dampens variations in R_i. Dr.GRPO (Liu et al., 2025c) addresses this by removing the denominator:

$$\hat{A}_t = R_i - \bar{R}. \quad (5)$$

This allows shaped rewards to influence optimisation magnitude directly. Although Dr.GRPO is commonly implemented without KL, empirical results in this task reveal that omitting KL causes unstable training. Therefore, the adopted objective is Dr.GRPO with KL regularisation, which preserves both stability and effective scaling.

Margin Gain: Quantifying Neighbouring Contribution. During reasoning over neighbouring nodes, the neighbourhood information may either complement or distract from the target node's textual signal. To identify cases where neighbour information plays a pivotal role, we introduce a margin gain metric that quantifies how much the classification decision boundary shifts after incorporating neighbours.

Let $e_i = f(x_i) \in \mathbb{R}^d$ be the embedding of node text x_i and $e_c = f(y_c)$ the embedding of label text y_c, where $f(\cdot)$ is a frozen text encoder. The raw logit of node i for class c is

$$\ell_{i,c} = e_i^\top e_c. \quad (6)$$

Let y_i denote the ground-truth class of node i. The raw margin score is defined as

$$m_i(\ell) = \ell_{i,y_i} - \max_{c\neq y_i} \ell_{i,c}, \quad (7)$$

which measures how confidently the encoder classifies the node text in isolation.

To measure the influence of neighbours, we apply a lightweight K-layer Simple Graph Convolution (SGC)-style aggregator (Wu et al., 2019):

$$\tilde{E} = \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)^K E, \quad (8)$$

where A is the adjacency matrix with self-loops, D is its degree matrix, and E stacks all node embeddings row-wise. The aggregated embedding $\tilde{e}_i$ induces aggregated logits

$$\tilde{\ell}_{i,c} = \tilde{e}_i^\top e_c, \quad (9)$$

and a corresponding aggregated margin

$$m_i(\tilde{\ell}) = \tilde{\ell}_{i,y_i} - \max_{c\neq y_i} \tilde{\ell}_{i,c}. \quad (10)$$

Therefore, the margin gain can be defined to quantify the contribution of the neighbourhood:

$$\Delta_i = m_i(\tilde{\ell}) - m_i(\ell), \tag{11}$$

which captures how much neighbourhood aggregation improves (or degrades) the classification margin.

Intuitively, $\Delta_i > 0$ indicates that neighbours provide helpful context; $\Delta_i \approx 0$ suggests neighbourhood information is redundant; and $\Delta_i < 0$ implies neighbours are distracting. We use the absolute value $|\Delta_i|$ to measure the strength of neighbourhood influence, regardless of whether the effect is positive or negative.

Reward Design with Margin Gain. Reinforcement learning with GRPO assigns each prompt a scalar reward R_i, which determines the magnitude of policy gradient updates through the advantage estimator. Rather than treating all prompts equally, we scale the rewards by the neighbourhood influence so that samples where neighbours have a stronger impact on decisions receive greater emphasis.

For a rollout o_i associated with node v_i, the base reward comprises two components:

$$R_i^{\text{base}} = s_{\text{format}}(o_i) + s_{\text{acc}}(o_i), \tag{12}$$

where s_{format} enforces adherence to the output schema (e.g., correct use of `<think>` and `<answer>` tags), and s_{acc} verifies whether the final answer matches the ground-truth identifier token.

To reflect the importance of the neighbourhood via the margin gain from Eq. (11), we define a reshaping factor:

$$g_i = \exp(\alpha \cdot |\Delta_i|), \tag{13}$$

where α is a temperature hyperparameter controlling sensitivity.

This exponential form has two intuitive effects: (i) when $|\Delta_i| = 0$, $g_i = 1$ and the reward remains unchanged; (ii) larger $|\Delta_i|$ values exponentially amplify the reward, encouraging the model to focus more on neighbour-influenced samples during policy optimisation. The final reward is therefore:

$$R_i = g_i \cdot R_i^{\text{base}} = \exp(\alpha|\Delta_i|)\big(s_{\text{format}}(o_i) + s_{\text{acc}}(o_i)\big). \tag{14}$$

Incorporating this reward design into LLM update using objective in Eq. (2) will emphasise structurally informative neighbourhoods, guiding the LLM generation to more effectively leverage relational context for reasoning on text-rich networks.

3.4 Inference on Edge and Graph Tasks

Although TRN-R1-Zero is trained only on node-level tasks, extending it to other TRN tasks such as link prediction and graph reasoning is straightforward. The input prompt requires only the sampled graph with neighbour information, and the task instruction. Detailed prompt designs are provided in Appendix B, with cross-task experiments reported in Section 4.3.

4 Experiments

4.1 Setup

Datasets. Experiments are conducted on nine datasets spanning four relational structures (citation, hyperlink, social, and co-purchase) and three task types (node-, graph-, and edge-level). For RL training, **Citeseer** and **History** are used to capture citation and co-purchase relations. The remaining **Cora**, **Photo**, **WikiCS**, and **Instagram** are used for zero-shot in-domain and cross-domain evaluation, while **Expla-Graph** (He et al., 2024b), **WikiCS-Link**, and **Instagram-Link** are used for cross-task evaluation. All datasets (detailed in Table 7 and 8) are sourced from NodeBed (Wu et al., 2025a).

- **Cora, Citeseer**: Each node represents a scientific publication including the paper title and abstract. Edges denote citation links between papers, forming a citation network.
- **WikiCS**: Each node corresponds to a Wikipedia article, and edges represent hyperlinks between articles, forming a web graph.
- **Instagram**: Each node represents a user account, and edges correspond to social-follow relations. Node texts are profile descriptions or short post contents, reflecting social interaction context.
- **Photo, History**: Each node corresponds to a product on the Amazon platform. Nodes are customer reviews in Photo and product descriptions in History, and edges capture co-purchase relations between products.
- **Expla-Graph**: Each node denotes a commonsense concept, and each edge denotes the relation between two concepts.
- **WikiCS-Link and Instagram-Link**: Both datasets are constructed from original node-level datasets by retaining the original edges

Type	Method	Cora		WikiCS		Instagram		Photo		Avg.	
		Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
LLM	GPT-4o	70.30	**71.44**	69.69	64.51	42.42	39.79	**69.93**	**68.55**	63.09	61.07
	Llama-3.1-8B	64.55	64.41	59.43	54.16	36.98	28.32	45.49	50.44	51.61	49.33
	Qwen2.5-1.5B-it	47.96	49.91	61.71	56.17	36.82	28.37	50.72	51.50	49.30	46.49
	Qwen2.5-7B-it	67.59	67.19	67.44	63.93	52.20	50.32	55.67	59.37	60.73	60.20
	Qwen2.5-14B-it	67.22	68.26	73.03	**70.78**	**55.60**	**52.94**	58.51	61.45	63.59	63.36
GFM	ZeroG	62.55	57.56	62.71	57.87	50.71	50.43	46.27	51.52	55.56	54.35
	LLaGA	18.82	8.49	8.20	8.29	47.93	47.70	39.18	4.71	28.53	17.30
SFT + RL	Graph-R1 (14B)	68.15	67.34	73.25	70.11	52.03	52.06	-	-	-	-
RL Only	**TRN-R1-Zero (7B)**	**72.59**	70.33	**73.63**	70.30	54.76	52.54	65.12	64.22	**66.53**	**64.35**

Table 2: Performance comparison under the zero-shot setting with Accuracy (%) and Macro-F1 (%) reported with benchmarks following (Wu et al., 2025a). The **best** and second-best results are highlighted per column (paired t-test, $p \leq 0.05$, Bonferroni corrected). Graph-R1 is excluded from the Photo dataset since Photo was used in its pre-training, not qualifying for zero-shot evaluation.

as positives and uniformly sampling an equal number of non-existent edges as negatives.

Baselines. The comparison includes three categories of baselines:

- **LLMs**: GPT-4o is included to represent a potential upper bound of LLM performance. LLaMA-3.1-8B, Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, and Qwen2.5-14B-Instruct are selected to cover diverse open-source model families and scales.
- **Graph Foundation Models (GFMs)**: ZeroG (Li et al., 2024a) and LLaGA (Chen et al., 2024b) are evaluated in an intra-domain manner, where each model is pre-trained on the same domain dataset (e.g., arXiv for academic data) before being tested on the target dataset.
- **Reasoning LLMs**: Graph-R1 (Wu et al., 2025b) introduces a *rethink* template that encourages LLMs to reason carefully and revise their responses before producing the final answer. In the original setup, DeepSeek-v3 is used to summarise node texts into compact representations. For fairness, the following experiments use raw node texts directly.

Implementations. The dataset statistics are summarised in Table 8, covering four relation types: citation, co-purchase, hyperlink, and social. During training, **Citeseer** (citation domain) and **History** (co-purchase domain) are used to fine-tune the base LLM, enabling it to capture the semantic characteristics of two distinct relational types and to learn reasoning under relational constraints. To ensure that evaluation reflects genuine cross-domain and cross-relation generalisation, datasets from the hyperlink and social domains are deliberately excluded from training. All datasets are randomly split into 60%/20%/20% for training, validation, and testing, respectively. Prompt templates for generative LLMs are listed in Table 3.2. Qwen2.5-7B-Instruct serves as the base model for TRN-R1-Zero. Low-Rank Adaptation (LoRA (Hu et al., 2022)) is employed for memory-efficient fine-tuning, with the rank set to 64. All experiments are conducted on a single AMD MI300X GPU.

For the margin-gain computation, the SGC aggregator in Eq. (8) is applied with K=1. The temperature in the reshaping factor of Eq. (13) is set to α=10, amplifying the reward contribution of samples whose classification margin shifts substantially under neighbourhood aggregation.

4.2 Overall Results

The overall zero-shot node classification results are presented in Table 2. **TRN-R1-Zero attains the highest average Accuracy and Macro-F1 across all datasets**, validating its effectiveness and superior generalisation across domains.

LLaGA exhibits noticeably lower performance, indicating limited domain transferability, as its mapping layer to align graph embeddings with LLM embeddings is trained on a source graph and struggles to generalise to unseen graphs. In contrast, ZeroG achieves competitive results, since its post-encoding information aggregation does not compromise generalisation ability.

Among pure LLMs, GPT-4o achieves the best performance on Cora and Photo, whereas Qwen2.5-14B-Instruct surpasses GPT-4o on WikiCS and Instagram. For models of comparable scale, Qwen2.5-7B-Instruct consistently outperforms LLaMA-3.1-8B across all datasets. Al-

though the smaller Qwen2.5-1.5B-Instruct exceeds LLaMA-3.1-8B on WikiCS and Photo, it falls behind on Cora. These results suggest that both larger model capacity and instruction tuning contribute positively to zero-shot relational reasoning.

For reasoning-based LLMs, TRN-R1-Zero not only achieves the best or near-best results overall but also demonstrates strong generalisation capability. Despite being trained only on the Citeseer and History datasets and without any exposure to test graphs, TRN-R1-Zero performs well on both in-domain datasets (Cora, Photo) and out-of-domain datasets (WikiCS, Instagram).

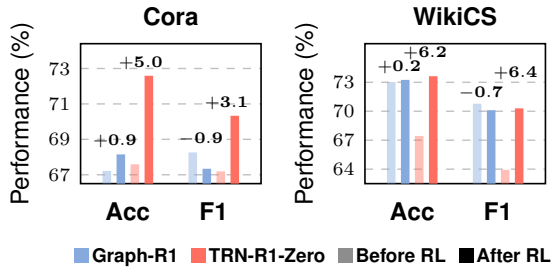


Figure 3: Performance comparison with RL training between our TRN-R1-Zero (red) and Graph-R1 (blue).

4.3 Generalisation to Graph and Edge Tasks

Although TRN-R1-Zero is trained only on node classification, its zero-shot ability is further examined on two unseen tasks across Expla-Graph, WikiCS-Link and Instagram-Link. For graph-level reasoning, TRN-R1-Zero improves over the base model at both 7B and 14B scales, and under the same Qwen2.5-14B backbone, it also surpasses Graph-R1, even though Graph-R1 is explicitly trained on graph-level tasks. For edge-level prediction, the gains here are particularly substantial (e.g., +16.10 on WikiCS-Link for 7B), indicating effective transfer of relational reasoning, and again TRN-R1-Zero exceeds Graph-R1 under the same 14B backbone despite never being trained on edge-level supervision. **Together, these results highlight the zero-shot ability of TRN-R1-Zero on graph and edge level tasks, derived from the effective neighbour-aware reasoning training only on node level tasks.**

4.4 Effectiveness of Neighbour-aware RL

This experiment investigates how our proposed neighbour-aware RL post-training can enhance the zero-shot node classification capability of LLMs. The comparison is between using the vanilla CoT distillation in Graph-R1 (14B) and our TRN-R1-Zero (7B). The evaluation is conducted on the Cora and WikiCS datasets, which represent the citation and hyperlink domains, using accuracy and macro-F1 as the primary metrics.

Scale	Model	Expla-Graph	WikiCS-Link	Insta-Link
7B	Qwen2.5	84.12	52.10	64.90
	TRN-R1-ZERO	87.18 +3.06	68.20 +16.10	66.80 +1.90
14B	Qwen2.5	89.89	72.10	71.80
	Graph-R1	89.71	48.90	56.40
	TRN-R1-ZERO	90.25 +0.36	73.90 +1.80	74.20 +2.40

Table 3: Zero-shot performance on graph reasoning and link prediction. Trained only on node level tasks.

Figure 3 shows the performance gains achieved through RL training on the Cora and WikiCS datasets. TRN-R1-Zero consistently yields larger improvements in both Accuracy and F1, whereas Graph-R1 even experiences a decline in F1 across both datasets. These results indicate that the neighbour-aware reward design effectively stabilises the optimisation process and promotes more balanced metric improvements. **TRN-R1-Zero not only delivers consistent and robust performance gains through reinforcement learning, but also exhibits superior optimisation stability and generalisation** compared with baselines.

4.5 Effectiveness of Margin Gain

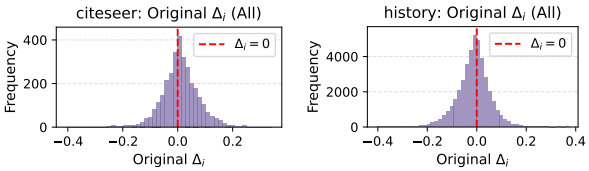
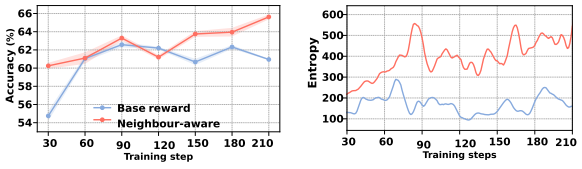


Figure 4: Original margin gain values Δ_i across the training datasets (Citeseer and History). These results demonstrate the distribution of impact from neighbour information towards the target node, motivating the neighbour-aware reward design.

The margin gain visualisations in Figure 4 provide an intuitive view of how neighbour aggregation influences decision confidence during RL training. Specifically, a positive Δ_i indicates that neighbour aggregation shifts the target embedding closer to the ground-truth label embedding, whereas a negative Δ_i indicates the opposite effect. A larger $|\Delta_i|$ therefore reflects a stronger influence of neighbour information on the classification of the target node, and such high-impact samples are the ones most worth emphasising during policy optimisation.

To examine the effect of the neighbour-aware policy objective on training dynamics, the Qwen2.5-1.5B-Instruct model is used as the base policy model for computational efficiency. Two reward variants are compared: (i) the base reward without scaling and (ii) the $\exp(|\Delta_i|)$-scaled reward. The Cora dataset is used for evaluation. Figure 5a illustrates the average accuracy across training steps under both settings. The results show that incorporating neighbour-aware reward shaping stabilises optimisation and yields more consistent performance improvements compared with the unshaped baseline. Each checkpoint model is evaluated five times on the Cora dataset to ensure robustness. The training statistics in Figure 5b, c, and d further support the effectiveness of neighbour-aware shaping: the entropy remains relatively high, encouraging the policy model to explore more diverse and optimised responses rather than over-exploit early patterns; meanwhile, the average response length steadily increases, suggesting that the model performs deeper reasoning. Additionally, the rising frequency of the word "neighbour" indicates that the model gradually learns to leverage relational context more effectively. Our **neighbour-aware margin gain enhances both the stability and utilisation of neighbourhood in reasoning depth** for node classification.



(a) Cora Evaluation (b) Entropy

(c) Response Length (d) #Neighbour Word

Figure 5: Accuracy comparison between base reward and neighbour-aware reward across Cora dataset. Neighbour-aware shaping consistently improves both optimisation stability and reasoning depth.

4.6 Impact of Different LLM Backbones

To assess the generality of TRN-R1-ZERO across LLM backbones, models spanning different families and scales are trained, including Llama-3.2-3B, Llama-3.1-8B, and Qwen2.5-14B. Across all families and scales, TRN-R1-ZERO consistently improves zero-shot node classification, with the largest gains observed on smaller backbones (e.g., +14.4 and +9.0 in average accuracy on the 3B and 8B models, respectively). These results indicate that the **proposed training paradigm is not tied to a specific backbone and generalises across architectures and scales.**

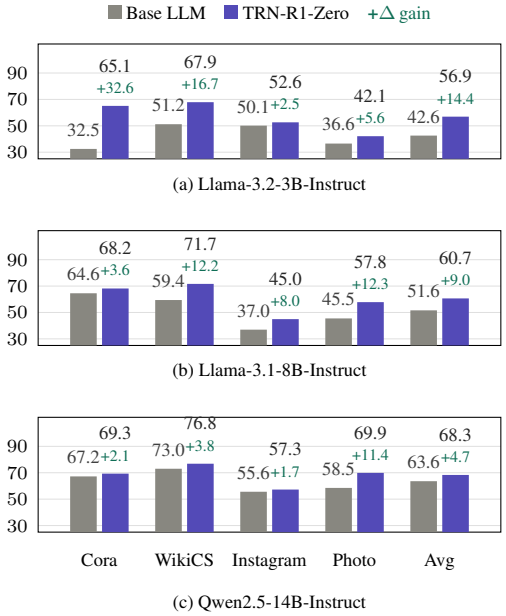


Figure 6: Zero-shot node classification accuracy across three model scales. Green annotations show absolute gain over the base LLM.

4.7 Performance under Supervised Settings

In the supervised setting, 60% of each dataset is used for training and 20% for testing. This configuration follows the common practice in GNN and GNN-adapter frameworks (e.g., LLaGA) for LLMs. Since direct supervision signals are available from the training-testing split within the same dataset, traditional supervised models such as Graph Convolutional Networks (GCNs) can perform effectively. As shown in Table 4, TRN-R1-Zero still outperforms both GCN and LLaGA under the supervised setting, demonstrating strong optimisation and reasoning capabilities even when explicit labels are provided. **Overall, TRN-R1-Zero maintains superior performance and stable optimisation behaviour under supervised learning conditions.**

4.8 Case Study and Efficiency

Compared with other reasoning-based methods, **TRN-R1-Zero enjoys a much higher efficiency**

Model	Citeseer	History
GCN	76.45	84.23
LLaGA	76.73	85.56
TRN-R1-Zero	**77.74**	**86.71**

Table 4: Comparison of TRN-R1-Zero with GCN (Kipf and Welling, 2017) and LLaGA under the supervised setting on the Citeseer and History datasets.

by (1) employing a substantially smaller model; and (2) significantly shortens the average response length, producing more concise yet effective reasoning traces while achieving superior performance (see Box2 and Box3). This improvement not only reduces GPU memory consumption but also accelerates inference, demonstrating that reinforcement learning alone can promote efficient reasoning behaviour without relying on large-scale supervised fine-tuning or external LLMs.

Dataset	Model	Size	Resp. Len.	Acc.
Cora	TRN-R1-Zero	7B	152.57	72.59
Cora	Graph-R1	14B	930.47	68.15
WikiCS	TRN-R1-Zero	7B	129.27	73.63
WikiCS	Graph-R1	14B	1129.22	73.25

Table 5: Comparison of model size, average response length (Resp. Len.), and accuracy (Acc.) between TRN-R1-Zero and Graph-R1 on Cora and WikiCS.

Box 2: Response of TRN-R1-Zero

Response for Node 11 in Cora by TRN-R1-Zero

<think> The target node discusses the MIN-FEATURES bias in machine learning, which is related to the selection of relevant features. It talks about learning algorithms, sample complexity, and generalization performance. The focus is on understanding and improving the learning process by considering the relevance of features, which is a theoretical aspect of machine learning. Looking at the neighbor nodes: - One discusses maximizing coverage in learning algorithms. - Another focuses on prototype and feature selection using sampling and random mutation hill climbing. - The third uses compression-based methods for feature subset selection. All these topics are closely related to feature selection and learning algorithms, which fall under the broader category of machine learning theory. Therefore, the most appropriate category for the target node would be "Theory" (category 4). </think> <answer> 4 ✓ </answer>

Box 3: Response of Graph-R1

Response for Node 11 in Cora by Graph-R1

Okay, let's try to figure out how to classify the paper segment represented by node0. ...**38 words**... First, I need to understand what each node is about. Node0's paper is about the MIN-FEATURES bias in machine learning, focusing on feature selection. ...**35 words**... Looking at the connections: node0 is connected to node1, node2, and node3. Node1 talks about feature subset selection using Information Theory. Node2 discusses MDL (Minimum Description Length) for feature selection. Node3 is about constructive induction and feature selection with a non-greedy strategy. ...**334 words**... Given the options, "Rule_Learning" (0) seems the most fitting because it directly relates to the algorithms discussed (ID3, FOCUS) and the context of feature selection in decision trees. "Theory" is a possible contender but might be too broad. The other categories don't align as well. Therefore, the best fit is Rule_Learning (0). </think> <structure> The subgraph forms a star topology with node0 at the center connected to nodes1,2,3. ...**43 words**... </structure> <semantic> ...**40 words**..."Theory" (category4) is plausible due to PAC-learning and Information Theory, but the emphasis on practical algorithms (ID3, FOCUS) leans more toward applied methods. "Probabilistic_Methods" (6) is less fitting as MDL is information-theoretic, not probabilistic. Other categories (Neural_Networks, Case_Based, Genetic_Algorithms) are irrelevant. </semantic> <comprehensive> Candidate answers: 1. **Rule_Learning (0)**: Directly supported by ID3/FOCUS comparisons, feature selection in decision trees, and algorithmic focus. 2. **Theory (4)**: PAC-learning bounds and Information Theory overlap, but less emphasis on theoretical frameworks compared to applied algorithms. 3. **Probabilistic_Methods (6)**: MDL relates to information theory but lacks explicit probabilistic modeling. Rethinking: - **Rule_Learning** best captures the practical algorithm comparisons (ID3, FOCUS) and feature selection in decision trees. - **Theory** is secondary but insufficient due to the applied focus of node0. - Other categories are invalid based on content exclusion. Final Answer: 0 (Rule_Learning) ✗

5 Conclusion

This paper introduces TRN-R1-Zero, a reinforcement learning-only framework for text-rich node reasoning. A neighbour-aware group relative policy optimisation objective models the impact of neighbouring information during training with a novel margin gain. Without distillation or supervised fine-tuning, TRN-R1-Zero directly optimises a base LLM for strong zero-shot reasoning. Extensive experiments verify the effectiveness of TRN-R1-Zero.

Limitations

For zero-shot text-rich network (TRN) tasks, LLMs must not only extract useful information from the target node text, neighbour node texts, and candidate label texts, but also comprehend the underlying semantics of these texts. Therefore, if the base LLM lacks sufficient domain knowledge, reinforcement learning may offer limited improvements, as such knowledge primarily originates from the pretraining phase.

Acknowledgments

This research has been supported by Australian Research Council Discovery Projects (CE200100025, DP230101196 and DE250100919).

References

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovic, and Martin T. Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions. *CoRR*, abs/2505.23281.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *COLM*.

Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. 2024a. Graphwiz: An instruction-following language model for graph computational problems. In *KDD*.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. 2024b. Llaga: Large language and graph assistant. In *ICML*.

Zhaoling Chen, Robert Tang, Gangda Deng, Fang Wu, Jialong Wu, Zhiwei Jiang, Viktor K. Prasanna, Arman Cohan, and Xingyao Wang. 2025. Locagent: Graph-guided LLM agents for code localization. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Yi Fang, Dongzhe Fan, Sirui Ding, Ninghao Liu, and Qiaoyu Tan. 2025. Uniglm: Training one unified language model for text-attributed graphs embedding. In *WSDM*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *ICLR*.

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 *ICLR*.

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 *NeurIPS*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Hanxu Hu, Hongyuan Lu, Huajian Zhang, Yun-Ze Song, Wai Lam, and Yue Zhang. 2024. Chain-of-symbol prompting for spatial reasoning in large language models. In *COLM*.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. Can GNN be good adapter for llms? In *WWW*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can language models resolve real-world github issues? In *ICLR*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

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 *ICLR*.

Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. 2024a. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *KDD*.

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 *NeurIPS*.

Yilun Liu, Ruihong Qiu, and Zi Huang. 2023. Cat: Balanced continual graph learning with graph condensation. In *ICDM*.

Yilun Liu, Ruihong Qiu, and Zi Huang. 2025a. Gcondenser: Benchmarking graph condensation. In *CIKM*.

Yilun Liu, Ruihong Qiu, Yanran Tang, Hongzhi Yin, and Zi Huang. 2025b. PUMA: efficient continual graph learning for node classification with graph condensation. *TKDE*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025c. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024a. Graphgpt: Graph instruction tuning for large language models. In *SIGIR*.

Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2024b. Casegnn: Graph neural networks for legal case retrieval with text-attributed graphs. In *ECIR*.

Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2026. LEXA: legal case retrieval via graph contrastive learning with contextualised LLM embeddings. *World Wide Web (WWW)*, 29(2):20.

Yanran Tang, Ruihong Qiu, Hongzhi Yin, Xue Li, and Zi Huang. 2024c. Caselink: Inductive graph learning for legal case retrieval. In *SIGIR*.

Danny Wang, Ruihong Qiu, Guangdong Bai, and Zi Huang. 2025a. Text meets topology: Rethinking out-of-distribution detection in text-rich networks. In *EMNLP*.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *NeurIPS*.

Haoyu Wang, Shikun Liu, Rongzhe Wei, and Pan Li. 2025b. Model generalization on text attribute graphs: Principles with large language models. In *ICML*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? In *NeurIPS*.

Yuyao Wang, Bowen Liu, Jianheng Tang, Nuo Chen, Yuhan Li, Qifan Zhang, and Jia Li. 2025c. Graph-r1: Unleashing LLM reasoning with np-hard graph problems. *CoRR*, abs/2508.20373.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying graph convolutional networks. In *ICML*.

Xixi Wu, Yifei Shen, Fangzhou Ge, Caihua Shan, Yizhu Jiao, Xiangguo Sun, and Hong Cheng. 2025a. When do LLMs help with node classification? a comprehensive analysis. In *ICML*.

Xixi Wu, Yifei Shen, Caihua Shan, Kaitao Song, Siwei Wang, Bohang Zhang, Jiarui Feng, Hong Cheng, Wei Chen, Yun Xiong, and Dongsheng Li. 2024. Can graph learning improve planning in llm-based agents? In *NeurIPS*.

Yicong Wu, Guangyue Lu, Yuan Zuo, Huarong Zhang, and Junjie Wu. 2025b. Graph-r1: Incentivizing the zero-shot graph learning capability in llms via explicit reasoning. In *EMNLP*.

Li Zhong, Zilong Wang, and Jingbo Shang. 2024. Debug like a human: A large language model debugger via verifying runtime execution step by step. In *ACL*.

A Text Quality for Current Text-rich Networks

Textual nodes in text-rich networks often meet noisy or incomplete text. It is hard for a LLM directly give a reasonable label for such nodes. Examples for Cora and Photo datasets are shown in Table 6.

Cora	TABLE DES MATI ERES 1 Apprentissage et approximation les techniques de regularisation 3 1.1 Introduction:
Photo	Good product, good price good price without shipping fee. With shipping fee, it is still a good deal.

Table 6: Examples of meaningless and noisy texts in TRNs.

B Prompt Design for Edge-level and Graph-level Tasks

The following prompts are used to evaluate TRN-R1-Zero for edge-level and graph-level tasks. The differences between them are the graph prompt and the task instruction parts.

Edge-level Prompt for TRN-R1-Zero

```
# System Prompt
You are a helpful AI Assistant that provides well-reasoned and detailed responses. You first think about the reasoning process as an internal monologue and then provide the user with the answer. Respond in the following format:
<think>
...
</think>
<answer>
...
</answer>

# Graph Prompt
Source node: {source_node}
Target node: {target_node}
Neighbours of source node: {source_neighbors}
Neighbours of target node: {target_neighbors}

# Task Instruction
Your task is to predict whether a link exists between two nodes in a graph. Each node represents a {node_type}. The relation type in this graph is {relation}.
Question: Based on the attributes and neighbourhood structure of both nodes, predict whether a {relation} link exists between the source and target nodes. Answer with 0 (no link) or 1 (link exists).
```

Graph-level Prompt for TRN-R1-Zero

```
# System Prompt
You are a helpful AI Assistant that provides well-reasoned and detailed responses. You first think about the reasoning process as an internal monologue and then provide the user with the answer. Respond in the following format:
<think>
...
</think>
<answer>
...
</answer>

# Graph Prompt
Nodes: {node_list}
Relationships: {edge_list}

# Task Instruction
Your task is to determine if two arguments support or counter each other, based on the provided commonsense graph. The commonsense graph is defined by nodes and their relationships.
Based on this graph, consider the following: {question}.
Your answer must be a single integer ID, where 0 means support and 1 means counter.
```

C Extended Dataset Statistics

The following tables include the extended dataset statistics (Table 8) with detailed description of meta data like label in text for each dataset (Table 7).

Relation	Dataset	Node Type	Labels
Citation	Cora	Paper segment	Rule_Learning; Neural_Networks; Case_Based; Genetic_Algorithms; Theory; Reinforcement_Learning; Probabilistic_Methods
	Citeseer	Paper segment	Agents; ML (Machine Learning); IR (Information Retrieval); DB (Databases); HCI (Human-Computer Interaction); AI (Artificial Intelligence)
Hyperlink	WikiCS	Wikipedia article	Computational Linguistics; Databases; Operating Systems; Computer Architecture; Computer Security; Internet Protocols; Computer File Systems; Distributed Computing Architecture; Web Technology; Programming Language Topics
Social	Instagram	Instagram User Bio	Normal Users; Commercial Users
Co-purchase	History	Product description	World; Americas; Asia; Military; Europe; Russia; Africa; Ancient Civilizations; Middle East; Historical Study & Educational Resources; Australia & Oceania; Arctic & Antarctica
	Photo	Customer review	Video Surveillance; Accessories; Binoculars & Scopes; Video; Lighting & Studio; Bags & Cases; Tripods & Monopods; Flashes; Digital Cameras; Film Photography; Lenses; Underwater Photography
Commonsense	Expla-Graph	Commonsense concept	Support; Counter

Table 7: Meta information for benchmark TRNs grouped by relation type.

Domain	Dataset	#Nodes	#Graphs	#Edges	Avg. Deg.	Homo.	#Classes
Citation	Citeseer*	3,186	1	8,450	2.65	0.72	6
	Cora	2,708	1	5,429	3.90	0.83	7
Webpage	WikiCS	11,701	1	431,206	36.85	0.68	10
Social	Instagram	11,339	1	144,010	12.70	0.59	2
Co-purchase	Photo	48,362	1	873,782	18.07	0.79	12
	History*	41,551	1	503,180	12.11	0.78	12
Commonsense	Expla-Graph	5.2	2,766	4.3	-	-	2

Table 8: Statistics of benchmark datasets. Datasets marked with * (*Citeseer* and *History*) are used for RL training, while the others are held out for evaluation and generalisation studies.