REPANA: Reasoning Path Navigated Program Induction for Transferable Reasoning over Heterogeneous Knowledge Bases

Jinxin Liu[1], **Shulin Cao**[1], **Jiajie Zhang**[1], **Xiaoyin Che**[2], **Weichuan Liu**[2],
Liangming Pan[3], **Lei Hou**[1], **Juanzi Li**[1,2]
[1] Department of Computer Science and Technology, BNRist;
[2] THU - Siemens Ltd., China Joint Research Center for Industrial Intelligence and IoT;
Tsinghua University, Beijing, China
[3] School of Computer Science, Peking University, Beijing, China

***Abstract*—Program induction is a typical approach that helps Large Language Models (LLMs) in complex knowledge-intensive question answering over knowledge bases (KBs) to alleviate the hallucination of LLMs. However, accurate program induction requires extensive high-quality parallel data for a specific KB, which is scarce for low-resource KBs. Moreover, the heterogeneity of questions and KB schemas limits the transferability of models trained on a single dataset. To this end, we propose REPANA, a reasoning path navigated program induction framework that enables LLMs to reason over heterogeneous KBs. We decouple the program induction into perceiving the KB and mapping questions to program sketches. Accordingly, our framework consists of (1) an LLM-based navigator, which retrieves reasoning paths of the input question from the given KB; (2) and a KB-agnostic parser trained on multiple heterogeneous datasets, which takes the retrieved paths and the question as input and generates the corresponding program. Experiments show that REPANA exhibits strong generalization and transferability. It can directly perform inference on datasets not seen during training, outperforming other state-of-the-art low-resource methods, even approaching the performance of supervised methods.**

***Index Terms*—Knowledge base Question Answering, Low-Resource NLP, Parameter-Efficient Fine-Tuning**

I. Introduction

Recently, incorporating knowledge bases (KBs) as external knowledge to augment large language models (LLMs) [1] in knowledge-intensive question answering has become a typical approach [2]–[4] to address the challenge of hallucination [5], which describes the tendency that LLMs often confidently make up that is factually incorrect content.

To leverage KBs for question answering, there are two typical methods. The first is program induction (PI) [6] that translates a given question into an interpretable logical form, such as KoPL [7] or SPARQL [8], which is executable against the KB to obtain the answer. To generate more accurate programs, some approaches [9] retrieve relevant information from the KB before generation, others [10] introduce question–program pairs as demonstrations, or employ instruction tuning [11] to train the model. these works typically require training on a single KB with a large number of question-program pairs, whose collection demands substantial human labor. The second is the agent-based method [2], [12], [13] that uses LLMs to dynamically explore the knowledge graph step by step with predefined reasoning actions and tools that can interact with KBs. However, these methods may be restricted by the incompleteness of the action space, unable to answer questions that require complex reasoning operations. Although [14] defines a comprehensive toolbox, learning to use complex tools still relies on large amounts of training data. More importantly, agent-based methods struggle with questions that lack a topic entity, since these questions offer no starting point for the LLM to initiate graph search.

In this paper, we propose REPANA, a reasoning path navigated PI framework that enables LLMs to reason over heterogeneous KBs, aiming to improve data efficiency and cross-KB generalization while retaining the interpretability and execution efficiency of symbolic programs. Inspired by recent studies [15], we hypothesize that general reasoning ability—*i.e.,* mapping a question to a composition of program functions—is transferable across KBs. Unlike existing PI models that jointly learn the KB schema and the question-to-program mapping from parallel data, our approach decouples the reasoning process on KB into two stages, which corresponds to two key modules in the framework. The first is the LLM-based **KB navigator** that locates the reasoning path corresponding to the input question. The reasoning path provides the necessary program arguments, such as relation labels, allowing the system to capture the local schema of the KB. The second is the **KB-agnostic parser** trained on rich-resource KB, which primarily learns program syntax and grammar as well as how to map questions to program sketches with the guidance of the extracted reasoning path.

Through this novel two-stage design, we ensure retrieval efficiency and accuracy, reducing noise while enabling reasoning on low-resource KBs without additional training. Specifically, in the first stage, we design an LLM-based KB-walk search strategy similar to beam search. We first select a root based on the topic entity or concept in the question. Starting from

Corresponding author: houlei@tsinghua.edu.cn

the root, the navigator selects the most relevant relations in each walking step and returns the most viable path through backtracking. In the second stage, to prevent the parser from memorizing the KB schema and increase the diversity of training data, we expand the rich-resource KB into more diverse training data through paraphrasing. Then we perform instruction tuning on these data, which takes both the question and the retrieved reasoning path as input, selects the necessary elements from the reasoning path as arguments, and generate the final program. The LLM-based KB navigator is training-free and the KB-agnostic parser is trained once on a resource-rich KB, therefore REPANA effectively addresses the challenge of transferring to new, unseen low-resource KBs.

Extensive experiments demonstrate that REPANA outperforms state-of-the-art(SoTA) low-resource PI methods with up to 20 times smaller backbone model, even approaching the performance of supervised methods. Our contributions in this paper include: (1) proposing REPANA, a novel reasoning path navigated program induction framework for heterogeneous low-resource datasets; (2) designing a novel two-stage generation strategy that utilizes a retrieved reasoning path from KB to train a KB-agnostic parser for program induction; (3) demonstrating the effectiveness of our decoupled two-stage generation strategy through extensive experiments.

II. Related Work

A. Knowledge Base Question Answering

Knowledge Base Question Answering (KBQA) aims to answer natural language questions based on fact triples stored in the KB, such as Wikidata [16] and Freebase [17]. Typical methods for solving KBQA can be broadly divided into two groups: (1) program induction-based method, which converts questions to executable logical forms called program. The programs are usually generated by step-by-step graph search [2], [6], [18] or by sequence-to-sequence model trained with parallel data [9], [15], [19]; (2) information retrieval-based method, which usually outputs the answer by retrieving triples and subgraphs related to the question from KB or embedded memory [20]–[22]. Recent works [12], [18] that leverage LLMs as agents to explore the KB also belong to this group. They search the KB by prompting the LLMs step by step for the next action. However, they can only handle a limited range of questions with their pre-defined actions, and cannot easily adapt between different KBs.

B. Low-resource Program Induction

One line of work is utilizing the in-context-learning ability of LLMs to perform few-shot program generation [10], [13], [23], but their performance usually is limited by the context window. They also face challenges in distinguishing similar schema items in the KB, causing models to overly rely on post-processing steps like relation linking. A variation [24] is using LLMs to few-shot generate question given the program, then training a smaller model with the generated pseudo pairs. But their programs either come from existing datasets or templates, leading to insufficient diversity and scalability.

The other line is program transfer method, which leverages the annotation from rich-resource KB to aid program induction for low-resource KB. ProgramTransfer [15] proposes a two-stage parsing framework that first generates the program sketch, then fills in the rest arguments by searching the KB. However, due to the heterogeneity, it performs poorly without fine-tuning using annotated data from low-resource KB. KB-plugin [25] proposes a plug-and-play framework that encodes the KB schema into the parameters of a LoRA [26] module. But parameterizing the KB may introduce additional errors and result in a loss of interpretability.

We follow the second line of work, aiming to address the challenge of transferability, interpretability, and accuracy at the same time.

III. Preliminary

In this section, we introduce the formal definition of the knowledge base and then formulate our task.

Knowledge Base (KB). A knowledge base can be formally described by $\mathcal{G} = \{\mathcal{E}, \mathcal{C}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$, $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{T}$ denote the set of entities, concepts, relations and triples, respectively. Each entity $e \in \mathcal{E}$ is assigned a unique ID and belongs to one or more concepts $c \in \mathcal{C}$. The relation set $\mathcal{R}$ contains the special relation $r_e =$"`instanceOf`", $r_c =$ "`subClassOf`" and the general relation set $R_l = \{r_l\}$. Given $\mathcal{E}$, $\mathcal{C}$ and $\mathcal{R}$, $\mathcal{T}$ can be divided into three subsets: (1) "`instanceOf`" triple set $\mathcal{T}_e = \{(e, r_e, c) | e \in \mathcal{E}, c \in \mathcal{C}\}$; (2) "`subClassOf`" triple set $\mathcal{T}_c = \{(c_i, r_c, c_j) | c_i, c_i \in \mathcal{C}\}$; (3) general relation set $\mathcal{T}_l = \{(e_i, r_l, e_j) | e_i, e_j \in \mathcal{E}\}$.

Program. In this work, we choose KoPL [7] as the program language, for it is well modularized and LLM-friendly [27]. KoPL is composed of symbolic functions with arguments arranged in tree structure. Each function defines a fundamental operation in KB. This tree $y = \langle f_1(arg_1), \cdots, f_i(arg_i), \cdots, f_{|y|}(arg_{|y|}) \rangle$ can be serialized with post-order traversal, where $f_i \in \mathcal{F}, arg_i \in \mathcal{E} \cup \mathcal{C} \cup \mathcal{R}_l \cup \{\emptyset\}$

IV. Method

In this section, we introduce the main components of REPANA framework and how they work together.

The key challenge in transferable reasoning is that parsers trained on a certain KB tend to overfit to the underlying schema, which hinders their generalization to unseen KBs. To mitigate this issue, we design two independent modules for KB awareness and logical form generation, forming a two-stage retrieval–generation paradigm, as shown in Fig. 1.

The framework generally follows the retrieve-and-generate paradigm. In the **training phase** of the KB-agnostic parser, we first employ the KB navigator module to extract the reasoning path p_S of the input question q_S from the source KB. To increase the diversity of training data, we apply n rounds of paraphrasing to both the questions and the schema items in the source KB (see Section IV-B1). This results in n versions of the dataset, $Q^S = \{Q^{S_1}, Q^{S_2}, \cdots, Q^{S_n}\}$, each paired with its corresponding reasoning path p_{S_i} extracted from the expanded KBs $KB^S = \{KB^{S_1}, KB^{S_2}, \cdots, KB^{S_n}\}$. We then merge

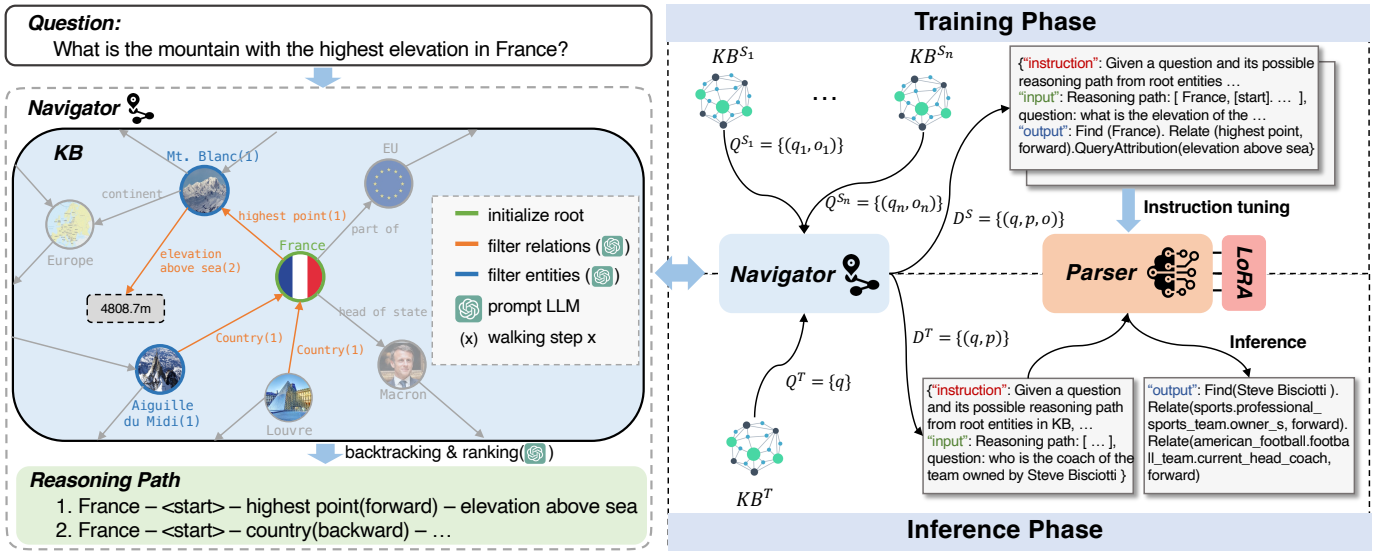

Fig. 1: An illustration of the training and inference of REPANA framework.

these variants to construct an instruction training set $D^S = \{(q, p, o)\}$. After instruction tuning the KB-agnostic parser on this mixed dataset, the model is ready to perform inference on the target low-resource KB KB^T. In the **inference phase**, the framework first retrieves the reasoning path p_T from the target KB^T. It then feeds the input question q_T and the retrieved path p_T, along with instructions, into the parser, which finally generates an executable program for KB^T.

In the following sections, we present the implementation details of the main components of our framework: KB Navigator (Section IV-A) and KB-agnostic parser (Section IV-B).

A. KB Navigator

Given a question, the KB navigator leverages its underlying KB to identify the corresponding reasoning paths. We design a KB-walk search strategy based on two observations: (1) despite schema differences, all KBs share a graph-structured organization, enabling a unified graph-walk algorithm; and (2) LLMs effectively select question-relevant relations from candidate sets without additional fine-tuning, making them suitable for unsupervised navigation.

1) Root Entity Selection: For methods that leverage LLMs to explore and reason on the knowledge graph step by step, it is necessary to select one or more root entities as the starting points. For question that contains topic entities, we directly use them as the root entities the same as previous methods. such as "Mt.Blanc" in question "How high is Mt. Blanc". However, this simple solution are unable to handle questions that do not contain a topic entity.

For questions without explicit topic entities, we sample root entities from those associated with the concept mentioned in the question. For example, for "Which mountain is the highest in the world?", entities under the concept mountain (e.g., Mount Fuji) can serve as pseudo-topic entities for exploration. This is based on the observation that conceptually similar entities tend to share relations. Since our framework focuses on retrieving relation names for the reasoning path—rather than identifying intermediate entities or the final answer—we can use a pseudo-topic entity as the root. In practice, given a concept c, we sample the entity e_r from $\{e \mid (e, r_e, c)\}$ that maximizes coverage of relations.

2) Reasoning Path Construction: We first summarize common characteristics of most knowledge bases (KBs). Following the formalization in Section III, we define three basic knowledge elements in a KB: entities, concepts, and relations. These elements largely follow their standard semantics, except that we use the term "relation" in a broader sense to cover both inter-entity relations (e.g., part of) and attribute–value associations (e.g., population), which we uniformly treat as relations throughout this paper.

We use entities e and relations r to form the core structure of a reasoning path. A reasoning path is denoted as $p_r = [e_r, \langle start \rangle, r_1, \cdots, r_k]$, where e_r is the root entity and k is the walk length. Depending on the question, a concept c may be appended to p_r to facilitate parsing. When multiple root entities $e_r{}_{r=1}^{m}$ exist, the KB navigator returns a set of paths $\mathbf{p} = p_r$, one per root entity. These paths may partially overlap, and all are passed to the parser in the next stage.

3) KB-walk Search Process: The KB-walk process consists of four steps: **initialize, filter relations, filter entities, and backtrack & rank**. Steps 2 and 3 are iterated for up to k rounds, where k is the maximum walk length. The process may terminate early if the same relation is repeatedly selected.
Initialize. The KB navigator initializes the root entities $\{e_r\}_{r=1}^{m}$ as described in Section IV-A1. These roots are obtained from topic entities or concepts provided by the dataset. While off-the-shelf named entity recognition models can be used, entity extraction is not the focus of this work.
Filter relations. This step explores candidate relations for advancing the walk. In the i-th round, the start entities are

denoted as $E_i = \{e_{1,i}, \ldots, e_{b,i}\}$, where $b = m$ when $i = 1$ and otherwise equals the beam size. We query the KB to retrieve all incoming and outgoing relations associated with each entity in E_i, forming a candidate relation set R_i. Given the question and E_i, the LLM selects up to b relations (or returns no answer), yielding a filtered set F_i. In Fig 1, $E_1 = \{\text{France}\}$, $R_1 = \{(\text{highest point, part of, head of state, country})\}$, and $F_i = \{\text{highest point, country}\}$.

Filter entities. This step advances the walk along the selected relations F_i. Starting from E_i, we query the KB to obtain candidate target entities for each relation in F_i, forming buckets of entities $\hat{E}$. One entity is selected from each bucket to construct E_{i+1} for the next round. To reduce inference cost, we can randomly select entities instead of using LLMs, assuming entities reached by the same relation share similar types and relational patterns. This simplification is effective since intermediate entities are not explicitly included in the reasoning path. In Fig 1, the $\hat{E} = \{(\text{Mt. Blanc}),(\text{Louvre, Aiguille du Midi})\}$ and $E_{i+1} = \{\text{Mt. Blanc}, \cdots\}$.

Backtrack & rank. Finally, we backtrack from the leaf nodes to the root entities to collect reasoning paths of varying lengths. All candidate paths are then ranked by the LLM according to their relevance to the question. Relation directions are preserved and explicitly tagged in each path.

B. KB-agnostic Parser

1) Training Data Expansion: To increase training diversity, we expand the dataset into n paraphrased variants. Both the input questions and the schema items in the output programs are paraphrased. Given a data triple (q, p, o)—for example, q = "Height of Mt. Blanc?", p = [Mt. Blanc, $\langle start \rangle$, elevation above sea], and o = Find(Mt. Blanc).QueryAttribution(elevation above sea)—we first prompt the LLM to paraphrase relation names and record the mappings between original and paraphrased relations (e.g., "elevation above sea" $\rightarrow$ "altitude"). We then replace the original relations in both p and o, yielding p' and o', and paraphrase the question to obtain a new triple (q', p', o').

In this way, we expand the original KB into n variations $KB^S = \left\{KB^{S_1}, KB^{S_2}, \cdots, KB^{S_n}\right\}$ along with dataset Q^S. By performing instruction tuning on these data, the parser is encouraged to focus on the program sketches (*i.e.,* function names and their composition) rather than memorizing specific schema items from the source KB.

2) Navigation-Guided Instruction Tuning: To mitigate schema overfitting and improve cross-KB transferability, we employ a navigation-guided instruction tuning strategy in which reasoning paths are explicitly provided as part of the input. Although these paths may contain noise, the training objective encourages the parser to denoise and recover the correct executable program.

As described above, we collect n paraphrased variants of each question together with their corresponding reasoning paths from paraphrased KBs, forming the dataset $D^S = \{(q, p, o)\}$. For instruction tuning, entity identifiers (e.g., `m.0f8l9c`) are converted into human-readable names (e.g., `France`). All samples are then normalized into a unified format, where q and p are placed in the `input` field and o in the `output` field, as illustrated in Fig. 1. The instruction prompt is kept consistent across training and evaluation.

3) Parameter Efficient Fine-tuning: We adopt parameter-efficient fine-tuning using LoRA [26], which introduces a small number of trainable parameters via low-rank adaptation modules. Specifically, for a weight matrix $W_i \in \mathbb{R}^{m \times n}$ in the backbone LLM, LoRA modifies the forward computation from $h = W_i x$ to $h = (W_i + A_i B_i)x$, where $A_i \in \mathbb{R}^{m \times r}$ and $B_i \in \mathbb{R}^{r \times n}$ with $r \ll \min(m, n)$. During training, W_i is frozen and only A_i and B_i are updated. This design substantially reduces training cost while enabling the use of larger backbone models.

4) Inference: During inference, we apply a post-verification step consisting of direction and relation checks. This module validates the generated program against both the reasoning path and the target KB, correcting two common error cases: (i) incorrect relation directions despite correct cues in the reasoning path, and (ii) semantically similar but invalid relation labels produced when the true schema item is absent from the path. In the latter case, the generated label is replaced with the closest matching relation in the KB. This lightweight rule-based verification significantly improves final accuracy.

V. Experiments

A. Datasets

Rich-resource Dataset. KQA Pro [7] is a widely used, well-annotated KBQA benchmark built on Wikidata. We sample from it to construct a 60K training set consisting of questions that require 2–4 relational steps (i.e., multi-hop reasoning).

Low-resource Datasets. In addition to KQA Pro, we adopt GrailQA [6], WebQuestions Semantic Parses (WebQSP) [28], ComplexWebQuestions (ComplexWQ) [29], and MetaQA [30] as low-resource target datasets. The first three are built on Freebase, a general-domain KB, while MetaQA is built on WikiMovies, a domain-specific KB focused on movies. MetaQA is further divided into subsets by reasoning hops, enabling fine-grained evaluation under single-hop and multi-hop settings. Since most relations in MetaQA overlap with those in KQA Pro, we remove overlapping schema items from the KQA Pro training set to avoid leakage. We use the KQA Pro test set to evaluate generalization under mixed training data, and the test sets of the remaining datasets to assess cross-KB and cross-domain transferability.

For evaluation, following prior work [7], [25], we report F1 scores on GrailQA, WebQSP, and ComplexWQ, as these datasets may contain multiple correct answers. For MetaQA and KQA Pro, we use Hit@1.

B. Baselines

We compare our method with representative KBQA approaches, grouped into supervised and low-resource methods. **Supervised methods** train models using labeled question–program pairs from the target KB. These include RnG-

KBQA [9], RoG [31], SubgraphRAG [32], UniKGQA [18], ChatKBQA [11], and KG-Agent [14].

Low-resource methods aim to reduce or eliminate reliance on labeled data from the target KB, including few-shot, transfer, and training-free approaches. We consider KB-Binder [10], StructGPT [2], Pangu [13], ToG [12], PoG [33], Program-Trans [15], and KB-Plugin [25].

To ensure fairness, we report results either from the original papers or from our own reimplementations using backbone models with comparable parameter scales.

C. Implementation

In experiments, we use LLaMA-3-8B-Instruct [34], and Qwen2.5-7B-Instruct [35] as backbone LLMs for training the parser. We use PEFT [36] framework for parameter-efficient model training. The parameter of LoRA is set to $r = 8, \alpha = 32$. With respect to the KB navigator, we use GPT-4o [1] as the navigation LLM and set the beam size to 5 and walk range to 3. We utilize 4×A100 GPUs to train the parser for 5 epochs with learning rate $1e-4$, batch size 64, gradient accumulation 2 and weight decay 0.01, with total time cost of about 20 gpu hours.

VI. Results

A. Main Results

We primarily compare REPANA with low-resource methods. Results are reported in Tables I and II.

In Table I, the three datasets are all unseen during training. On GrailQA and WebQSP, REPANA outperforms other low-resource methods by a large margin, despite models like StructGPT and Pangu using much larger backbone models, and is even comparable to some supervised methods. This indicates that REPANA performs excellently on questions with fewer inference hops in WebQSP. We believe this is because our framework can accurately provide paths in the target KB that include the correct relations for the parser to select from, thus generating correct programs. It is worth noting that when reproducing StructGPT, ToG, PoG on GrailQA, as expected, we observe that their performance is significantly affected by questions with no topic entity.

On the more difficult ComplexWQ dataset that involves longer reasoning chains, our method only outperforms PoG by 1.5%. In our observations, we found that it is probably due to the error accumulation in longer paths.

Regarding MetaQA, since its KB is relatively small, most recent low-resource methods have achieved or even surpassed supervised methods, and REPANA also reaches state-of-the-art performance. We noticed that all methods perform bad on 1-hop set compared to 2-hop sets. For REPANA, it is because the 1-hop set contains queries with attribute-based entity lookup, which is currently not supported by our framework.

Table II presents the result on the dev set of KQA Pro. We use the dev set so that we can exclude questions that have schema overlap with the training set, so it can actually be regarded as zero-shot on unseen KB schema items, but we still put REPANA into the supervised group. The results indicate that our performance on KQA Pro is comparable to supervised SoTA. Considering the fact that we did not use the complete training set, we can safely conclude that, REPANA generalizes well on the paraphrased heterogeneous training set.

B. Ablation Study

1) Reasoning Path Effectiveness Evaluation: To validate the importance of the structure of reasoning path as part of the input, we compare parsers that trained with three input of KB information: (1) gold program and reasoning path; (2) gold program and lists of schema items (entity, relation, concept) (3) gold program only with no KB information.

Results in Table III indicate that, in addition to providing the correct schema item names from the target KB, the structural information contained in the reasoning paths is crucial for transfer performance. When the input includes only the relevant schema items but omits their structural relationships, the model struggles to organize them correctly, leading to a significant performance drop. Furthermore, a parser trained solely on program-question pairs from the training set fails to generalize to heterogeneous KBs. Notably, this performance is even lower than directly prompting an LLM for the answer, further highlighting the challenge of transferring across knowledge bases in KBQA.

2) Data Expansion Effectiveness Evaluation: To evaluate the effectiveness of our data expansion strategy with paraphrase, we compare REPANA models trained on the original KQA Pro dataset and on its variants containing different numbers of paraphrased versions, using Qwen-2.5-7B-Instruct as the backbone parser model.

The results in Table IV show that REPANA, without data expansion, achieves performance on WebQSP comparable to other low-resource methods such as KB-Plugin and Pangu, relying solely on the input reasoning path. In contrast, incorporating expanded data consistently improves transfer performance on heterogeneous datasets. Adding three expanded versions yields about a 6% improvement over the original KQA Pro. These results suggest that integrating more heterogeneous training data from additional KBs could further enhance the model's transferability.

3) Intermediate Entity Selection and Post-Verification: We evaluate two components of our framework using Qwen-2.5-7B-Instruct: intermediate entity selection during reasoning path retrieval (described in Section IV-A3) and the post-verification module after program generation (Section IV-B4). For entity selection, we compare LLM-guided selection with random selction, while post-verification applies rule-based checks on relation directions and relation names.

Table V shows that LLM-guided entity selection (LES) has only a marginal effect on performance while introducing additional overhead. This suggests that, for reasoning path retrieval, LLMs are not particularly effective at identifying relevant intermediate entities, which is reasonable since these entities may have little semantic connection to the question. Consequently, we adopt random entity selction in the main experiments. In contrast, the post-verification (PV) module

TABLE I: F1 results on GrailQA, WebQSP and ComplexWQ. Hits@1 results on MetaQA. PV means the post-verification step. Most results are copied from original paper. For StructGPT, ToG, PoG, we reproduce part of results on GPT-4o.

Model	GrailQA	WebQSP	ComplexWQ	MetaQA		
				1-hop	2-hop	3-hop
Supervised						
RnG-KBQA	76.9	75.6	-	-	-	-
RoG	73.2	70.8	56.2	-	-	-
SubgraphRAG	72.7	70.6	47.2	-	-	-
UniKGQA	-	72.2	49.0	98.0	99.9	99.9
ChatKBQA	80.6	79.8	77.8	97.2	98.4	97.1
KG-Agent	86.1	81.0	69.8	97.1	98.0	92.1
Low-resource						
KB-Binder	56.0	53.2	-	93.5	99.6	96.4
Pangu(Codex, 100 shots)	62.7	54.5	-	-	-	-
StructGPT	53.2	69.6	50.5	97.1	97.3	87.0
ToG	52.4	66.5	51.8	-	-	-
PoG	54.6	69.3	53.2	-	-	-
ProgramTrans	-	53.8	45.9	-	-	-
KB-Plugin	65.0	61.1	-	97.1	100.0	**99.3**
ours(Llama3-8B)	78.1	75.2	**54.7**	97.3	100.0	97.5
ours(Qwen2.5-7B)	**80.1**	**76.1**	54.2	**97.5**	**100.0**	97.8

TABLE II: Performance on the KQA Pro dataset, comparing our method with both supervised and low-resource baselines.

	Model	Accuracy
Supervised	RGCN [37]	35.1
	BART+KoPL [7]	90.6
	CFQ IR [38]	89.0
	GraphQ IR [39]	91.7
	KG-Agent	**92.2**
	Ours(Llama2-7B)	90.2
	Ours(Qwen2.5-7B)	90.7
Low-resource	Fine-tuning	22.5
	LLM-ICL	31.8
	FlexKBQA [24]	46.9

TABLE III: Ablation on the form of reasoning path input. $none$, $list$, and $_{path}$ denote using no KB information, a list of candidate relations, and the full reasoning path, respectively.

Model	WebQSP	ComplexWQ
REPANA $_{none}$	19.2	6.7
REPANA $_{list}$	41.6	25.3
REPANA $_{path}$	76.1	54.2

substantially improves program correctness, which means that LLMs still hallucinate sometimes even given the reasoning path, highlighting its crucial role in ensuring accurate relation directions and names.

4) Ablation on Reasoning Chain Length: This ablation study is to examine how performance and retrieval latency vary with the length of the reasoning path. Retrieval latency includes both LLM inference time and KB querying time. We sample questions from WebQSP and ComplexWQ to construct a 100-example test set, evenly distributed across 1-, 2-, 3-, and 4-hop questions. Each experiment is repeated three times.

As shown in Fig. 2, retrieval latency increases steadily with the number of reasoning hops as expected. Neverthe-

TABLE IV: Ablation of data expansion. $kqa\text{-}pro$: trained on KQA Pro only; $exp\text{-}n$: trained on KQA Pro and n expanded datasets.

Model	WebQSP	ComplexWQ
REPANA $_{kqa\text{-}pro}$	70.2	48.7
REPANA $_{exp\text{-}1}$	73.7	51.8
REPANA $_{exp\text{-}2}$	75.2	53.6
REPANA $_{exp\text{-}3}$	76.1	54.2

TABLE V: Ablation on intermediate entity selection and post-verification. *Base* uses random entity selection without post-verification; *LES* indicates LLM-based entity selection; *PV* indicates post-verification.

Model Variant	WebQSP	ComplexWQ
Base	65.6	40.7
Base + LES	65.8	41.1
Base + PV	76.1	54.2
Base + LES + PV	76.2	54.1

less, the overall latency remains moderate and acceptable for most questions, indicating that the proposed framework scales reasonably with reasoning depth. In terms of performance, we observe a consistent degradation as the reasoning length increases, which aligns with observations from prior work. Despite this decline, REPANA maintains strong performance on 3-hop questions and achieves acceptable results on 4-hop questions, comparing to prior low-resource methods. These results suggest that while long reasoning chains remain challenging, the proposed navigation-guided framework is relatively robust to increasing reasoning depth.

VII. Conclusion

In this paper, we propose REPANA, a reasoning path navigated program induction framework that enables LLMs

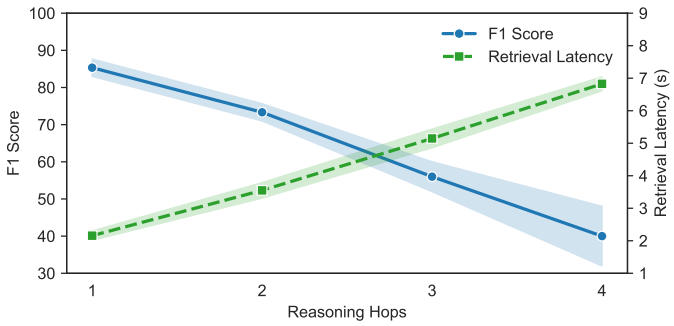

Fig. 2: Performance and retrieval latency under different reasoning hop lengths.

to universally perform reasoning on low-resource datasets by providing the KB-agnostic parser with the reasoning paths in target KBs with the help of the novel KB navigator. REPANA achieves better performance on the four heterogeneous target datasets even when our backbone models is much smaller, and can perform on par with some supervised methods. The ablation studies further validate the effectiveness of our proposed KB-walk retrieving strategy and mixed instruction tuning in low-resource scenarios.

VIII. Acknowledgments

We would like to thank all the anonymous reviewers for their careful work and thoughtful suggestions. This work is supported by a grant from the Institute for National Natural Science Foundation of China (62476150), Guo Qiang, Tsinghua University (2019GQB0003), and Tsinghua University (Department of Computer Science and Technology)-Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIIOT).

References

[1] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, A. Madry, A. Baker-Whitcomb, A. Beutel, A. Borzunov, A. Carney, A. Chow, A. Kirillov, A. Nichol, A. Paino, A. Renzin, A. T. Passos, A. Kirillov, A. Christakis, A. Conneau, A. Kamali, A. Jabri, A. Moyer, A. Tam, A. Crookes, A. Tootoonchian, A. Kumar, A. Vallone, A. Karpathy, A. Braunstein, A. Cann, A. Codispoti, A. Galu, A. Kondrich, A. Tulloch, A. Mishchenko, A. Baek, A. Jiang, A. Pelisse, A. Woodford, A. Gosalia, A. Dhar, A. Pantuliano, A. Nayak, A. Oliver, B. Zoph, B. Ghorbani, B. Leimberger, B. Rossen, B. Sokolowsky, B. Wang, B. Zweig, B. Hoover, B. Samic, B. McGrew, B. Spero, B. Giertler, B. Cheng, B. Lightcap, B. Walkin, B. Quinn, B. Guarraci, B. Hsu, B. Kellogg, B. Eastman, C. Lugaresi, C. L. Wainwright, C. Bassin, C. Hudson, C. Chu, C. Nelson, C. Li, C. J. Shern, C. Conger, C. Barette, C. Voss, C. Ding, C. Lu, C. Zhang, C. Beaumont, C. Hallacy, C. Koch, C. Gibson, C. Kim, C. Choi, C. McLeavey, C. Hesse, C. Fischer, C. Winter, C. Czarnecki, C. Jarvis, C. Wei, C. Koumouzelis, and D. Sherburn, "Gpt-4o system card," *CoRR*, vol. abs/2410.21276, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.21276

[2] J. Jiang, K. Zhou, Z. Dong, K. Ye, X. Zhao, and J. Wen, "Structgpt: A general framework for large language model to reason over structured data," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 9237–9251. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.574

[3] X. Li, R. Zhao, Y. K. Chia, B. Ding, L. Bing, S. R. Joty, and S. Poria, "Chain of knowledge: A framework for grounding large language models with structured knowledge bases," *CoRR*, vol. abs/2305.13269, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.13269

[4] T. Xie, C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer, and T. Yu, "Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 602–631. [Online]. Available: https://doi.org/10.18653/v1/2022.emnlp-main.39

[5] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *CoRR*, vol. abs/2311.05232, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.05232

[6] Y. Gu, S. Kase, M. Vanni, B. M. Sadler, P. Liang, X. Yan, and Y. Su, "Beyond I.I.D.: three levels of generalization for question answering on knowledge bases," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds. ACM / IW3C2, 2021, pp. 3477–3488. [Online]. Available: https://doi.org/10.1145/3442381.3449992

[7] S. Cao, J. Shi, L. Pan, L. Nie, Y. Xiang, L. Hou, J. Li, B. He, and H. Zhang, "KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 6101–6119. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-long.422

[8] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of SPARQL," in *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, ser. Lecture Notes in Computer Science, I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, Eds., vol. 4273. Springer, 2006, pp. 30–43. [Online]. Available: https://doi.org/10.1007/11926078_3

[9] X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, and C. Xiong, "RNG-KBQA: generation augmented iterative ranking for knowledge base question answering," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 6032–6043. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-long.417

[10] T. Li, X. Ma, A. Zhuang, Y. Gu, Y. Su, and W. Chen, "Few-shot in-context learning on knowledge base question answering," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 6966–6980.

[11] H. Luo, H. E, Z. Tang, S. Peng, Y. Guo, W. Zhang, C. Ma, G. Dong, M. Song, and W. Lin, "Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models," *CoRR*, vol. abs/2310.08975, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2310.08975

[12] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. M. Ni, H. Shum, and J. Guo, "Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=nnVO1PvbTv

[13] Y. Gu, X. Deng, and Y. Su, "Don't generate, discriminate: A proposal for grounding language models to real-world environments," 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*, A. Rogers, J. L. Boyd-Graber, and N. Okazaki, Eds. Association for Computational Linguistics, 2023, pp. 4928–4949. [Online]. Available: https://doi.org/10.18653/v1/2023.acl-long.270

[14] J. Jiang, K. Zhou, W. X. Zhao, Y. Song, C. Zhu, H. Zhu, and J. Wen, "Kg-agent: An efficient autonomous agent framework for complex

reasoning over knowledge graph," *CoRR*, vol. abs/2402.11163, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.11163

[15] S. Cao, J. Shi, Z. Yao, X. Lv, J. Yu, L. Hou, J. Li, Z. Liu, and J. Xiao, "Program transfer for answering complex questions over knowledge bases," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 8128–8140. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-long.559

[16] D. Vrandecic and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, 2014. [Online]. Available: https://doi.org/10.1145/2629489

[17] K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. ACM, 2008. [Online]. Available: https://doi.org/10.1145/1376616.1376746

[18] J. Jiang, K. Zhou, X. Zhao, and J. Wen, "Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=Z63RvyAZ2Vh

[19] D. Yu, S. Zhang, P. Ng, H. Zhu, A. H. Li, J. Wang, Y. Hu, W. Y. Wang, Z. Wang, and B. Xiang, "Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=XHc5zRPxqV9

[20] H. Sun, T. Bedrax-Weiss, and W. W. Cohen, "Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text," 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*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 2380–2390. [Online]. Available: https://doi.org/10.18653/v1/D19-1242

[21] J. Shi, S. Cao, L. Hou, J. Li, and H. Zhang, "Transfernet: An effective and transparent framework for multi-hop question answering over relation graph," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. Association for Computational Linguistics, 2021, pp. 4149–4158. [Online]. Available: https://doi.org/10.18653/v1/2021.emnlp-main.341

[22] H. Luo, H. E, Y. Guo, Q. Lin, X. Wu, X. Mu, W. Liu, M. Song, Y. Zhu, and A. T. Luu, "Kbqa-o1: Agentic knowledge base question answering with monte carlo tree search," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=QuecSemZIy

[23] B. Bogin, S. Gupta, P. Clark, and A. Sabharwal, "Leveraging code to improve in-context learning for semantic parsing," *CoRR*, vol. abs/2311.09519, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.09519

[24] Z. Li, S. Fan, Y. Gu, X. Li, Z. Duan, B. Dong, N. Liu, and J. Wang, "Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 18 608–18 616. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29823

[25] J. Zhang, S. Cao, L. Hu, L. Feng, L. Hou, and J. Li, "Kb-plugin: A plug-and-play framework for large language models to induce programs over low-resourced knowledge bases," *CoRR*, vol. abs/2402.01619, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.01619

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[27] J. Liu, S. Cao, J. Shi, T. Zhang, L. Nie, L. Hu, L. Hou, and J. Li, "How proficient are large language models in formal languages? an in-depth insight for knowledge base question answering," in *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 792–815. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-acl.45

[28] W. Yih, M. Richardson, C. Meek, M. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016. [Online]. Available: https://doi.org/10.18653/v1/p16-2033

[29] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 641–651. [Online]. Available: https://doi.org/10.18653/v1/n18-1059

[30] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *AAAI*, 2018.

[31] L. Luo, Y. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=ZGNWW7xZ6Q

[32] M. Li, S. Miao, and P. Li, "Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=JvkuZZ04O7

[33] L. Chen, P. Tong, Z. Jin, Y. Sun, J. Ye, and H. Xiong, "Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024.

[34] Meta, "Introducing meta llama 3: The most capable openly available llm to date," https://ai.meta.com/blog/meta-llama-3/, 2024.

[35] Q. Team, "Qwen2.5: A party of foundation models," September 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen2.5/

[36] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan, "Peft: State-of-the-art parameter-efficient fine-tuning methods," https://github.com/huggingface/peft, 2022.

[37] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, ser. Lecture Notes in Computer Science, A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., vol. 10843. Springer, 2018, pp. 593–607. [Online]. Available: https://doi.org/10.1007/978-3-319-93417-4_38

[38] J. Herzig, P. Shaw, M. Chang, K. Guu, P. Pasupat, and Y. Zhang, "Unlocking compositional generalization in pre-trained models using intermediate representations," *CoRR*, vol. abs/2104.07478, 2021. [Online]. Available: https://arxiv.org/abs/2104.07478

[39] L. Nie, S. Cao, J. Shi, J. Sun, Q. Tian, L. Hou, J. Li, and J. Zhai, "Graphq IR: unifying the semantic parsing of graph query languages with one intermediate representation," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 5848–5865. [Online]. Available: https://doi.org/10.18653/v1/2022.emnlp-main.394