LLM as Prompter: Low-resource Inductive Reasoning on Arbitrary Knowledge Graphs

Anonymous ACL submission

Abstract

Knowledge Graph (KG) inductive reasoning, which aims to infer missing facts from new KGs that are not seen during training, has been widely adopted in various applications. One critical challenge of KG inductive reasoning is handling low-resource scenarios with scarcity in both textual and structural aspects. In this paper, we attempt to address this challenge with Large Language Models (LLMs). Particularly, we utilize the state-of-the-art LLMs to generate a graph-structural prompt to enhance the pre-trained Graph Neural Networks (GNNs), which brings us new methodological insights into the KG inductive reasoning methods, as well as high generalizability in practice. On the methodological side, we introduce a novel pretraining and prompting framework PROLINK, designed for low-resource inductive reasoning across arbitrary KGs without requiring additional training. On the practical side, we experimentally evaluate our approach on 36 low-resource KG datasets and find that PROLINK outperforms previous methods in three-shot, one-shot, and zero-shot reasoning tasks, exhibiting average performance improvements by 20%, 45%, and 147%, respectively. Furthermore, PROLINK demonstrates strong robustness for various LLM promptings as well as full-shot scenarios. The anonymous version of our source code is available on `https://anonymous.4open.science/r/ProLINK-069D`.

1 Introduction

Knowledge Graph (KG) Reasoning, also known as KG Link Prediction, aims at inferring new facts from existing KGs in the triple format (head entity, relation, tail entity)(Wang et al., 2021b; Li et al., 2023). This technique has been widely studied and applied in various domains, including information retrieval, e-commerce recommendations, drug discovery, and financial prediction (Deng et al., 2019; Zeng et al., 2020; Bonner et al., 2022; Ge et al., 2023). For example, the query (q_1) in Figure 1 consists of '*Entity 3*' and a relation type '*occupation*'. Its expected answer for KG reasoning is '*Entity 6*'.

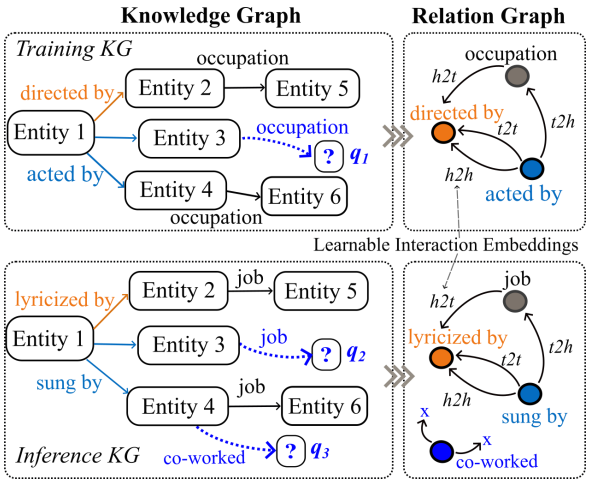


Figure 1: Illustrations of knowledge graphs and corresponding relation graphs. The interaction edge $h2t$ means the source relation has a head entity in the KG which is a tail entity of the target relation.

The dynamic nature of real-world KGs fosters recent research interest in *inductive KG reasoning*—inferring from new KGs which have entities/relations unseen during training (Sadeghian et al., 2019; Teru et al., 2020). In contrast to earlier methods that learn graph-specific embeddings for the training KG (Bordes et al., 2013; Yang et al., 2015; Sun et al., 2019), recent inductive reasoning models are trained to address the query (q_1) in the training KG, enabling them to answer the query (q_2) in a fully new KG (Lee et al., 2023; Galkin et al., 2023). As illustrated in Figure 1, this transferability stems from Graph Neural Networks (GNNs) (Kipf and Welling, 2017) which capture the shared interaction patterns in relation graphs without using entity/relation textual information.

One critical challenge of KG inductive reasoning is *handling low-resource scenarios marked by scarcity of both textual and structural information*. For the query (q_3) in Figure 1, the inference KG

lacks textual context for entities and has no support triple for the '*co-worked*' relation. Such low-resource scenarios frequently occur in specialized or emerging domains due to long-tail distribution and new relation additions (Pei et al., 2023; Wu et al., 2023), hindering the wide adoption of KG inductive reasoning. Text-based methods, even the current powerful Large Language Models (LLMs) (Touvron et al., 2023; OpenAI, 2023), are constrained by limited textual data and complicated graph structures, while graph-based methods struggle with few-shot relation types that lack enough interaction edges in the relation graph[1].

Human reasoning, however, may address the above low-resource queries without expert knowledge or prior learning. Simply leveraging limited relation semantics to understand graph structures, humans can successfully deduce that the '*co-worked*' relation should connect two persons, suggesting the answer entity can be a head entity of the '*job*' relation and near the query entity '*Entity 4*'. Inspired by the human reasoning process which mostly relies on relation semantics, we pose a question: *Can LLMs be used to emulate human reasoning for relation semantic understanding, thereby enhancing the GNN-based inductive reasoning?*

This work gives a positive answer to the above question and presents an approach that leverages LLM's basic language power to improve KG inductive reasoning in any low-resource scenarios without additional model training for new KGs. This ability is crucial for elevating the generalizability of AI technologies to handle data dynamicity in the real world. Specifically, we propose a novel pretraining and **Pro**mpting framework PROLINK for **L**ow-resource **IN**ductive reasoning across arbitrary **K**Gs. First, PROLINK pretrains a GNN-based KG reasoner with novel techniques enhancing few-shot prediction performance. Then, given a new inference KG with sparse relation types, PROLINK employs a pre-trained LLM to construct a prompt graph through concise relation descriptions (a dozen words per relation). The prompt graph is calibrated to eliminate noise and subsequently injected into the relation graph of the KG to improve the performance of the GNN reasoner. In summary, this work has four novel contributions:

- We introduce a new KG reasoning problem, i.e., low-resource inductive reasoning on KGs with arbitrary entity and relation vocabularies, which generalizes most KG link prediction tasks.
- To the best of our knowledge, this is the first work leveraging LLMs as the graph prompter for inductive KG reasoning, potentially inspiring further innovation in the research community.
- We design a unique *pretraining and prompting* framework PROLINK, containing several novel techniques for low-resource inductive reasoning and prompt graph generation.
- We construct 36 low-resource inductive datasets from real-world KGs, in which PROLINK outperforms previous state-of-the-art methods in both few-shot and zero-shot reasoning tasks.

[1]We verified this challenge in preliminaries in Section 2.3.

2 Background

2.1 Notations and Definitions

Let $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ denote a Knowledge Graph (KG), where $\mathcal{E}, \mathcal{R}$ are the sets of entities and relations. $\mathcal{T} = \{(e_h, r, e_t)|e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$ is the set of factual triples of the KG, where e_h, r and e_t are called a triple's head entity, relation and tail entity, respectively. Given a query (e_q, r_q) containing a query entity $e_q \in \mathcal{E}$ and a query relation $r_q \in \mathcal{R}$, the KG reasoning task aims to identify the correct entity $e_a \in \mathcal{E}$, such that the triple (e_q, r_q, e_a) or (e_a, r_q, e_q) is a valid factual triple of $\mathcal{G}$.

KG Inductive Reasoning: Given a model trained on a knowledge graph $\mathcal{G}_{tr} = \{\mathcal{E}_{tr}, \mathcal{R}_{tr}, \mathcal{T}_{tr}\}$, the KG reasoning task in the fully inductive scenario evaluates the trained model on a new inference graph $\mathcal{G}_{inf} = \{\mathcal{E}_{inf}, \mathcal{R}_{inf}, \mathcal{T}_{inf}\}$, in which all entities and relations are different from those in $\mathcal{G}_{tr}$, i.e., $\mathcal{E}_{inf} \cap \mathcal{E}_{tr} = \emptyset$ and $\mathcal{R}_{inf} \cap \mathcal{R}_{tr} = \emptyset$.

K-shot Inductive Reasoning: In the inductive scenario with $\mathcal{G}_{inf} = \{\mathcal{E}_{inf}, \mathcal{R}_{inf}, \mathcal{T}_{inf}\}$, given a new relation type $r_q \notin \mathcal{R}_{inf}$ and support triples $\mathcal{T}_{r_q} = \{(e_h, r_q, e_t)|e_h, e_t \in \mathcal{E}_{inf}\}(|\mathcal{T}_{r_q}| = K)$, the K-shot reasoning task is to predict over a query set $\{(e_q, r_q)|e_q \in \mathcal{E}_{inf}\}$ with the augmented KG $\mathcal{G}^{r_q}_{inf} = \{\mathcal{E}_{inf}, \mathcal{R}_{inf} \cup \{r_q\}, \mathcal{T}_{inf} \cup \mathcal{T}_{r_q}\}$.

In this work, we focus on the low-resource challenge of KG inductive reasoning, inferring not merely unseen but also few-shot relation types. Existing methods for few-shot link prediction (Chen et al., 2019; Zhang et al., 2020; Huang et al., 2022; Wu et al., 2023) either only work on the training graph structure or have to use graph-specific textual or ontological information to train, therefore cannot achieve inductive reasoning on arbitrary KGs. Other related studies also have difficulty accomplishing this low-resource task, including entity-

level inductive reasoning (Sadeghian et al., 2019; Teru et al., 2020; Zhu et al., 2021; Zhang and Yao, 2022) and text-based inductive reasoning (Daza et al., 2021; Wang et al., 2021c; Markowitz et al., 2022; Gesese et al., 2022). Detailed related work is introduced in the Appendix D.

2.2 Baseline: ULTRA

We first introduce ULTRA (Galkin et al., 2023), the state-of-the-art GNN-based model for inductive reasoning on entirely new KGs. Its core idea is leveraging the 'invariance' of the KG relational structure. With the nature of the triple form, there are four basic interaction types when two relations are connected, i.e., *tail-to-head (t2h)*, *head-to-head (h2h)*, *head-to-tail (h2t)*, and *tail-to-tail (t2t)*. As shown in Figure 1, different KGs may have similar patterns in the relation interactions. Thereby, the pre-trained embeddings of four interaction types can be universally shared across KGs to parameterize any unseen relations.

Given an inference KG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, ULTRA constructs a relation graph $\mathcal{G}_r = \{\mathcal{R}, \mathcal{R}_{fund}, \mathcal{T}_r\}$ from the original triple data $\mathcal{T}$, in which each node is a relation type and edges have four interaction types $\mathcal{R}_{fund}$. After adding inverse relations into $\mathcal{G}$ (Zhu et al., 2021; Zhang and Yao, 2022), $\mathcal{G}_r$ would contain $2|\mathcal{R}|$ nodes. Then, ULTRA employs a graph neural network $GNN_r(\cdot)$ over the relation graph $\mathcal{G}_r$, and obtains the relative representation of each relation conditioned on a query, which then can be used by any off-the-shelf entity-level GNN-based models $GNN_e(\cdot)$ for KG reasoning (Zhu et al., 2021; Zhang and Yao, 2022; Wang et al., 2023). Specifically, given a query $q = (e_q, r_q)$, the score $p(e_q, r_q, e_t)$ of one candidate entity e_t is calculated as follows:

$$\mathbf{R}_q = GNN_r(\theta_r, \mathbf{r}_q, \mathcal{G}_r), \quad (1)$$

$$\mathbf{E}_q = GNN_e(\theta_e, \mathbf{e}_q, \mathbf{R}_q, \mathcal{G}), \quad (2)$$

$$p(e_q, r_q, e_t) = MLP(\mathbf{E}_q[e_t]). \quad (3)$$

where θ_r, θ_e denote the parameters of two GNN modules, and $\mathbf{e}_q, \mathbf{r}_q$ are initialized embedding vectors of e_q and r_q[2]. Because ULTRA does not require any input features of entities or relations nor learn graph-specific entity or relation embeddings, it enables inductive reasoning across arbitrary KGs.

[2]In the implementation, they are initialized as all-one vectors whereas other nodes in the graph are initialized with zeros, which is verified generalizing better to unseen graphs.

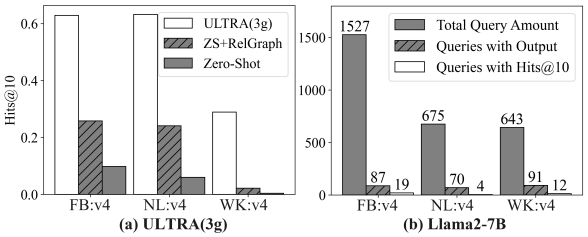


Figure 2: Preliminary results on ULTRA and Llama2.

2.3 Preliminaries: Low-resource Challenge

To explore the low-resource challenge, we verify the performance of the GNN-based ULTRA (Galkin et al., 2023) and the LLM-based Llama2 (Touvron et al., 2023) in zero-shot inductive reasoning. On three real-world KG benchmarks, the zero-shot condition is created by removing all r_q-involved triples in $\mathcal{T}_{inf}$ for each query relation r_q. The preliminary results of the two pre-trained models are shown in Figure 2.

In Figure 2(a), we observe that the zero-shot performance of ULTRA(3g) drops sharply compared to its original version. It indicates the inductive ability of ULTRA highly relies on sufficient support triples in the inference graph. Surprisingly, ULTRA obtains better performance after adding the complete relation graph built on the original $\mathcal{G}_{inf}$, which motivates us to enhance the relation graph in low-resource scenarios. Following recent work (Ye et al., 2023), we further evaluate the zero-shot graph reasoning performance of Llama2-7B by converting queries and KG subgraphs into textual questions[3]. Due to the context window limitations of Llama2, we select hundreds of queries whose answers are in the 2-hop neighborhood subgraph of the query entity. Even though, the results are not promising. In Figure 2(b), aside from issues related to excessive length or incorrect formatting, only dozens of requests obtain standard outputs and even fewer include the correct answer in the top ten entities outputted ('Hits10').

In summary, the above preliminary results expose the challenges faced by existing methods in low-resource scenarios for KG inductive reasoning, highlighting the necessity for innovative solutions to more effectively utilize sparse data.

3 Methodology

This work presents a novel pre-training and prompting framework, PROLINK, which employs pre-trained language models (LLMs) and graph neural networks (GNNs) to handle semantic and struc-

[3]For a fair comparison, both our methods and here use only short relation context instead of entity context.

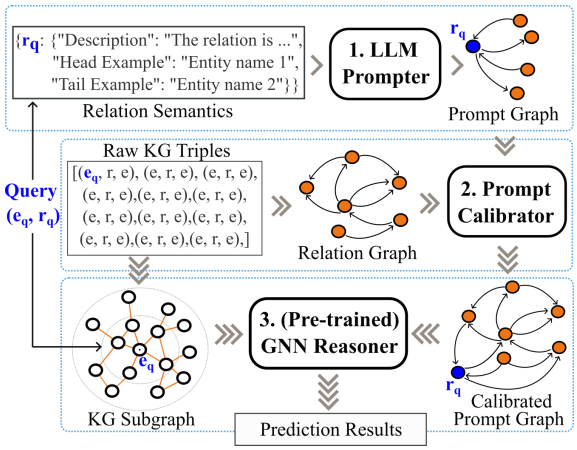


Figure 3: PROLINK prompting process.

tural information, respectively. As illustrated in Fig. 3, for a few-shot query relation in a new inferring KG, the pre-trained *GNN Reasoner* (Section 3.1) infers from KG subgraphs without model fine-tuning, guided by a relation-specific prompt graph. The prompt graph is constructed by a frozen *LLM Prompter* (Section 3.2) from relation semantics, and then calibrated by *Prompt Calibrator* (Section 3.3) to mitigate noise.

3.1 GNN-based Reasoner

Our GNN-based reasoner follows the basic framework of ULTRA (Galkin et al., 2023) in Section 2.2. Given a $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, we first construct the relation graph $\mathcal{G}_r = \{\mathcal{R}, \mathcal{R}_{fund}, \mathcal{T}_r\}$, which is efficiently obtained from the original graph $\mathcal{G}$ with sparse matrix multiplications. After that, given a query $q = (e_q, r_q)$, we generate relative relation/entity representations $\mathbf{R}_q$/$\mathbf{E}_q$ from the graph structure of $\mathcal{G}_r$ and $\mathcal{G}$, respectively. Then, calculating the triple score $p(e_q, r_q, e_t)$ of any candidate entity e_t follows Equation 3. Specifically, the GNN architecture follows NBFNet (Zhu et al., 2021) with a non-parametric DistMult (Yang et al., 2015) message function and sum aggregation. The relation encoder $GNN_r(\cdot)$ utilizes randomly initialized edge embeddings for $\mathcal{R}_{fund}$. In contrast, $GNN_e(\cdot)$ initializes the embeddings of edge types using the relative relation embeddings $\mathbf{R}_q$. We suggest consulting the ULTRA paper (Galkin et al., 2023) for further details. To improve the pre-training performance on low-resource inductive reasoning, we propose two enhancing techniques:

Role-aware Relation Encoding: The embeddings generated by $GNN_r(\cdot)$ for relative relations are currently missing vital information: the specific role that each relation type assumes in the reasoning process of a query. We delineate three unique roles: *query relation*, *inverse query relation*, and *other relation*. As a relation may serve different roles across various queries, its embedding vector should reflect the nuances of its designated role. To imbue this role-aware capability, we introduce trainable role embeddings $\mathbf{R}_o \in \mathbb{R}^{[3\times d]}$ to augment each relation embedding via a two-layer MLP, formulated as: $\hat{\mathbf{r}} = \delta\Big(\boldsymbol{W}_2\delta\big(\boldsymbol{W}_1(\mathbf{R}_q[r] : \mathbf{R}_o[role_q(r)])\big)\Big)$, where $\mathbf{R}_q[r]$ is the relative relation vector of r, concatenated with its specific role vector as determined by $role_q(\cdot)$. The enhanced relation embeddings $\hat{\mathbf{R}}_q$ are then utilized in Equation 2 replacing $\mathbf{R}_q$. This component requires far fewer parameters than the original ULTRA. Therefore, efficiency issues can usually be ignored.

Low-resource Pretraining Objective: The original ULTRA is trained by minimizing the binary cross entropy loss over positive and negative triples. Due to the sufficient support triples in the training KG, the GNN reasoner would prioritize reasoning patterns related to these triples, which would not present in low-resource scenarios. To prompt GNNs to derive insights from a broader range of relations, we devise an extra pre-training task that operates on 'pseudo' low-resource KGs. Specifically, for queries whose relation is r_q, we construct a specific KG $\mathcal{G}_{tr}^{r_q}$, where support triples containing r_q or its inverse relation are randomly masked, using a hyperparameter γ to determine the masking proportion. The total loss pretraining on the original KG and masked KGs is calculated as follows:

$$\mathcal{L}_{\mathcal{G}} = -\log p_{\mathcal{G}}(e_q, r_q, e_a) - \frac{1}{n}\sum_{i=1}^{n}\log(1 - p_{\mathcal{G}}(e_q, r_q, e_i))$$
$$\mathcal{L} = \alpha\mathcal{L}_{\mathcal{G}_{tr}} + (1-\alpha)\mathcal{L}_{\mathcal{G}_{tr}^{r_q}}, \quad (4)$$

where $p_{\mathcal{G}}(e_q, r_q, e_a)$ represents the score of a positive triple within the specific KG $\mathcal{G}$. The set $\{(e_q, r_q, e_i)\}_{i=1}^{n}$ comprises negative samples, which are generated by corrupting either the head or the tail entity in the positive triple. The hyperparameter α balances two parts of loss. In practical scenarios, we manage two pre-training tasks through controlled batch sampling. With a probability of $(1\text{-}\alpha)$, we sample batches with identical query relations for low-resource pretraining, and otherwise, we sample regular batches for pretraining on the original KG.

3.2 LLM-based Prompter

To improve the GNN reasoner's accuracy in low-resource scenarios, we create a prompt graph $\mathcal{G}_p$

connecting few-shot relations with others according to semantic features, thereby filling in the gaps of the topological relation graph $\mathcal{G}_r$. This is achieved by a frozen Large Language Model (LLM) extracting relation semantics from concise textual information. This graph prompting process requires no model fine-tuning, preserving the generalizability across distinct KGs.

Initial trials showed that asking the LLM for all possible relational interactions was inefficient and led to inaccuracies, involving too many requests and often resulting in flawed outputs. To overcome this, we streamlined the process by using the LLM to determine potential entity types for the heads and tails of relations. When two relations have matching entity types at their head sides, we assume an *h2h* interaction between them (similarly for other interaction types). This strategy cuts down LLM queries to a single one for each relation, simplifying the task and enhancing accuracy.

Instruction Prompt Design: To obtain entity types of two sides per relation, we design a series of instruction prompts, ensuring detailed guidance for each query relation. The prompt template is defined as $\mathcal{P}(\cdot)$, and $\mathcal{I} = \mathcal{P}(\mathcal{D}_{\mathcal{R}_s}, L_{\text{et}})$ is the input message to the LLM. $\mathcal{R}_s$ is the set of query relations with relation information $\mathcal{D}_{\mathcal{R}_s}$. The list of candidate entity types L_{et} is utilized to control the range of LLM responses. As shown in Figure 4, these prompts are distinct in two aspects:

(1) Relation Information Form:

- **des:** Short textual description of one relation.
- **exp:** Textual entity names of one support triple.
- **d&e:** Both description and example.

(2) Output Entity Type:

- **fixed:** Limited to predefined entity types.
- **refer:** Allow new types besides predefined.
- **free:** No constraints on the type range.

As prompt examples in Table 8 in the Appendix, the semantic information required for each relation is concise, facilitating user editing or automatic generation. The list of candidate entity types is domain-dependent; for a general KG, it includes types like person, location, and event. When the type category is **free**, LLM would output any reasonable entity types with no constraints.

Prompt Graph Construct: Collecting responses from the LLM, we obtain the set of entity types $\mathcal{S}(r, \text{'h'})$ and $\mathcal{S}(r, \text{'t'})$ for the head and tail sides of each relation r. Then, we construct the prompt graph $\mathcal{G}_p = \{\mathcal{R}, \mathcal{R}_{fund}, \mathcal{T}_p\}$, where the relation interactions in $\mathcal{T}_p$ obeys the following rules:

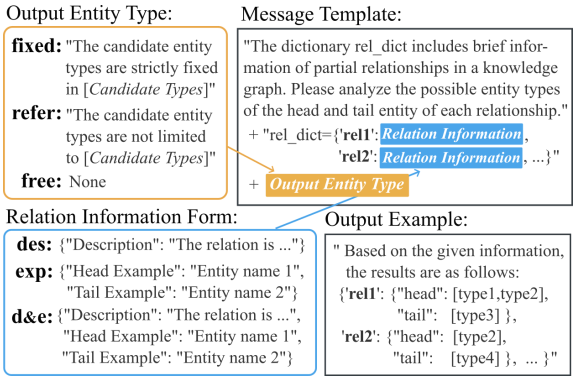


Figure 4: Examples of instruction prompts for LLMs.

$$\mathcal{S}(r_1, s_1) \cap \mathcal{S}(r_2, s_2) \neq \emptyset \implies \mathcal{T}_p[r_1, r_2, s_1, s_2] = 1;$$

$$\mathcal{T}_p[r_1, r_2, s_1, s_2] = \mathcal{T}_p[r_1, r_2', s_1, s_2'] = \mathcal{T}_p[r_1', r_2, s_1', s_2];$$

$$\mathcal{T}_p[r_1, r_2, s_1, s_2] = \begin{cases} \mathcal{T}_p[r_2, r_1, s_1, s_2] & s_1 = s_2 \\ \mathcal{T}_p[r_2, r_1, s_1', s_2'] & s_1 \neq s_2 \end{cases};$$

where $r_1, r_2 \in \mathcal{R}$ and the side symbol $s_i \in \{\text{'h'}, \text{'t'}\}$. The inverse relation of r_i is denoted as r_i', and s_i' denotes the opposite side of s_i. The connection value $\mathcal{T}_p[r_1, r_2, \text{'h'}, \text{'t'}]$ is equal to one when there is an *h2t* edge between r_1 and r_2, otherwise zero. The first rule specifies a sufficient condition for establishing relation interactions via entity types, while the other two rules detail the equivalence of inverse relations and inverse interactions. These rules collectively serve to minimize redundant computations in constructing $\mathcal{G}_p$.

3.3 Prompt Calibrator

This component aims to improve the quality of the LLM-based prompt graph $\mathcal{G}_p$ leveraging the ground-true information in the topological relation graph $\mathcal{G}_r$. Due to the natural gap between relation semantics and graph-specific topology, $\mathcal{G}_p$ cannot cover all expected interactions inevitably. Besides, mistaken edges would be included due to the relatively loose construction rules and the uncertainty of LLM response quality. Therefore, we design a novel calibrating process to extract high-confidence prompting edges that link the query relation with other relations in the KG. As shown in Algorithm 1, for each few-shot relation type r_q, two learning-free mechanisms are utilized to extract a series of calibrated interaction edges. When inferring r_q-invloved queries, these interaction edges would be injected into $\mathcal{G}_r$ to form the final r_q-specific prompt graph $\hat{\mathcal{G}}_r^{r_q}$.

Few-shot Support Expanding: In few-shot scenarios, support triples of this query relation in $\mathcal{G}_r$ are valuable. Therefore, we leverage these triples

Algorithm 1: Prompt Graph Calibrating

Input : relation graph $\mathcal{G}_r$, prompt graph $\mathcal{G}_p$, query relation r_q, threshold β.

Output : calibrated prompt graph $\hat{\mathcal{G}}_r$.

Gather r_q-related edges from two graphs:

$\mathcal{T}_r^{r_q} \leftarrow \{(e_i, r, e_j) \in \mathcal{T}_r \mid e_i = r_q \vee e_j = r_q\}$

$\mathcal{T}_p^{r_q} \leftarrow \{(e_i, r, e_j) \in \mathcal{T}_p \mid e_i = r_q \vee e_j = r_q\}$

Expand potential edges:

$\mathcal{T}_{ex}^{r_q} \leftarrow \textit{SupportExpanding}(r_q, \mathcal{T}_r)$

Identify unmatched edges in $\mathcal{G}_p$:

$\mathcal{T}_m \leftarrow (\mathcal{T}_p - \mathcal{T}_p^{r_q}) - (\mathcal{T}_r - \mathcal{T}_r^{r_q})$

Filter prompting edges of r_q:

$\mathcal{T}_{conf}^{r_q} \leftarrow \textit{ConflictFiltering}(\mathcal{T}_{ex}^{r_q} \cup \mathcal{T}_p^{r_q}, \mathcal{T}_m, \beta)$

Inject prompting edges into $\mathcal{G}_r$:

$\hat{\mathcal{G}}_r^{r_q} \leftarrow \{\mathcal{R}, \mathcal{R}_{fund}, \mathcal{T}_{conf}^{r_q} \cup \mathcal{T}_r\}$

to find more potential interaction edges via the following rules:

$$\exists r_j, (r_q, i_r, r_j) \in \mathcal{G}_r, i_r = [s_1, s_2] \implies \mathcal{S}(r_q, s_1) \leftarrow \mathcal{S}(r_q, s_1) \cup \mathcal{S}(r_j, s_2)$$

$$\exists r_k, (r_q, i_{r1}, r_j), (r_j, i_{r2}, r_k) \in \mathcal{G}_r, i_{r1} = [s_1, s_2], i_{r2} = [s_2, s_2] \implies \mathcal{T}_p[r_q, r_k, s_1, s_2] = 1;$$

where $i_r = [s_1, s_2]$ denotes an interaction edge whose type is 's_1-to-s_2' (e.g., '*h2t*'). Given the relations involved by support triples, the first rule inserts their entity types into the type set of r_q, thereby expanding more potential edges. The second rule adds new interaction edges directly by finding the 2-hop neighbors of the query relation in $\mathcal{G}_r$. These new edges $\mathcal{T}_{ex}^{r_q}$ as well as original r_q related edges $\mathcal{T}_p^{r_q}$ will be filtered in the next step.

Type Conflict Filtering: In a specific KG, two relations sharing an entity type in semantics may link to two disjoint entity sets, thereby the interaction edge between them is mistaken. Because r_q lacks support triples, we can only detect whether the relations $\mathcal{R}_s$, having the same entity type as r_q at side s, have conflicts with each other. Therefore, we compare the $\mathcal{R}_s$-involved edges in $\mathcal{G}_r$ and $\mathcal{G}_p$, and extract unmatched edges $\mathcal{T}_m$. If a relation has many unmatched edges with other relations in $\mathcal{R}_s$, it is less likely connected with r_q. The conflict determination of each relation is defined as follows:

$$C(r_j) = |\{r_j | (r_j, i, r_k) \in \mathcal{T}_m, r_j, r_k \in \mathcal{E}_p\}| \geq \beta$$

The threshold hyperparameter β determines the maximum accepted number of unmatched edges for each relation r_j. If $C(r_j)$ is true, the interaction edges connecting r_j and r_q would be removed.

4 Experiments

We extensively evaluate our method on K-shot inductive reasoning of KGs. In particular, we wish to answer the following research questions: **Q1**: How effective is our method under the low-resource scenarios of distinct KGs? **Q2**: How do different LLM prompts impact prompt graphs and KG inductive reasoning? **Q3**: How do the main components of our method impact the performance? **Q4**: How does our method perform in full-shot scenarios? **Q5**: How efficient is our model compared with traditional approaches? **Q6**: How does GNN reasoning change before and after injecting the prompt graph in case studies? Due to the space limitation, discussions about Q5 and Q6 are detailed in the Appendix.

4.1 Experimental Setup

Low-resource Datasets. We conduct K-shot inductive reasoning experiments on 108 low-resource datasets, constructed upon the 12 datasets used by the InGram work(Lee et al., 2023). The InGram datasets were derived from three real-world KG benchmarks: FB15k237 (Toutanova and Chen, 2015), Wikidata68K (Gesese et al., 2022), and NELL-995 (Xiong et al., 2017), abbreviated as FB, WK, and NL. There are four KG datasets (we call v1-v4) for each benchmark. For FB and NL datasets, we utilize the word-segmented relation names as the short description, while extracting official relation descriptions from WikiData for WK datasets. Both structural and textual statistics for these datasets can be found in Table 6 and Table 7 in the Appendix. Based on the InGram datasets, we create K-shot datasets tailored for 3-shot, 1-shot, and 0-shot scenarios. In each InGram dataset, we randomly retain K support triples for each query relation r_q and mask others when reasoning from $\mathcal{G}_{inf}^{r_q}$. Ensuring a robust evaluation of our model, we perform this sampling process three times to create three variants for each few-shot dataset.

Baselines and Implementation. We compare PROLINK with eight baseline methods. NBFNet (Zhu et al., 2021), RED-GNN (Zhang and Yao, 2022), InGram (Lee et al., 2023), DEqInGram (Gao et al., 2023), and ISDEA (Gao et al., 2023) are state-of-the-art GNN-based inductive models. The recent pre-trained model ULTRA(Xg) (Galkin et al., 2023) has three variants (3g, 4g, 50g), of which X denotes the amount of pre-trained KG datasets. Previous few-shot link prediction and

Model	FB:v1	FB:v2	FB:v3	FB:v4	WK:v1	WK:v2	WK:v3	WK:v4	NL:v1	NL:v2	NL:v3	NL:v4	AVG
NBFNet	9.2	1.9	1.3	0.0	3.3	0.1	0.0	0.0	0.0	1.4	0.5	0.0	1.5
REDGNN	6.8	6.6	11.0	6.5	0.0	0.3	0.0	0.0	8.1	7.1	3.6	2.5	4.4
DEqInGram	7.6	0.9	7.0	1.1	0.6	0.2	0.5	5.2	6.8	4.6	4.3	7.4	3.9
ISDEA	1.8	1.3	0.9	0.8	1.4	0.3	1.2	0.3	3.5	3.1	3.5	1.9	1.7
InGram	9.8	6.2	11.4	7.0	7.0	0.2	2.2	2.3	12.2	12.4	13.8	14.9	8.3
ULTRA(3g)	35.9	33.4	33.5	31.9	43.0	9.2	17.2	**11.4**	27.5	29.1	32.4	36.9	28.5
ULTRA(4g)	40.4	35.8	32.7	31.6	13.8	6.3	17.6	9.2	25.8	25.5	26.5	30.0	24.6
ULTRA(50g)	36.8	35.4	32.9	31.0	15.7	5.5	16.6	8.6	22.7	22.0	21.9	26.2	22.9
Our(Llama2-7B)	51.3	**48.0**	**40.8**	**37.7**	46.1	10.8	18.6	11.1	33.5	**37.2**	33.5	39.4	34.0
Our(Llama2-13B)	49.2	46.8	40.5	35.8	**46.3**	11.6	18.6	11.0	32.0	36.7	35.4	40.3	33.7
Our(Mistral-7B)	49.4	46.7	40.2	36.7	46.1	11.5	**18.8**	10.8	32.3	33.3	35.4	39.4	33.4
Our(GPT-3.5)	50.5	47.1	**40.8**	38.0	45.5	11.5	18.7	11.3	**34.2**	35.1	**35.5**	**41.1**	**34.1**
Our(GPT-4)	**51.7**	47.3	40.4	37.5	45.6	**11.8**	**18.8**	11.2	33.0	35.2	33.8	39.8	33.8

Table 1: 3-shot inductive reasoning results on three series of datasets, evaluated with Hits@10 (%).

Model	FB:v1	FB:v2	FB:v3	FB:v4	WK:v1	WK:v2	WK:v3	WK:v4	NL:v1	NL:v2	NL:v3	NL:v4	AVG
NBFNet	8.7	1.8	1.3	0.0	1.9	0.1	0.0	0.0	0.0	7.1	0.4	0.0	1.8
REDGNN	6.3	6.4	10.3	5.4	0.2	0.1	0.4	0.0	6.3	6.4	1.9	0.9	3.7
DEqInGram	6.3	0.7	5.0	1.3	0.4	0.2	0.4	1.2	9.6	4.3	3.5	6.6	3.3
ISDEA	2.2	0.8	0.7	0.6	1.1	0.2	1.1	0.1	2.6	2.7	2.3	1.8	1.4
InGram	6.9	5.7	6.2	4.9	8.0	0.3	1.4	0.9	9.6	8.2	8.5	11.6	6.0
ULTRA(3g)	27.0	25.6	23.5	20.4	24.9	4.2	10.1	2.9	21.6	19.4	22.3	25.5	19.0
ULTRA(4g)	29.6	27.3	23.0	22.1	4.3	2.5	10.1	2.0	18.3	17.1	15.1	16.0	15.6
ULTRA(50g)	27.5	25.4	23.0	23.5	6.5	2.3	9.6	2.0	18.5	17.6	15.5	18.6	15.8
Our(Llama2-7B)	44.2	**41.7**	36.3	29.8	37.1	5.6	11.6	2.9	29.5	27.7	24.7	31.1	26.9
Our(Llama2-13B)	44.5	40.4	35.0	28.0	**38.4**	5.4	**11.9**	3.4	27.2	27.9	25.6	31.2	26.6
Our(Mistral-7B)	44.1	40.4	35.9	29.5	38.1	6.1	**11.9**	3.1	27.0	25.9	**26.8**	28.5	26.4
Our(GPT-3.5)	43.8	**41.7**	36.9	**32.4**	38.1	5.8	11.8	**3.5**	29.2	**29.1**	26.6	**32.2**	**27.6**
Our(GPT-4)	**45.1**	40.9	**37.0**	31.9	37.7	**6.3**	11.8	**3.5**	**29.7**	26.7	26.0	30.0	27.2

Table 2: 1-shot inductive reasoning results on three series of datasets, evaluated with Hits@10 (%).

text-based methods are neglected because they cannot perform on our task settings. For PROLINK, we train the GNN reasoner following the settings of ULTRA(3g) and employ five LLMs in the LLM prompter, including Llama2-7B, Llama2-13B (Touvron et al., 2023), Mistral-7B (Das et al., 2017), GPT-3.5 (Brown et al., 2020), and GPT-4 (OpenAI, 2023). We utilize two evaluation metrics, MRR (Mean Reciprocal Rank) and Hits@N. Hyperparameters are selected via grid search according to the metrics on the validation set. All experiments are performed on Intel Xeon Gold 6238R CPU @ 2.20GHz and $4\times$ NVIDIA RTX A30 GPUs. Implementation details and hyperparameter configurations are shown in Appendix A.

4.2 Main Experimental Results (RQ1)

We compare PROLINK with baselines on inductive reasoning tasks under the 3-shot, 1-shot, and 0-shot settings. We report the average Hits@10 results over three variants of each K-shot InGram dataset in Table 1, Table 2, and Table 3. More detailed results on other metrics can be found in the Appendix. We observe that the first four GNN-based methods underperform pre-trained ULTRA and our method. NBFNet and REDGNN struggle with unseen relations, while InGram, ISDEA, and DEqInGram rely on sufficient support triples to form a similar distribution of node degrees. Conversely, ULTRA excels in low-resource settings due to its pre-training on multiple KGs, especially in the 3-shot scenario, though more KG pre-training in ULTRA(50g) doesn't markedly boost performance. Pre-trained on three KGs like ULTRA(3g), PROLINK significantly outperforms previous methods by a wide margin on average. Notably, in the 0-shot setting, the average Hits@10 for Our(GPT-4) method is twice as high as that of ULTRA, highlighting the effectiveness of our prompting paradigm. In the comparison of different LLMs, GPT-3.5 and GPT-4 exhibit superior performance in the 0-shot setting. Due to our simplification for LLM requests, the lightweight Llama2-7B already performs well in few-shot scenarios, which indicates the robustness of our method.

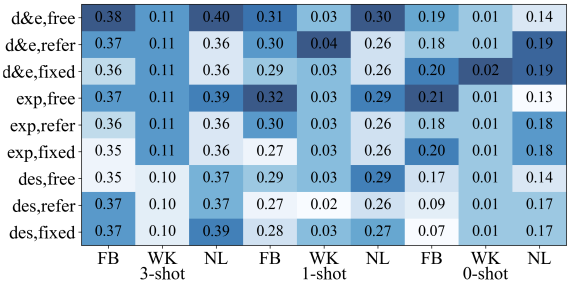
4.3 Performance of Different Prompts (RQ2)

To generate the prompt graph, we employ a series of textual prompts to guide the LLM prompter. As shown in Figure 5, we compare the performance of Our(GPT-4) on the three $v4$ datasets by varying the *Relation Information Form* and *Output Entity Type*, respectively. In Figure 5(a), we observe that the 'd&e' form outperforms the other two in most scenarios, because textual descriptions reflect the core features while two entity names indicate the direction of the relation. Regarding entity types, the 'free' setting works better in few-shot scenarios. In zero-shot datasets, prompts using ('d&e', 'fixed') outperform the others.

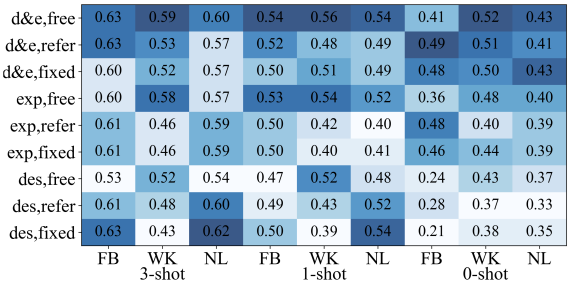
In Figure 5(b), we analyse the effectiveness of the prompt graph by calculating the F1 metric between the prompt interaction edges and ground-truth edges in the full-shot KGs. The trend of F1

Model	FB:v1	FB:v2	FB:v3	FB:v4	WK:v1	WK:v2	WK:v3	WK:v4	NL:v1	NL:v2	NL:v3	NL:v4	AVG
NBFNet	8.7	1.8	1.3	0.0	1.9	0.1	0.0	0.0	0.0	7.1	0.4	0.0	1.8
REDGNN	6.3	6.4	10.3	5.4	0.2	0.1	0.0	0.0	6.0	6.4	0.8	2.9	3.7
DEqInGram	0.3	0.4	1.3	0.7	2.3	0.1	0.0	0.1	4.0	3.2	2.2	2.0	1.4
ISDEA	1.7	0.1	0.2	0.2	0.2	0.0	0.4	0.0	0.9	1.4	0.3	1.2	0.6
InGram	1.7	4.3	5.8	2.7	5.4	0.1	0.2	0.2	7.0	7.0	2.9	8.1	3.8
ULTRA(3g)	14.5	12.8	10.1	9.8	1.9	0.8	1.3	0.4	11.3	6.8	6.1	6.0	6.8
ULTRA(4g)	15.5	14.8	10.1	10.7	1.8	1.0	1.3	0.4	9.6	7.1	5.9	6.1	7.0
ULTRA(50g)	10.3	10.0	6.3	10.0	2.3	1.2	1.3	0.4	8.9	7.7	5.8	4.6	5.7
Our(Llama2-7B)	19.3	20.3	13.3	15.3	2.9	1.0	1.6	0.7	18.0	**19.0**	9.5	16.1	11.4
Our(Llama2-13B)	20.9	17.9	13.6	12.1	7.8	1.1	1.6	1.6	**19.1**	18.4	**15.5**	14.8	12.0
Our(Mistral-7B)	25.4	25.4	22.2	19.5	4.3	1.2	1.7	1.1	17.2	15.6	14.3	11.9	13.3
Our(GPT-3.5)	**30.0**	**26.7**	24.0	**22.5**	30.4	2.0	3.0	1.4	17.8	17.8	15.3	17.7	**17.4**
Our(GPT-4)	28.4	25.1	**25.6**	21.3	**35.0**	**2.4**	**3.1**	**1.8**	18.3	14.0	14.7	**19.1**	**17.4**

Table 3: 0-shot inductive reasoning results on three series of datasets, evaluated with Hits@10 (%).



(a) Hits@10 of Query Prediction



(b) F1 Metrics of Interaction Edges

Figure 5: Performance of different prompt settings in GPT-4. Darker colors indicate higher values.

	Model	FB:v4	WK:v4	NL:v4
3-shot	Our(GPT-4)	**37.5**	**11.2**	**38.8**
	with ULTRA	34.6	11.0	37.1
	with OODKG	36.8	10.6	35.9
	w/o Prompt	37.7	11.1	37.7
	w/o Calibrator	37.5	10.6	37.4
	w/o LLM	30.9	10.1	33.4
1-shot	Our(GPT-4)	31.9	**3.5**	**29.3**
	with ULTRA	28.1	**3.5**	27.0
	with OODKG	**32.3**	3.2	26.6
	w/o Prompt	28.0	2.6	25.9
	w/o Calibrator	31.1	2.8	**29.3**
	w/o LLM	23.8	2.6	25.7
0-shot	Our(GPT-4)	**21.3**	**1.8**	**18.5**
	with ULTRA	18.2	1.4	15.5
	with OODKG	21.2	1.3	18.3
	w/o Prompt	8.9	0.4	3.1
	w/o Calibrator	20.2	1.4	**18.5**
	w/o LLM	8.9	0.4	3.1
full-shot	Our(GPT-4)	63.1	28.5	**67.7**
	with Llama2-7B	**63.3**	27.8	**67.7**
	w/o Prompt	63.1	26.9	67.1
	InGram	37.1	16.9	50.6
	ULTRA(3g)	62.9	**28.9**	63.2

Table 4: Ablation studies evaluated with Hits@10 (%).

metrics across different prompts is similar to that of Hits@10, it shows that a prompt graph closer to ground truth leads to better performance. The results indicate the importance of prompt settings and more refined prompts would be our future work.

4.4 Ablation Studies (RQ3, RQ4)

To validate the impact of the three components on model performance, we conduct **ablation experiments** on PROLINK with GPT-4, as shown in Table 4. The variant 'with ULTRA' employs ULTRA(3g) to replace our pre-trained GNN reasoner and the performance decline indicates the effect of our enhancing techniques. 'w/o Prompt' denotes utilizing the pre-trained KG reasoner directly without prompt graphs, which performs well in 3-shot scenarios but struggles with fewer-shot ones. The next two variants prove that the LLM prompter and the prompt calibrator both enhance the model's performance. **'with OODKG'** denotes pre-training the GNN reasoner with three KGs out of the evaluation KG distribution, including YAGO3-10, DBPedia100k, and WN18RR. Its competitive performance indicates the robustness of our pre-training strategy. We also verify PROLINK in the **full-shot scenarios**. Our(GPT-4) and Our(Llama2-7b) outperform ULTRA(3g) and InGram on FB:v4 and NL:v4 datasets. The results prove the good applicability of our method in practice.

5 Conclusions

We propose a novel pre-training and prompting framework, PROLINK, for low-resource inductive reasoning across arbitrary KGs. To generate an effective prompt graph for few-shot relation types, we design enhancing techniques for the GNN reasoner and instruction prompts for the LLM prompter. Besides, a novel prompt calibrator is proposed to mitigate the potential noise and achieve the information alignment of the above two components. Extensive Experiments have verified that the PROLINK achieves significant performance in both few-shot and zero-shot scenarios. Besides, our PROLINK requires no model fine-tuning, thereby having advantages of better efficiency and scalability than previous GNN-based methods.

6 Limitations.

Here, we discuss two potential limitations of PROLINK. First, unlike recent GNN-based fully inductive methods, our method requires a brief relational context for each relation type. Despite our efforts to minimize text requirements, it may be unavailable in certain scenarios without manual inputs from users. Second, the usage of textual context in our method is not sufficient enough. To circumvent the need for additional model training, we streamline the LLM queries by only asking for entity types. Incorporating semantic embeddings into GNNs or fine-tuning the LLM prompter to leverage relation semantics could further enhance performance. Exploring this avenue will constitute a primary focus of our future work. The potential risks of our work may include generating factual triples about privacy or fake information, depending on the legality and reliability of the input data. In addition, we utilize the ChatGPT AI assistant when polishing some paragraphs of the paper draft.

References

Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Charles Tapley Hoyt, and William L Hamilton. 2022. Understanding the performance of knowledge graph embeddings in drug discovery. *Artificial Intelligence in the Life Sciences*, 2:100036.

Antoine Bordes, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning*, 94:233–259.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. 2023. Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11489–11503. Association for Computational Linguistics.

Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. 2021. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6271–6278. AAAI Press.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4216–4225. Association for Computational Linguistics.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over knowledge bases with reinforcement learning. In *6th Workshop on Automated Knowledge Base Construction, AKBC@NIPS 2017, Long Beach, California, USA, December 8, 2017*. OpenReview.net.

Daniel Daza, Michael Cochez, and Paul Groth. 2021. Inductive entity representations from text via link prediction. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 798–808. ACM / IW3C2.

Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z. Pan, and Huajun Chen. 2019. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 678–685. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Mikhail Galkin, Etienne G. Denis, Jiapeng Wu, and William L. Hamilton. 2022. Nodepiece: Compositional and parameter-efficient representations of large

knowledge graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2023. Towards foundation models for knowledge graph reasoning. *CoRR*, abs/2310.04562.

Jianfei Gao, Yangze Zhou, and Bruno Ribeiro. 2023. Double permutation equivariance for knowledge graph completion. *arXiv preprint arXiv:2302.01313*.

Xiou Ge, Yun-Cheng Wang, Bin Wang, and C.-C. Jay Kuo. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6947–6965. Association for Computational Linguistics.

Yuxia Geng, Jiaoyan Chen, Jeff Z Pan, Mingyang Chen, Song Jiang, Wen Zhang, and Huajun Chen. 2023a. Relational message passing for fully inductive knowledge graph completion. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1221–1233. IEEE.

Yuxia Geng, Jiaoyan Chen, Yuhang Zeng, Zhuo Chen, Wen Zhang, Jeff Z. Pan, Yuxiang Wang, and Xiaoliang Xu. 2023b. Prompting disentangled embeddings for knowledge graph completion with pretrained language model. *CoRR*, abs/2312.01837.

Genet Asefa Gesese, Harald Sack, and Mehwish Alam. 2022. RAILD: towards leveraging relation features for inductive link prediction in knowledge graphs. In *Proceedings of the 11th International Joint Conference on Knowledge Graphs, IJCKG 2022, Hangzhou, China, October 27-28, 2022*, pages 82–90. ACM.

Qian Huang, Hongyu Ren, and Jure Leskovec. 2022. Few-shot relational reasoning via connection subgraph pretraining. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. 2023. InGram: Inductive knowledge graph embedding via relation graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 18796–18809. PMLR.

Rui Li, Xu Chen, Chaozhuo Li, Yanming Shen, Jianan Zhao, Yujing Wang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, and Xing Xie. 2023. To copy rather than memorize: A vertical learning paradigm for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6335–6347. Association for Computational Linguistics.

Ben Liu, Miao Peng, Wenjie Xu, and Min Peng. 2023. Neighboring relation enhanced inductive knowledge graph link prediction via meta-learning. *World Wide Web (WWW)*, 26(5):2909–2930.

Elan Markowitz, Keshav Balasubramanian, Mehrnoosh Mirtaheri, Murali Annavaram, Aram Galstyan, and Greg Ver Steeg. 2022. Statik: Structure and text for inductive knowledge graph completion. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 604–615. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shichao Pei, Ziyi Kou, Qiannan Zhang, and Xiangliang Zhang. 2023. Few-shot low-resource knowledge graph completion with multi-view task representation generation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 1862–1871. ACM.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: end-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15321–15331.

Tara Safavi and Danai Koutra. 2020. Codex: A comprehensive knowledge graph completion benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8328–8350.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9448–9457. PMLR.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66.

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, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Bin Wang, Guangtao Wang, Jing Huang, Jiaxuan You, Jure Leskovec, and C.-C. Jay Kuo. 2021a. Inductive learning on commonsense knowledge graph completion. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE.

Kai Wang, Yu Liu, Dan Lin, and Michael Sheng. 2021b. Hyperbolic geometry is not necessary: Lightweight euclidean-based models for low-dimensional knowledge graph embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 464–474.

Kai Wang, Siqiang Luo, and Dan Lin. 2023. River of no return: Graph percolation embeddings for efficient knowledge graph reasoning. *arXiv preprint arXiv:2305.09974*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021c. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguistics*, 9:176–194.

Han Wu, Jie Yin, Bala Rajaratnam, and Jianyuan Guo. 2023. Hierarchical relational learning for few-shot knowledge graph completion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 564–573. Association for Computational Linguistics.

Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, and Chao Chen. 2022. Cycle representation learning for inductive relation prediction. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 24895–24910. PMLR.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *CoRR*, abs/2308.07134.

Xiangxiang Zeng, Xiang Song, Tengfei Ma, Xiaoqin Pan, Yadi Zhou, Yuan Hou, Zheng Zhang, Kenli Li, George Karypis, and Feixiong Cheng. 2020. Repurpose open data to discover therapeutics for covid-19 using deep learning. *Journal of proteome research*, 19(11):4624–4636.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. 2020. Few-shot knowledge graph completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3041–3048. AAAI Press.

Yongqi Zhang and Quanming Yao. 2022. Knowledge graph reasoning with relational digraph. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 912–924. ACM.

Jincheng Zhou, Beatrice Bevilacqua, and Bruno Ribeiro. 2023. An ood multi-task perspective for link prediction with new relation types and nodes. *arXiv preprint arXiv:2307.06046*.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *CoRR*, abs/2305.13168.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29476–29490.

A Implementation Details and Hyperparameters

We introduce the statistics of pre-training and evaluation datasets in Table 5, Table 6. Following previous work (Zhang and Yao, 2022; Zhu et al., 2021), we augment the triples in each $\mathcal{G}$ with reverse and identity relations. The augmented triple set $\mathcal{T}^+$ is defined as: $\mathcal{T}^+ = \mathcal{T} \cup \{(e_t, r^{-1}, e_h) | (e_h, r, e_t) \in \mathcal{T}\} \cup \{(e, r^i, e) | e \in \mathcal{E}\}$, where the relation r^{-1} is the reverse relation of a relation r, the relation r^i refers to the identity relation, and the number of augmented triples is $|\mathcal{T}^+| = 2|\mathcal{T}| + |\mathcal{E}|$.

Baseline Implementation Details. We train NBFNet, RED-GNN, and ISDEA using the training graph of InGram datasets (Lee et al., 2023) and evaluate the low-resource datasets. When evaluating InGram, DEqInGram, and ULTRA, we utilize their officially provided checkpoints directly. Although there are some previous baselines (Galkin et al., 2022; Sadeghian et al., 2019; Teru et al., 2020), they exhibit near-zero performance in original full-shot inductive tasks reported by InGram (Lee et al., 2023). Therefore, we ignore them in this more challenging low-resource reasoning task.

PROLINK Implementation Details. We train our KG reasoner with three commonly-used KG datasets, WN18RR (Bordes et al., 2014), FB15k237 (Toutanova and Chen, 2015), and CodexM (Safavi and Koutra, 2020), following the settings of ULTRA(3g). Concerns about potential relation leakage during pre-training can be ignored because neither our method nor ULTRA learns relation-specific parameters. Moreover, in low-resource settings, the available relational information is limited, and markedly distinct from the original data. We employ five popular LLMs as the LLM prompter, including Llama2-7B, Llama2-13B (Touvron et al., 2023), Mistral-7B (Das et al., 2017), GPT-3.5 (Brown et al., 2020), and GPT-4 (OpenAI, 2023). Llama2 and Mistral-7B are hosted directly on our servers, while the GPT-3.5 and GPT-4 models are accessed through the OpenAI API. For Prompt Calibrater, we set the loss balancing ratio α to 0.5, and the low-resource masking ratio γ to 0.1. The filtering threshold β is selected from $\{1, 3, 5, \text{Mean}, \text{Max}\}$, in which the latter two are calculated from values of all relation conflicts.

All experiments are performed on Intel Xeon Gold 6238R CPU @ 2.20GHz and NVIDIA RTX A30 GPUs (four for pretraining and one for evaluation), and are implemented in Python using the PyTorch framework. Our source code is implemented based on ULTRA[4], which is available under the MIT License. We utilize the Llama2 and Mistral models under the corresponding licenses and call the GPT-3.5 and GPT-4 APIs obeying OpenAI terms. All employed KG datasets are open and commonly used.

Dataset	$\lvert\mathcal{E}_{tr}\rvert$	$\lvert\mathcal{R}_{tr}\rvert$	$\lvert\mathcal{T}_{tr}\rvert$ #Train	#Validation	#Test
WN18RR	40.9k	11	86.8k	3.0k	3.1k
FB15k-237	14.5k	237	272.1k	17.5k	20.4k
CodexMedium	17.0k	51	185.5k	10.3k	10.3k

Table 5: Statistics of pre-training KG datasets.

	FB			WK			NL		
	$\lvert\mathcal{E}_{inf}\rvert$	$\lvert\mathcal{R}_{inf}\rvert$	$\lvert\mathcal{T}_{inf}\rvert$	$\lvert\mathcal{E}_{inf}\rvert$	$\lvert\mathcal{R}_{inf}\rvert$	$\lvert\mathcal{T}_{inf}\rvert$	$\lvert\mathcal{E}_{inf}\rvert$	$\lvert\mathcal{R}_{inf}\rvert$	$\lvert\mathcal{T}_{inf}\rvert$
v1	2146	120	3717	3228	74	5652	4097	216	28579
v2	2335	119	4294	9328	93	16121	4445	205	19394
v3	1578	116	3031	2722	65	5717	2792	186	15528
v4	1709	53	3964	12136	37	22479	2624	77	11645

Table 6: Statistics of evaluating InGram datasets.

Dataset	Textual Form	Details
FB	Description	Average Words: 4.53 Average Tokens: 61.56
	(Example)	"/people/person/profession"
	Entity Name	Average Words: 2.30 Average Tokens: 15.21
	(Example)	"Stan Lee"
WK	Description	Average Words: 15.93 Average Tokens: 103.42
	(Example)	"residence: the place where the person is or has been, resident"
	Entity Name	Average Words: 2.57 Average Tokens: 17.68
	(Example)	"coquette (film)"
NL	Description	Average Words: 3.86 Average Tokens: 24.46
	(Example)	"person graduated from university"
	Entity Name	Average Words: 2.98 Average Tokens: 20.91
	(Example)	"city: portland"

Table 7: Statistics of KG textual data.

B Efficiency Analysis (RQ5)

The primary benefit of PROLINK lies in its ability to infer new, arbitrary Knowledge Graphs (KGs) without the need for training. It stems from the fact that both the GNN reasoner and the LLM prompter remain parameter-frozen during the prompting process. As a result, the only computational overhead introduced is associated with the prompt graphs. Except for the LLM requests for each relation in the KG, the prompt graph is generated only once for each few-shot relation type, whose computational complexity would not exceed $\mathcal{O}(|\mathcal{R}_{inf}|^2)$. Given that the quantity of relations in most KGs is significantly lower than that of entities, the time and space costs in this process are negligible. Specifically, the checkpoint file size of our KG reasoner is only 2.18 MB, which is similar to that of ULTRA (2.03 MB). The total pretraining time is around eight

[4]https://github.com/DeepGraphLearning/ULTRA

Prompt Setting	(FB, des, fixed)
Task Description	The dictionary rel_dict includes brief information of partial relationships in a knowledge graph. Please analyze the possible entity types of each relationship's head and tail entities.
EntType	The candidate entity types are strictly fixed in ["genre/type", "person", "animal", "location/place", "organization", "creative work", "time", "profession", "event", "actual item", "language"].
RelInfo	rel_dict = {"rel0": "music artist origin", "rel1": "film actor film. film performance film"}
Output	Here are the results: { rel0: {"head": ["person"], "tail": ["location/place"]}, rel1: {"head": ["person"], "tail": ["creative work"]} }
Prompt Setting	(WK, exp, refer)
Task Description	The dictionary rel_dict includes brief information of partial relationships in a knowledge graph. Please analyze the possible entity types of each relationship's head and tail entities.
EntType	The candidate entity types are not limited to ["genre/type", "person", "animal", "location/place", "organization", "creative work", "time", "profession", "event", "actual item", "language"].
RelInfo	rel_dict = {"rel6": {"head entity": "lesser asiatic yellow bat", "tail entity": "morphospecies"}, "rel8": {"head entity": "hexacinia", "tail entity": "peacock flies"}}
Output	Here are the results: { rel6: {"head": ["animal"], "tail": ["morphospecies"]}, rel8: {"head": ["creative work"], "tail": ["genre/type"]} }
Prompt Setting	(NL, d&e, free)
Task Description	The dictionary rel_dict includes brief information of partial relationships in a knowledge graph. Please analyze the possible entity types of each relationship's head and tail entities.
EntType	*None*
RelInfo	rel_dict = { "rel0": {"description": "sport fans in country", "head entity": "sport: skiing", "tail entity": "country: america"}, "rel1": {"description": "animal eat vegetable", "head entity": "bird: chickens", "tail entity": "vegetable: corn"}}
Output	Here are the results: { rel0: {"head": ["sport"], "tail": ["country"]}, rel1: {"head": ["animal"], "tail": ["vegetable"]} }

Table 8: Examples of textual prompts on three KGs.

Dataset	K	LLM	Input	Output	β
FB:v1	3	gpt4	des	refer	1
FB:v1	1	gpt4	des	refer	1
FB:v1	0	gpt3.5	exp	free	max
FB:v2	3	gpt4	des	refer	1
FB:v2	1	llama7b	des	fixed	5
FB:v2	0	gpt3.5	exp	free	max
FB:v3	3	llama7b	d&e	fixed	5
FB:v3	1	gpt4	exp	free	5
FB:v3	0	gpt4	exp	fixed	max
FB:v4	3	gpt3.5	d&e	free	max
FB:v4	1	gpt3.5	d&e	free	max
FB:v4	0	gpt3.5	d&e	refer	max
NL:v1	3	gpt3.5	des	free	max
NL:v1	1	gpt4	d&e	free	max
NL:v1	0	llama13b	d&e	free	max
NL:v2	3	llama7b	des	free	3
NL:v2	1	gpt3.5	des	free	5
NL:v2	0	llama7b	d&e	free	max
NL:v3	3	gpt3.5	des	free	max
NL:v3	1	mistral7b	des	refer	max
NL:v3	0	llama13b	exp	refer	max
NL:v4	3	gpt3.5	des	free	max
NL:v4	1	gpt3.5	des	free	max
NL:v4	0	gpt4	d&e	refer	max
WK:v1	3	llama13b	exp	free	max
WK:v1	1	llama13b	exp	free	max
WK:v1	0	gpt4	d&e	free	3
WK:v2	3	gpt4	exp	free	1
WK:v2	1	gpt4	d&e	free	3
WK:v2	0	gpt4	d&e	refer	mean
WK:v3	3	gpt4	d&e	free	3
WK:v3	1	mistral7b	d&e	free	1
WK:v3	0	gpt4	d&e	fixed	max
WK:v4	3	gpt3.5	des	free	max
WK:v4	1	gpt4	d&e	refer	1
WK:v4	0	gpt4	d&e	fixed	5

Table 9: Hyperparameter settings of best Hits@10.

GPU hours for ten training epochs. The evaluation time on each dataset only costs several minutes. Regarding the model scalability, PROLINK can be applied in large-scale KGs the same as ULTRA (Galkin et al., 2023). This work focuses on low-resource challenges, more experiments on large-scale datasets will be our future work.

C Case Studies (RQ6)

We select several queries from the zero-shot NL:v4 dataset and compare the outputs of Our(GPT-4) with and without the prompt graph. The final relation graphs and top five outputted entities are shown in Figure 6. Generally, we observe that PROLINK injects multiple prompt edges into the relation graph which directly changes the outputted entity ranking. Most linked relations by prompt edges are reasonable, thereby improving the relative relation embedding for the query relation.

D Detailed Related Work

KG Inductive Reasoning: Traditional *transductive* KG embedding models, represented by TransE (Bordes et al., 2013), DistMult (Yang et al., 2015) and RotatE (Sun et al., 2019), learn continuous vectors in the embedding space to represent each entity and relation in the knowledge graph. In contrast, *inductive* KG reasoning methods (Zhu et al., 2021) overcome this limitation by generalizing to KGs with unseen entities or relations. Most existing inductive methods (Yan et al., 2022; Wang et al., 2021a; Liu et al., 2023; Chen et al., 2021) leverage Graph Neural Networks (GNN) to generate "relative" entity embeddings, by extracting local structural features from an induced graph of the query entity. GraIL (Teru et al., 2020) extracts an enclosing subgraph between the query entity and each candidate entity, but suffers from high computational complexity. NBFNet (Zhu et al., 2021) and RED-GNN (Zhang and Yao, 2022) propagate query features through the L-hop neighborhood subgraph of the query entity. These inductive methods struggle to generalize to KGs with new relation types, as the entity embeddings are still a function of a predetermined relation vocabulary.

Few-shot and Unseen Relation Reasoning: To generalize to unseen relations, early efforts have explored meta-learning for few-shot link prediction, which predicts KG facts of unseen relations using a limited number of support triples (Chen et al., 2019; Zhang et al., 2020; Huang et al., 2022; Pei et al., 2023; Wu et al., 2023). However, those methods

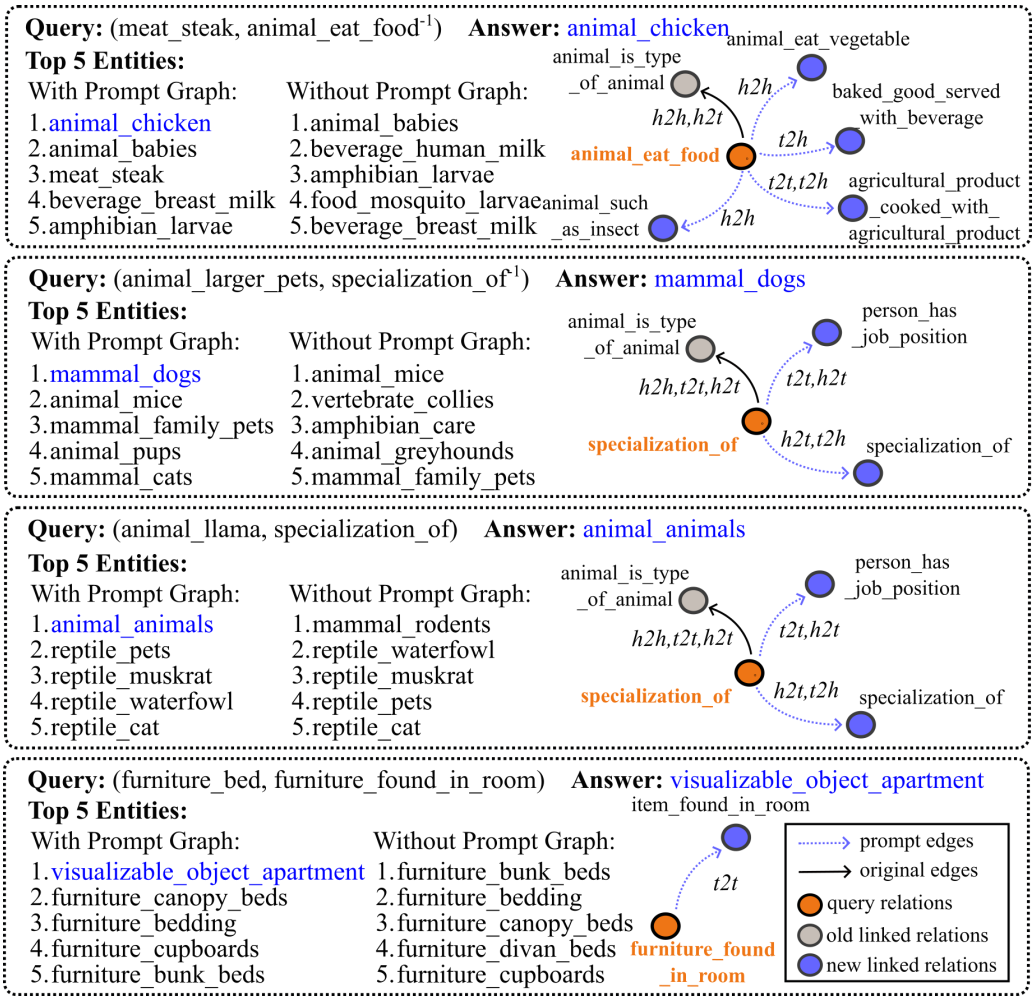


Figure 6: Case studies on the one-shot NL:v4 dataset.

cannot work on an entire unseen inference graph. Recent approaches focus on constructing graphs of relations to generalize to unseen relations (Geng et al., 2023a; Zhou et al., 2023). InGram (Lee et al., 2023) relies on a featurization strategy based on the discretization of node degrees, which is effective only for KGs with a similar relational distribution and falls short in transferring to arbitrary KGs. ISDEA (Gao et al., 2023) utilizes relation exchangeability in multi-relational graphs, which imposes stringent assumptions on the class of transferable relations coming from the same distribution. ULTRA (Galkin et al., 2023) employs a pre-training and fine-tuning framework, with a global relation graph extracted from the entire set of KG triples. However, the effectiveness of the above methods is contingent on sufficient support triples for the unseen relations in the inference KG, struggling with few-shot relation types in low-resource scenarios that limit their applicability.

Text-based methods via Language Models: A line of text-based KG inductive reasoning methods like BLP (Daza et al., 2021), KEPLER (Wang et al., 2021c), StATIK (Markowitz et al., 2022), RAILD (Gesese et al., 2022) rely on textual descriptions of entities and relations and use pre-trained language models (PLMs) like BERT (Devlin et al., 2019) to encode them. To seek higher prediction performance, current methods mostly prefer to fine-tune PLMs which requires high computational complexity and limits its generalizability to other KGs with different formats of textual descriptions. CSProm-KG (Chen et al., 2023) and PDKGC (Geng et al., 2023b) perform KG link prediction via a frozen PLM with only the prompts trained. However, the trainable prompts are tuned for a fixed relation vocabulary and cannot be generalized to inductive reasoning. Recent work (Zhu et al., 2023) verifies the reasoning performance of large language models by converting a query into textual names of the query entity and relation. We consider this family of methods to be orthogonal to our work. Our approach operates under the low-resource assumption that the graphs lack explicit entity descriptions.

K-shot	Model	FB:v1	FB:v2	FB:v3	FB:v4	WK:v1	WK:v2
3shot	NBFNet	0.084 (±0.000)	0.016 (±0.000)	0.012 (±0.000)	0.001 (±0.000)	0.011 (±0.000)	0.002 (±0.000)
3shot	REDGNN	0.041 (±0.000)	0.035 (±0.001)	0.056 (±0.001)	0.025 (±0.001)	0.001 (±0.000)	0.003 (±0.001)
3shot	InGram	0.071 (±0.006)	0.029 (±0.002)	0.062 (±0.007)	0.035 (±0.005)	0.051 (±0.013)	0.001 (±0.001)
3shot	DEqInGram	0.102 (±0.004)	0.035 (±0.005)	0.108 (±0.010)	0.038 (±0.006)	0.021 (±0.008)	0.009 (±0.001)
3shot	ISDEA	0.009 (±0.001)	0.005 (±0.001)	0.005 (±0.000)	0.005 (±0.000)	0.008 (±0.001)	0.003 (±0.000)
3shot	ULTRA(3g)	0.241 (±0.007)	0.212 (±0.004)	0.228 (±0.012)	0.186 (±0.009)	0.298 (±0.017)	0.050 (±0.010)
3shot	ULTRA(4g)	0.268 (±0.007)	0.211 (±0.008)	0.218 (±0.009)	0.196 (±0.010)	0.123 (±0.028)	0.041 (±0.010)
3shot	ULTRA(50g)	0.248 (±0.004)	0.209 (±0.007)	0.215 (±0.015)	0.197 (±0.005)	0.129 (±0.008)	0.037 (±0.012)
3shot	Our(Llama2-7B)	0.331 (±0.006)	0.290 (±0.017)	0.260 (±0.008)	0.232 (±0.008)	0.321 (±0.024)	0.052 (±0.008)
3shot	Our(Llama2-13B)	0.313 (±0.006)	0.289 (±0.013)	0.253 (±0.005)	0.225 (±0.009)	0.333 (±0.010)	0.053 (±0.007)
3shot	Our(Mistral-7B)	0.316 (±0.003)	0.294 (±0.012)	0.253 (±0.011)	0.231 (±0.005)	0.325 (±0.018)	0.053 (±0.008)
3shot	Our(GPT-3.5)	0.325 (±0.002)	0.286 (±0.011)	0.256 (±0.011)	0.230 (±0.008)	0.331 (±0.011)	0.054 (±0.009)
3shot	Our(GPT-4)	0.332 (±0.004)	0.290 (±0.011)	0.256 (±0.013)	0.237 (±0.004)	0.326 (±0.013)	0.058 (±0.008)
1shot	NBFNet	0.079 (±0.000)	0.015 (±0.000)	0.012 (±0.000)	0.001 (±0.000)	0.008 (±0.000)	0.002 (±0.000)
1shot	REDGNN	0.038 (±0.000)	0.034 (±0.001)	0.055 (±0.001)	0.020 (±0.001)	0.001 (±0.000)	0.002 (±0.000)
1shot	InGram	0.050 (±0.007)	0.032 (±0.003)	0.035 (±0.011)	0.025 (±0.006)	0.060 (±0.010)	0.001 (±0.001)
1shot	DEqInGram	0.078 (±0.011)	0.029 (±0.004)	0.082 (±0.009)	0.034 (±0.004)	0.019 (±0.003)	0.008 (±0.002)
1shot	ISDEA	0.008 (±0.001)	0.004 (±0.001)	0.004 (±0.001)	0.004 (±0.000)	0.007 (±0.002)	0.002 (±0.001)
1shot	ULTRA(3g)	0.182 (±0.011)	0.153 (±0.004)	0.159 (±0.009)	0.108 (±0.010)	0.221 (±0.020)	0.030 (±0.001)
1shot	ULTRA(4g)	0.190 (±0.008)	0.149 (±0.008)	0.156 (±0.012)	0.132 (±0.013)	0.051 (±0.009)	0.023 (±0.003)
1shot	ULTRA(50g)	0.180 (±0.010)	0.142 (±0.005)	0.148 (±0.016)	0.141 (±0.011)	0.062 (±0.029)	0.020 (±0.002)
1shot	Our(Llama2-7B)	0.280 (±0.013)	0.250 (±0.003)	0.228 (±0.013)	0.185 (±0.006)	0.259 (±0.029)	0.031 (±0.004)
1shot	Our(Llama2-13B)	0.270 (±0.008)	0.249 (±0.010)	0.221 (±0.010)	0.174 (±0.011)	0.275 (±0.027)	0.030 (±0.004)
1shot	Our(Mistral-7B)	0.281 (±0.012)	0.251 (±0.001)	0.224 (±0.008)	0.179 (±0.011)	0.267 (±0.023)	0.035 (±0.004)
1shot	Our(GPT-3.5)	0.273 (±0.008)	0.246 (±0.009)	0.228 (±0.004)	0.190 (±0.007)	0.259 (±0.023)	0.034 (±0.005)
1shot	Our(GPT-4)	0.281 (±0.011)	0.248 (±0.009)	0.226 (±0.005)	0.194 (±0.008)	0.262 (±0.029)	0.035 (±0.006)
0shot	NBFNet	0.079 (±0.000)	0.015 (±0.000)	0.012 (±0.000)	0.001 (±0.000)	0.008 (±0.000)	0.002 (±0.000)
0shot	REDGNN	0.038 (±0.000)	0.033 (±0.000)	0.054 (±0.000)	0.020 (±0.000)	0.001 (±0.000)	0.002 (±0.000)
0shot	InGram	0.009 (±0.000)	0.021 (±0.000)	0.025 (±0.000)	0.014 (±0.000)	0.041 (±0.000)	0.001 (±0.000)
0shot	DEqInGram	0.015 (±0.001)	0.020 (±0.001)	0.037 (±0.005)	0.024 (±0.003)	0.053 (±0.029)	0.009 (±0.001)
0shot	ISDEA	0.007 (±0.000)	0.001 (±0.000)	0.002 (±0.000)	0.003 (±0.000)	0.003 (±0.001)	0.001 (±0.001)
0shot	ULTRA(3g)	0.048 (±0.000)	0.035 (±0.000)	0.034 (±0.000)	0.028 (±0.000)	0.019 (±0.000)	0.006 (±0.000)
0shot	ULTRA(4g)	0.053 (±0.000)	0.041 (±0.000)	0.033 (±0.000)	0.030 (±0.000)	0.020 (±0.000)	0.006 (±0.000)
0shot	ULTRA(50g)	0.035 (±0.000)	0.024 (±0.000)	0.023 (±0.000)	0.029 (±0.000)	0.019 (±0.000)	0.006 (±0.000)
0shot	Our(Llama2-7B)	0.092 (±0.000)	0.099 (±0.000)	0.064 (±0.000)	0.067 (±0.000)	0.016 (±0.000)	0.008 (±0.000)
0shot	Our(Llama2-13B)	0.110 (±0.000)	0.097 (±0.000)	0.067 (±0.000)	0.059 (±0.000)	0.036 (±0.000)	0.008 (±0.000)
0shot	Our(Mistral-7B)	0.152 (±0.000)	0.139 (±0.000)	0.131 (±0.000)	0.097 (±0.000)	0.018 (±0.000)	0.007 (±0.000)
0shot	Our(GPT-3.5)	0.181 (±0.000)	0.166 (±0.000)	0.134 (±0.000)	0.120 (±0.000)	0.191 (±0.000)	0.011 (±0.000)
0shot	Our(GPT-4)	0.186 (±0.000)	0.168 (±0.000)	0.141 (±0.000)	0.108 (±0.000)	0.235 (±0.000)	0.013 (±0.000)

Table 10: Detailed inductive reasoning results (1), evaluated with MRR.

K-shot	Model	WK:v3	WK:v4	NL:v1	NL:v2	NL:v3	NL:v4
3shot	NBFNet	0.005 (±0.000)	0.002 (±0.000)	0.003 (±0.000)	0.019 (±0.000)	0.009 (±0.000)	0.002 (±0.000)
3shot	REDGNN	0.001 (±0.000)	0.001 (±0.000)	0.063 (±0.000)	0.106 (±0.000)	0.092 (±0.003)	0.014 (±0.000)
3shot	InGram	0.015 (±0.008)	0.016 (±0.005)	0.060 (±0.007)	0.071 (±0.014)	0.063 (±0.003)	0.075 (±0.009)
3shot	DEqInGram	0.012 (±0.007)	0.068 (±0.007)	0.120 (±0.014)	0.083 (±0.014)	0.103 (±0.010)	0.133 (±0.017)
3shot	ISDEA	0.008 (±0.001)	0.002 (±0.000)	0.019 (±0.005)	0.017 (±0.003)	0.017 (±0.005)	0.009 (±0.000)
3shot	ULTRA(3g)	0.135 (±0.018)	0.078 (±0.013)	0.197 (±0.028)	0.211 (±0.033)	0.209 (±0.010)	0.245 (±0.027)
3shot	ULTRA(4g)	0.137 (±0.020)	0.072 (±0.014)	0.192 (±0.022)	0.192 (±0.033)	0.181 (±0.021)	0.216 (±0.025)
3shot	ULTRA(50g)	0.139 (±0.023)	0.068 (±0.011)	0.169 (±0.029)	0.180 (±0.040)	0.150 (±0.031)	0.184 (±0.028)
3shot	Our(Llama2-7B)	0.137 (±0.024)	0.076 (±0.012)	0.225 (±0.023)	0.243 (±0.020)	0.215 (±0.011)	0.261 (±0.019)
3shot	Our(Llama2-13B)	0.137 (±0.022)	0.076 (±0.013)	0.217 (±0.021)	0.232 (±0.014)	0.223 (±0.013)	0.264 (±0.026)
3shot	Our(Mistral-7B)	0.139 (±0.025)	0.076 (±0.012)	0.223 (±0.027)	0.222 (±0.041)	0.229 (±0.017)	0.261 (±0.021)
3shot	Our(GPT-3.5)	0.140 (±0.023)	0.078 (±0.013)	0.219 (±0.012)	0.235 (±0.012)	0.233 (±0.014)	0.261 (±0.021)
3shot	Our(GPT-4)	0.141 (±0.020)	0.077 (±0.013)	0.229 (±0.021)	0.239 (±0.020)	0.230 (±0.012)	0.249 (±0.025)
1shot	NBFNet	0.002 (±0.000)	0.002 (±0.000)	0.004 (±0.000)	0.076 (±0.000)	0.007 (±0.000)	0.002 (±0.000)
1shot	REDGNN	0.001 (±0.000)	0.001 (±0.000)	0.047 (±0.001)	0.102 (±0.000)	0.071 (±0.002)	0.015 (±0.000)
1shot	InGram	0.006 (±0.002)	0.007 (±0.001)	0.060 (±0.006)	0.050 (±0.002)	0.047 (±0.002)	0.062 (±0.002)
1shot	DEqInGram	0.012 (±0.002)	0.019 (±0.003)	0.135 (±0.028)	0.073 (±0.022)	0.084 (±0.004)	0.120 (±0.012)
1shot	ISDEA	0.008 (±0.002)	0.001 (±0.000)	0.020 (±0.003)	0.015 (±0.004)	0.016 (±0.005)	0.009 (±0.000)
1shot	ULTRA(3g)	0.100 (±0.018)	0.023 (±0.001)	0.150 (±0.018)	0.155 (±0.035)	0.153 (±0.016)	0.199 (±0.017)
1shot	ULTRA(4g)	0.100 (±0.020)	0.020 (±0.002)	0.160 (±0.012)	0.137 (±0.031)	0.100 (±0.011)	0.118 (±0.011)
1shot	ULTRA(50g)	0.099 (±0.023)	0.020 (±0.002)	0.137 (±0.014)	0.113 (±0.009)	0.093 (±0.003)	0.140 (±0.022)
1shot	Our(Llama2-7B)	0.103 (±0.024)	0.023 (±0.003)	0.207 (±0.030)	0.189 (±0.016)	0.174 (±0.004)	0.222 (±0.027)
1shot	Our(Llama2-13B)	0.100 (±0.024)	0.026 (±0.002)	0.196 (±0.005)	0.192 (±0.022)	0.176 (±0.003)	0.215 (±0.015)
1shot	Our(Mistral-7B)	0.104 (±0.025)	0.023 (±0.001)	0.193 (±0.014)	0.184 (±0.037)	0.171 (±0.003)	0.206 (±0.028)
1shot	Our(GPT-3.5)	0.103 (±0.023)	0.025 (±0.003)	0.196 (±0.017)	0.193 (±0.018)	0.169 (±0.005)	0.223 (±0.014)
1shot	Our(GPT-4)	0.104 (±0.023)	0.024 (±0.002)	0.203 (±0.021)	0.193 (±0.042)	0.171 (±0.004)	0.200 (±0.028)
0shot	NBFNet	0.002 (±0.000)	0.002 (±0.000)	0.004 (±0.000)	0.076 (±0.000)	0.007 (±0.000)	0.002 (±0.000)
0shot	REDGNN	0.001 (±0.000)	0.001 (±0.000)	0.046 (±0.000)	0.102 (±0.000)	0.062 (±0.000)	0.015 (±0.000)
0shot	InGram	0.001 (±0.000)	0.001 (±0.000)	0.032 (±0.000)	0.042 (±0.000)	0.015 (±0.000)	0.055 (±0.000)
0shot	DEqInGram	0.004 (±0.001)	0.009 (±0.001)	0.083 (±0.007)	0.067 (±0.006)	0.055 (±0.006)	0.049 (±0.001)
0shot	ISDEA	0.003 (±0.000)	0.000 (±0.000)	0.006 (±0.000)	0.008 (±0.001)	0.003 (±0.000)	0.006 (±0.002)
0shot	ULTRA(3g)	0.013 (±0.000)	0.004 (±0.000)	0.037 (±0.000)	0.021 (±0.000)	0.024 (±0.000)	0.022 (±0.000)
0shot	ULTRA(4g)	0.013 (±0.000)	0.003 (±0.000)	0.034 (±0.000)	0.024 (±0.000)	0.024 (±0.000)	0.022 (±0.000)
0shot	ULTRA(50g)	0.013 (±0.000)	0.003 (±0.000)	0.033 (±0.000)	0.027 (±0.000)	0.024 (±0.000)	0.017 (±0.000)
0shot	Our(Llama2-7B)	0.013 (±0.000)	0.006 (±0.000)	0.103 (±0.000)	0.125 (±0.000)	0.058 (±0.000)	0.102 (±0.000)
0shot	Our(Llama2-13B)	0.015 (±0.000)	0.011 (±0.000)	0.113 (±0.000)	0.128 (±0.000)	0.094 (±0.000)	0.100 (±0.000)
0shot	Our(Mistral-7B)	0.014 (±0.000)	0.009 (±0.000)	0.101 (±0.000)	0.082 (±0.000)	0.082 (±0.000)	0.067 (±0.000)
0shot	Our(GPT-3.5)	0.017 (±0.000)	0.009 (±0.000)	0.114 (±0.000)	0.125 (±0.000)	0.085 (±0.000)	0.110 (±0.000)
0shot	Our(GPT-4)	0.017 (±0.000)	0.009 (±0.000)	0.091 (±0.000)	0.095 (±0.000)	0.089 (±0.000)	0.116 (±0.000)

Table 11: Detailed inductive reasoning results (2), evaluated with MRR.

K-shot	Model	FB:v1	FB:v2	FB:v3	FB:v4	WK:v1	WK:v2
3shot	NBFNet	0.092 (±0.000)	0.019 (±0.000)	0.013 (±0.000)	0.000 (±0.000)	0.033 (±0.000)	0.001 (±0.000)
3shot	REDGNN	0.068 (±0.001)	0.066 (±0.001)	0.110 (±0.001)	0.065 (±0.001)	0.000 (±0.000)	0.003 (±0.002)
3shot	InGram	0.098 (±0.007)	0.062 (±0.005)	0.114 (±0.004)	0.070 (±0.007)	0.070 (±0.008)	0.002 (±0.001)
3shot	DEqInGram	0.076 (±0.006)	0.009 (±0.002)	0.070 (±0.012)	0.011 (±0.004)	0.006 (±0.005)	0.002 (±0.001)
3shot	ISDEA	0.018 (±0.001)	0.013 (±0.000)	0.009 (±0.001)	0.008 (±0.001)	0.014 (±0.002)	0.003 (±0.001)
3shot	ULTRA(3g)	0.359 (±0.008)	0.334 (±0.009)	0.335 (±0.015)	0.319 (±0.017)	0.430 (±0.004)	0.092 (±0.031)
3shot	ULTRA(4g)	0.404 (±0.023)	0.358 (±0.018)	0.327 (±0.011)	0.316 (±0.019)	0.138 (±0.031)	0.063 (±0.025)
3shot	ULTRA(50g)	0.368 (±0.016)	0.354 (±0.008)	0.329 (±0.015)	0.310 (±0.015)	0.157 (±0.010)	0.055 (±0.026)
3shot	Our(Llama2-7B)	0.513 (±0.012)	0.480 (±0.014)	0.408 (±0.017)	0.377 (±0.023)	0.461 (±0.016)	0.108 (±0.021)
3shot	Our(Llama2-13B)	0.492 (±0.008)	0.468 (±0.015)	0.405 (±0.019)	0.358 (±0.022)	0.463 (±0.021)	0.116 (±0.020)
3shot	Our(Mistral-7B)	0.494 (±0.005)	0.467 (±0.017)	0.402 (±0.019)	0.367 (±0.023)	0.461 (±0.007)	0.115 (±0.020)
3shot	Our(GPT-3.5)	0.505 (±0.005)	0.471 (±0.014)	0.408 (±0.016)	0.380 (±0.019)	0.455 (±0.002)	0.115 (±0.018)
3shot	Our(GPT-4)	0.517 (±0.012)	0.473 (±0.014)	0.404 (±0.014)	0.375 (±0.023)	0.456 (±0.001)	0.118 (±0.028)
1shot	NBFNet	0.087 (±0.000)	0.018 (±0.000)	0.013 (±0.000)	0.000 (±0.000)	0.019 (±0.000)	0.001 (±0.000)
1shot	REDGNN	0.063 (±0.000)	0.064 (±0.000)	0.103 (±0.000)	0.054 (±0.001)	0.002 (±0.000)	0.001 (±0.000)
1shot	InGram	0.069 (±0.008)	0.057 (±0.003)	0.062 (±0.022)	0.049 (±0.010)	0.080 (±0.023)	0.003 (±0.002)
1shot	DEqInGram	0.063 (±0.011)	0.007 (±0.002)	0.050 (±0.010)	0.013 (±0.003)	0.004 (±0.002)	0.002 (±0.001)
1shot	ISDEA	0.022 (±0.002)	0.008 (±0.001)	0.007 (±0.001)	0.006 (±0.000)	0.011 (±0.004)	0.002 (±0.001)
1shot	ULTRA(3g)	0.270 (±0.010)	0.256 (±0.010)	0.235 (±0.014)	0.204 (±0.020)	0.249 (±0.031)	0.042 (±0.003)
1shot	ULTRA(4g)	0.296 (±0.011)	0.273 (±0.016)	0.230 (±0.013)	0.221 (±0.011)	0.043 (±0.020)	0.025 (±0.002)
1shot	ULTRA(50g)	0.275 (±0.009)	0.254 (±0.012)	0.230 (±0.025)	0.235 (±0.013)	0.065 (±0.035)	0.023 (±0.004)
1shot	Our(Llama2-7B)	0.442 (±0.022)	0.417 (±0.013)	0.363 (±0.023)	0.298 (±0.005)	0.371 (±0.033)	0.056 (±0.017)
1shot	Our(Llama2-13B)	0.445 (±0.011)	0.404 (±0.010)	0.350 (±0.014)	0.280 (±0.021)	0.384 (±0.037)	0.054 (±0.012)
1shot	Our(Mistral-7B)	0.441 (±0.013)	0.404 (±0.004)	0.359 (±0.006)	0.295 (±0.013)	0.381 (±0.029)	0.061 (±0.019)
1shot	Our(GPT-3.5)	0.438 (±0.022)	0.417 (±0.011)	0.369 (±0.005)	0.324 (±0.007)	0.381 (±0.029)	0.058 (±0.017)
1shot	Our(GPT-4)	0.451 (±0.012)	0.409 (±0.018)	0.370 (±0.007)	0.319 (±0.006)	0.377 (±0.040)	0.063 (±0.021)
0shot	NBFNet	0.087 (±0.000)	0.018 (±0.000)	0.013 (±0.000)	0.000 (±0.000)	0.019 (±0.000)	0.001 (±0.000)
0shot	REDGNN	0.063 (±0.000)	0.064 (±0.000)	0.103 (±0.000)	0.054 (±0.000)	0.002 (±0.000)	0.001 (±0.000)
0shot	InGram	0.017 (±0.000)	0.043 (±0.000)	0.058 (±0.000)	0.027 (±0.000)	0.054 (±0.000)	0.001 (±0.000)
0shot	DEqInGram	0.003 (±0.002)	0.004 (±0.001)	0.013 (±0.005)	0.007 (±0.001)	0.023 (±0.031)	0.001 (±0.001)
0shot	ISDEA	0.017 (±0.000)	0.001 (±0.000)	0.002 (±0.000)	0.002 (±0.000)	0.002 (±0.002)	0.000 (±0.001)
0shot	ULTRA(3g)	0.145 (±0.000)	0.128 (±0.000)	0.101 (±0.000)	0.098 (±0.000)	0.019 (±0.000)	0.008 (±0.000)
0shot	ULTRA(4g)	0.155 (±0.000)	0.148 (±0.000)	0.101 (±0.000)	0.107 (±0.000)	0.018 (±0.000)	0.010 (±0.000)
0shot	ULTRA(50g)	0.103 (±0.000)	0.100 (±0.000)	0.063 (±0.000)	0.100 (±0.000)	0.023 (±0.000)	0.012 (±0.000)
0shot	Our(Llama2-7B)	0.193 (±0.000)	0.203 (±0.000)	0.133 (±0.000)	0.153 (±0.000)	0.029 (±0.000)	0.010 (±0.000)
0shot	Our(Llama2-13B)	0.209 (±0.000)	0.179 (±0.000)	0.136 (±0.000)	0.121 (±0.000)	0.078 (±0.000)	0.011 (±0.000)
0shot	Our(Mistral-7B)	0.254 (±0.000)	0.254 (±0.000)	0.222 (±0.000)	0.195 (±0.000)	0.043 (±0.000)	0.012 (±0.000)
0shot	Our(GPT-3.5)	0.300 (±0.000)	0.267 (±0.000)	0.240 (±0.000)	0.225 (±0.000)	0.304 (±0.000)	0.020 (±0.000)
0shot	Our(GPT-4)	0.284 (±0.000)	0.251 (±0.000)	0.256 (±0.000)	0.213 (±0.000)	0.350 (±0.000)	0.024 (±0.000)

Table 12: Detailed inductive reasoning results (1), evaluated with Hits@10.

K-shot	Model	WK:v3	WK:v4	NL:v1	NL:v2	NL:v3	NL:v4
3shot	NBFNet	0.000 (±0.000)	0.000 (±0.000)	0.000 (±0.000)	0.014 (±0.000)	0.005 (±0.000)	0.000 (±0.000)
3shot	REDGNN	0.000 (±0.000)	0.000 (±0.000)	0.081 (±0.000)	0.171 (±0.000)	0.136 (±0.007)	0.025 (±0.000)
3shot	InGram	0.022 (±0.011)	0.023 (±0.004)	0.122 (±0.029)	0.124 (±0.007)	0.138 (±0.014)	0.149 (±0.014)
3shot	DEqInGram	0.005 (±0.003)	0.052 (±0.006)	0.068 (±0.009)	0.046 (±0.017)	0.043 (±0.007)	0.074 (±0.013)
3shot	ISDEA	0.012 (±0.004)	0.003 (±0.000)	0.035 (±0.009)	0.031 (±0.006)	0.035 (±0.004)	0.019 (±0.001)
3shot	ULTRA(3g)	0.172 (±0.031)	0.114 (±0.007)	0.275 (±0.058)	0.291 (±0.048)	0.324 (±0.046)	0.369 (±0.018)
3shot	ULTRA(4g)	0.176 (±0.032)	0.092 (±0.010)	0.258 (±0.043)	0.255 (±0.020)	0.265 (±0.027)	0.300 (±0.017)
3shot	ULTRA(50g)	0.166 (±0.033)	0.086 (±0.009)	0.227 (±0.040)	0.220 (±0.037)	0.219 (±0.041)	0.262 (±0.025)
3shot	Our(Llama2-7B)	0.186 (±0.029)	0.111 (±0.007)	0.335 (±0.063)	0.372 (±0.031)	0.335 (±0.041)	0.394 (±0.001)
3shot	Our(Llama2-13B)	0.186 (±0.032)	0.110 (±0.008)	0.320 (±0.056)	0.367 (±0.029)	0.354 (±0.042)	0.403 (±0.017)
3shot	Our(Mistral-7B)	0.188 (±0.030)	0.108 (±0.006)	0.323 (±0.049)	0.333 (±0.062)	0.354 (±0.030)	0.394 (±0.007)
3shot	Our(GPT-3.5)	0.187 (±0.032)	0.113 (±0.009)	0.342 (±0.027)	0.351 (±0.066)	0.355 (±0.049)	0.411 (±0.001)
3shot	Our(GPT-4)	0.188 (±0.036)	0.112 (±0.011)	0.330 (±0.062)	0.352 (±0.045)	0.338 (±0.060)	0.398 (±0.017)
1shot	NBFNet	0.000 (±0.000)	0.000 (±0.000)	0.000 (±0.000)	0.071 (±0.000)	0.004 (±0.000)	0.000 (±0.000)
1shot	REDGNN	0.004 (±0.000)	0.000 (±0.000)	0.063 (±0.003)	0.164 (±0.000)	0.099 (±0.004)	0.029 (±0.000)
1shot	InGram	0.014 (±0.005)	0.009 (±0.001)	0.096 (±0.013)	0.082 (±0.006)	0.085 (±0.007)	0.116 (±0.015)
1shot	DEqInGram	0.004 (±0.004)	0.012 (±0.003)	0.096 (±0.031)	0.043 (±0.024)	0.035 (±0.005)	0.066 (±0.005)
1shot	ISDEA	0.011 (±0.002)	0.001 (±0.000)	0.026 (±0.005)	0.027 (±0.007)	0.023 (±0.012)	0.018 (±0.002)
1shot	ULTRA(3g)	0.101 (±0.026)	0.029 (±0.003)	0.216 (±0.031)	0.194 (±0.033)	0.223 (±0.016)	0.255 (±0.024)
1shot	ULTRA(4g)	0.101 (±0.026)	0.020 (±0.002)	0.183 (±0.010)	0.171 (±0.036)	0.151 (±0.003)	0.160 (±0.012)
1shot	ULTRA(50g)	0.096 (±0.025)	0.020 (±0.002)	0.185 (±0.032)	0.176 (±0.033)	0.155 (±0.003)	0.186 (±0.018)
1shot	Our(Llama2-7B)	0.116 (±0.030)	0.029 (±0.004)	0.295 (±0.022)	0.277 (±0.033)	0.247 (±0.014)	0.311 (±0.028)
1shot	Our(Llama2-13B)	0.119 (±0.030)	0.034 (±0.002)	0.272 (±0.011)	0.279 (±0.031)	0.256 (±0.009)	0.312 (±0.015)
1shot	Our(Mistral-7B)	0.119 (±0.032)	0.031 (±0.001)	0.270 (±0.012)	0.259 (±0.049)	0.268 (±0.009)	0.285 (±0.027)
1shot	Our(GPT-3.5)	0.118 (±0.029)	0.035 (±0.003)	0.292 (±0.029)	0.291 (±0.050)	0.266 (±0.018)	0.322 (±0.025)
1shot	Our(GPT-4)	0.118 (±0.022)	0.035 (±0.004)	0.297 (±0.033)	0.267 (±0.026)	0.260 (±0.009)	0.300 (±0.015)
0shot	NBFNet	0.000 (±0.000)	0.000 (±0.000)	0.000 (±0.000)	0.071 (±0.000)	0.004 (±0.000)	0.000 (±0.000)
0shot	REDGNN	0.000 (±0.000)	0.000 (±0.000)	0.060 (±0.000)	0.164 (±0.000)	0.088 (±0.000)	0.029 (±0.000)
0shot	InGram	0.002 (±0.000)	0.002 (±0.000)	0.070 (±0.000)	0.070 (±0.000)	0.029 (±0.000)	0.081 (±0.000)
0shot	DEqInGram	0.000 (±0.001)	0.001 (±0.001)	0.040 (±0.018)	0.032 (±0.002)	0.022 (±0.005)	0.020 (±0.001)
0shot	ISDEA	0.004 (±0.001)	0.000 (±0.000)	0.009 (±0.003)	0.014 (±0.004)	0.003 (±0.002)	0.012 (±0.005)
0shot	ULTRA(3g)	0.013 (±0.000)	0.004 (±0.000)	0.113 (±0.000)	0.068 (±0.000)	0.061 (±0.000)	0.060 (±0.000)
0shot	ULTRA(4g)	0.013 (±0.000)	0.004 (±0.000)	0.096 (±0.000)	0.071 (±0.000)	0.059 (±0.000)	0.061 (±0.000)
0shot	ULTRA(50g)	0.013 (±0.000)	0.004 (±0.000)	0.089 (±0.000)	0.077 (±0.000)	0.058 (±0.000)	0.046 (±0.000)
0shot	Our(Llama2-7B)	0.016 (±0.000)	0.007 (±0.000)	0.180 (±0.000)	0.190 (±0.000)	0.095 (±0.000)	0.161 (±0.000)
0shot	Our(Llama2-13B)	0.016 (±0.000)	0.016 (±0.000)	0.191 (±0.000)	0.184 (±0.000)	0.155 (±0.000)	0.148 (±0.000)
0shot	Our(Mistral-7B)	0.017 (±0.000)	0.011 (±0.000)	0.172 (±0.000)	0.156 (±0.000)	0.143 (±0.000)	0.119 (±0.000)
0shot	Our(GPT-3.5)	0.030 (±0.000)	0.014 (±0.000)	0.178 (±0.000)	0.178 (±0.000)	0.153 (±0.000)	0.177 (±0.000)
0shot	Our(GPT-4)	0.031 (±0.000)	0.018 (±0.000)	0.183 (±0.000)	0.140 (±0.000)	0.147 (±0.000)	0.191 (±0.000)

Table 13: Detailed inductive reasoning results (2), evaluated with Hits@10.