Tournament on Path: Global Relation Abstraction and Pair-wise Ranking for Knowledge Graph Reasoning

Anonymous ACL submission

Abstract

While Large Language Models (LLMs) integrated with Knowledge Graphs (KGs) show promise in multi-hop reasoning, existing step-wise frameworks suffer from shortsighted pruning decisions and unstable individual candidate scoring. To address these limitations, we propose Tournament on Path (ToP), which abstracts graphs into entity-agnostic relation paths, enabling global reasoning, and introduces a tournament pair-wise ranking strategy for more robust candidate selection. Evaluated on four benchmarks, ToP consistently outperforms strong baseline and improves scaling behavior across different foundation models. Source code: https://anonymous.4open.science/r/top-EE65.

1 Introduction

Recent advances in Large Language Models (LLMs) have demonstrated remarkable performance across various tasks. However, they still suffer from hallucinations and factual inaccuracies in knowledge-intensive scenarios (Ji et al., 2023; Huang and Chang, 2023). Retrieval-Augmented Generation (RAG) mitigates this issue by incorporating external knowledge during inference (Gao et al., 2024). Among different knowledge sources, Knowledge Graphs (KGs) provide structured and reliable factual information, making them particularly suitable for reasoning-intensive tasks. Consequently, Knowledge Graph Question Answering (KGQA), which answers questions through reasoning over KGs, has become an important research topic (Ma et al., 2025).

For complex questions, single-step reasoning is often insufficient because the starting entity (in the question) and answer entities are usually far away in the KGs, requiring multi-hop reasoning. To address this challenge, recent methods leverage LLMs as agents to perform step-wise reasoning over KGs by iteratively exploring relations (Pan et al., 2024). A representative approach is Think-on-Graph (ToG) (Sun et al., 2024), which utilizes the reasoning capability of LLMs to guide graph traversal. At each step, ToG scores one-hop neighboring entities individually and selects the most relevant candidates for further exploration. Such methods can directly leverage off-the-shelf LLMs and treat KG reasoning as a plug-and-play component, making them simple and practical to deploy.

However, this paradigm suffers from several inherent limitations. First, relation pruning decisions are locally scoped, failing to capture the global semantic intent of complex queries. Second, pruning is implemented by point-wise scoring, introducing instability in candidate selection. Finally, ToG also resorts to random sampling to resolve the large number of neighbors in KGs, inevitably discarding useful neighbors.

To address these issues, we propose Tournament on Path (ToP) with two main improvements. First, we merge exploration paths by relation path equivalence, ignoring intermediate entities. This structure enables global path-level evaluation and reduces entity-level noise. Second, to further optimize pruning stability, we introduce a tournament strategy. This replaces unstable point-wise scoring with a pair-wise ranked tournament algorithm after a semantic pre-filter. Our main contributions are as follows:

- We mitigate shortsighted pruning decisions by abstracting graphs into entity-agnostic relation paths, enabling global path-level reasoning.
- We design a tournament selection strategy that replaces point-wise scoring with pair-wise ranking, improving selection stability.
- Experiments on four benchmark datasets show that our method consistently improves reasoning robustness and pruning stability.

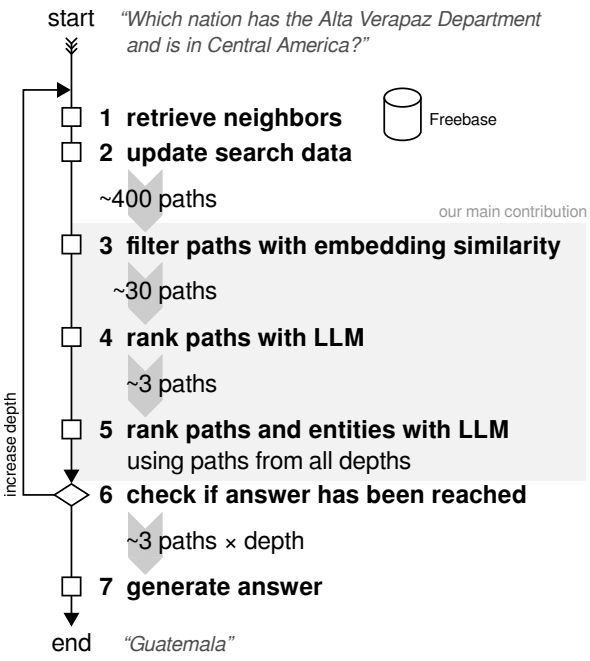

Figure 1: Overall architecture of ToP

2 Method

In this section, we propose two methods within ToP, each addressing a shortcoming in the ToG architecture. Figure 1 illustrates the overall architecture of our ToP framework.

2.1 Entity-Agnostic Relation Path Pruning

We observe that in multi-hop reasoning over knowledge graphs, the semantic utility of a reasoning path is largely governed by its sequence of relations. For example, *Chicago* $\xrightarrow{\textit{located_in}}$ *USA* $\xrightarrow{\textit{national_bird}}$ *Bald Eagle* can be abstracted as the compositional relation $R(X, Z)$ defined over relation paths.

$$R(X, Z) = \exists Y, \text{located_in}(X, Y) \wedge \text{national_bird}(Y, Z) \quad (1)$$

As ToG performs relation pruning step by step, it remains limited to local context and may fail to capture global path semantics. To address this, ToP performs relation pruning using—and only using—relation information. Intermediate entities are deliberately ignored in the relation path pruning process, thereby merging full entity-aware paths by identical start entity and sequence of relations (Figure 2). More formally, full paths of the form

$$e_0, r_1, e_1, r_2, e_2, \cdots, r_n, e_n$$

where e denotes entity, e_0 is a given topic entity, and r denotes relation, are path-equivalent if their subsequences $e_0, r_1, r_2, \cdots, r_n$ are identical. This approach enables more effective pruning by allowing the LLM to evaluate global path semantics while reducing entity-level noise.

This approach does have some limitations. For example, consider the question "What is the nationality of the tallest person in the world?" on a knowledge graph without the superlative relation "tallest person in". To answer this question, it is necessary to instantiate the entity referred to by "the tallest person in the world", which is impossible. Despite this limitation in expressiveness, we propose our method as it brings clear performance improvements and may still be extended to support this kind of expressiveness.

2.2 Tournament Candidate Selection

In order to improve the stability of scoring, ToP introduces a tournament algorithm after a pre-filter.

Semantic Pre-filter On Freebase, the size of the candidate pool usually reach hundreds after exploration at each depth. The original ToG framework manages this by randomly discarding candidates to fit a fixed budget prior to LLM scoring. This inevitably discards relevant entities.

To reduce the search space while preserving relevance, ToP employs a semantic pre-filter by selecting top m paths with the highest embedding similarity against the natural language question[1].

LLM as Tournament Chooser, Not Scorer We reformulate candidate selection as an LLM-bound pair-wise selection process.

The original ToG relies on assigning absolute numerical scores, leading to scoring failures such as degrading to extreme scores (0 or 1) and assigning the same score to vastly different candidates. This may be due to the absence of calibration in LLM scoring. Recent work on LLM-based evaluation has shown that relative comparative ranking improves evaluation consistency over absolute scoring (Zheng et al., 2026; Sun et al., 2023; Lee et al., 2025). Jointly evaluating candidates provides richer mutual context, enabling the LLM to capture fine-grained differences for improved decision-making. Building on this observation, ToP treats the LLM as a comparative chooser rather than a point-wise scorer.

LLM-Guided Tournament Selection Our tournament algorithm is a version of the well-

[1]We used $m = 30$.

documented merge sort algorithm, where the LLM powers the compare function. For our purpose, we do not require LLM comparison to be consistent (for example, $a > b > c > a$ is permissible) or transitive, as our goal is to exploit the LLM's reasoning capability enough to probabilistically find the more accurate paths.

Traditionally, the merge step in merge sort creates an empty new list l, compares the top item from each of the two input lists, moves the greater one to the new list l, and iterates until one input list is empty, in which case all remaining items are directly added to the new list l. We modify merge sort to limit the item count in l to some k during this merge step[2]. The modified merge routine ends once the merged list l reaches size k, saving time that would have been used to compare remaining (worse) candidates. This algorithm has a time complexity of $O(n \log n)$ if all comparisons are serial.

2.3 Prompt Design

Our prompt design reflects the architectural shifts introduced above: ignoring entities in relation paths for prompting, and replacing point-wise scoring with pair-wise selection. The design of our prompts can be found in the appendix A.

Diff prompting for path comparison In order to trick the LLM into comparison, we symbolically extract the longest common prefix of the two path candidates. This common prefix is presented as "given conditions" to the LLM. The remaining segment of each path, the *path diff*, each becomes an option. In case of any empty path diff, the question is formulated as a relevance decision problem and only the nonempty path diff is presented.

3 Experiment

In this section, we focus on the following research questions.

1. To what extent does the proposed framework improve multi-hop reasoning performance over standard step-wise baseline?

2. How does model scale affect reasoning performance, and how does the proposed method reshape this scaling behavior?

3. How consistently does the proposed framework perform across datasets with varying schema complexities, compared to the baseline?

To answer these questions, we first detail our experiment setup, and then conduct a detailed analysis across three dimensions.

3.1 Setup

Datasets and Metrics Following (Sun et al., 2024), we employ Freebase (Bollacker et al., 2008) as our underlying knowledge graph. We evaluate our method across four widely recognized benchmarks that feature reasoning tasks of different complexities: CWQ (Talmor and Berant, 2018), WebQuestions (Berant et al., 2013) (WebQ), WebQSP (Yih et al., 2016), and GrailQA (Gu et al., 2021). To optimize computational efficiency while maintaining evaluation integrity, we randomly sample a subset of 200 questions from each dataset for our evaluation. We adopt **Exact Match** (EM) as our primary metric to determine whether the predicted answer aligns with the ground-truth annotations.

Language Models To demonstrate the generalization and robustness of our proposed method across different scales of LLMs, we conduct experiments using four representative small to medium-sized open-source reasoning models: Qwen 3 14B, 32B (Yang et al., 2025); and Gemma 4 E2B IT, E4B IT (instruction-tuned) (Google DeepMind, 2026). The pre-filter in ToP is Qwen 3 Embedding 4B (Zhang et al., 2025) for all experiments.

3.2 Analysis

We conduct an analysis across three distinct dimensions. The results are shown in Table 1.

3.2.1 Performance Gains Across Frameworks

Our experiment results show that ToP consistently outperforms the standard ToG baseline across nearly all tested LLM and benchmarks. Notably, when paired with smaller models, such as Gemma 4 E2B, ToP achieves substantial EM improvements of 27.5% on CWQ, 35.0% on WebQ, 42.5% on WebQSP and 57.0% on GrailQA. Even for larger models like Qwen 3 32B, ToP still yields significant gains, improving performance by up to 24.0% absolute on GrailQA.

These empirical improvements suggest that restructuring step-wise path pruning into path-level reasoning in conjunction with tournament candidate selection leads to more effective filtering of reasoning paths, especially for small LLMs.

[2]We used $k = 3$.

Models		CWQ	WebQ	WebQSP	GrailQA
Qwen 3 32B	w/ ToG	38.0	46.0	59.5	50.0
	w/ ToP	43.5	63.5	73.0	74.0
Qwen 3 14B	w/ ToG	34.0	54.0	62.5	56.0
	w/ ToP	37.5	57.0	62.5	59.5
Gemma 4 E4B IT	w/ ToG	31.0	28.0	39.0	53.5
	w/ ToP	46.0	56.5	67.5	74.0
Gemma 4 E2B IT	w/ ToG	12.0	17.0	22.0	10.5
	w/ ToP	39.5	52.0	64.5	67.5

Table 1: Reasoning performance of various LLMs equipped with ToP versus ToG. The results demonstrate that ToP not only achieves an overall improvement over ToG, but also scales with LLM size and exhibits consistency across datasets.

3.2.2 Impact of LLM Model Scale and Capability

We observed that the baseline ToG results show a certain degree of performance inversion, or "negative scaling", among Qwen models as the parameter size increases. For instance, the larger Qwen 3 32B underperforms the smaller Qwen 3 14B across multiple datasets, including WebQ (46.0% vs 54.0%), GrailQA (50.0% vs 56.0%), and WebQSP (59.5% vs 62.5%). We attribute this to longer and more flexible responses, which more frequently violate formatting requirements, generated by larger models.

There is an apparent performance inversion between the Qwen 3 and Gemma 4 LLM families, specific to ToP and not ToG. We attribute this to the different sensitivities of LLMs to certain prompt styles. During pruning, ToG asks scoring questions with a list of candidates using few-shot prompting, while ToP asks zero-shot multiple-choice questions with exactly two options and requires the LLM to show its reasoning process. The choice of an instruction-tuned model (Gemma 4 E4B IT and E2B IT) may also have made a difference. Finally, it is worth noting that the Gemma 4 LLM family is approximately one year more recent than Qwen 3, entailing non-trivial systematic changes.

3.2.3 Performance Consistency Across Datasets

ToP exhibits stable performance across different benchmarks, with variations aligned with the baseline. This pattern indicates that ToP provides consistent behavior across datasets rather than being optimized to a specific benchmark, and the variations should be explained by differences in question complexity.

Methods	CWQ	WebQ
ToP	46.0	56.5
ToP w/o Relation Path	33.0	45.5
ToP w/o Tournament	32.5	53.0

Table 2: Ablation study under Gemma 4 E4B IT.

3.3 Ablation Study

Table 2 presents the ablation study results of the ToP framework using Gemma 4 E4B IT to evaluate our two core components.

Ablation of relation path pruning Only the final relation in the relation path is used for pruning.

Ablation of tournament The candidates are point-wise scored rather than ranked by pair-wise comparison. We batch 5 items per LLM prompt. Scoring is implemented for both entity-agnostic path pruning and final path ranking with entity information.

Both ablations reduce performance from full ToP on CWQ and WebQ, demonstrating the necessity of these modifications.

4 Conclusion

In this work, we introduced Tournament on Path, a novel framework that improves upon current paradigms through two key mechanisms: abstracting exploration paths into entity-agnostic relation sequences for global path-level reasoning, and replacing unstable point-wise scoring with a robust pair-wise tournament selection algorithm. Extensive experiments across four KGQA benchmarks and four LLMs demonstrate that ToP consistently outperforms baseline methods and scales positively across different models.

Limitations

Although our approach mitigates shortsighted pruning decisions by taking advantage of a global evaluation perspective, the reasoning process remains grounded in the connectivity of the graph. Consequently, if critical relation paths or entities are missing from the underlying KG, the structural search space becomes fragmented. In such scenarios, even ToP cannot retrieve the necessary evidence to complete the reasoning chain, compromising the overall performance.

Acknowledgments

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Google DeepMind. 2026. Gemma 4 model card. `https://ai.google.dev/gemma/docs/core/model_card_4`. Accessed: 2026-05-25.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, WWW '21, page 3477–3488, New York, NY, USA. Association for Computing Machinery.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. 2025. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17606–17619, Vienna, Austria. Association for Computational Linguistics.

Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025. Large language models meet knowledge graphs for question answering: Synthesis and opportunities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24578–24597, Suzhou, China. Association for Computational Linguistics.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap . *IEEE Transactions on Knowledge & Data Engineering*, 36(07):3580–3599.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. 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, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Shuai Wang and Yinan Yu. 2026. Kg-reasoner: A reinforced model for end-to-end multi-hop knowledge graph reasoning. *Preprint*, arXiv:2604.12487.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin,

Germany. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Pujun Zheng, Jiacheng Yao, Jinquan Zheng, Chenyang Gu, Guoxiu He, Jiawei Liu, Yong Huang, Tianrui Guo, and Wei Lu. 2026. From isolated scoring to collaborative ranking: A comparison-native framework for llm-based paper evaluation. *Preprint*, arXiv:2603.17588.

A Prompt Design

A.1 Relation Pruning

As part of merge sort, our prompt asks the LLM to compare the relevance of two relation paths.

Every pair of paths has a unique longest common prefix, which is extracted and presented as "given conditions" to the LLM. The remaining segment of each path, the *path diff*, is presented as an option.

In case of two nonempty path diffs, the following zero-shot prompt structure is used:

```
You are helping a computer program
    perform question answering on a
    knowledge graph.
Do not pick information that is
    redundant.
In combination with the known conditions
    , which one of the candidates is
    more relevant to the following
    question?
Question: m.03_dwn is the mascot for the
     team that last won the world series
     when?
Given conditions: -> GIVEN_TOPIC_ENTITY:
     m.03_dwn
Candidate A: -> sports.mascot.team <-
    common.webpage.topic
Candidate B: <- sports.sports_team.
    team_mascot <- award.
    award_nomination.award_nominee
First explain your reasons in one
    sentence for each option, then write
     the response as [A] or [B].
```

Sample response:

```
Candidate A describes a general
    relationship between a mascot and a
    webpage topic, which is too broad.
    Candidate B connects the team mascot
     to sports teams and awards, which
    is relevant to winning a world
    series.
[B]
```

In case of any empty path diff, the following zero-shot prompt structure is used:

```
You are helping a computer program
    perform question answering on a
    knowledge graph.
Do not pick information that is
    redundant.
In combination with the known conditions
    , is the candidate relevant to the
    following question?
Question: m.03_dwn is the mascot for the
     team that last won the world series
     when?
Given conditions: -> GIVEN_TOPIC_ENTITY:
     m.03_dwn <- sports.sports_team.
    team_mascot
Candidate: <- baseball.baseball_league.
    teams
First explain your reasons in one
    sentence for each option, then write
     the response as [RELEVANT] or [NOT
    RELEVANT].
```

Sample response:

```
The candidate is too broad and does not
    specifically mention the team that
    won the world series or the mascot
    mentioned in the question.
[NOT RELEVANT]
```

A.2 Answer Generation

For final answer generation, we present the LLM with the top-ranked paths along with the query results (entity names or values), then ask for the answer if one exists, or a "no answer" symbol in a zero-shot manner.

```
You are helping a computer program
    perform question answering on a
    knowledge graph.
You have access to some relation paths
    and their query results, and your
    task is to identify the answer, or
    report if it does not exist here.
If some paths provide the answer, simply
     give the answer.
If no path provides any answer, write [
    NO ANSWER].

Question:
m.03_dwn is the mascot for the team that
     last won the world series when?

Paths:
-> GIVEN_TOPIC_ENTITY: m.03_dwn <-
    sports.sports_team.team_mascot <-
    sports.sports_championship_event.
    champion query result: [['2010 World
     Series'], ['2012 World Series'],
    ['2014 World Series']] and 0 unnamed
     objects
-> GIVEN_TOPIC_ENTITY: m.03_dwn <-
    sports.sports_team.team_mascot <-
    sports.sports_league_participation.
    team query result: [] and 3 unnamed
    objects
-> GIVEN_TOPIC_ENTITY: m.03_dwn <-
    sports.sports_team.team_mascot query
     result: [['San Francisco Giants']]
    and 0 unnamed objects
```

```
-> GIVEN_TOPIC_ENTITY: m.03_dwn <-
    sports.sports_team.team_mascot <-
    baseball.baseball_league.teams query
     result: [['National League']] and 0
     unnamed objects
-> GIVEN_TOPIC_ENTITY: m.03_dwn ->
    sports.mascot.team query result: [['
    San Francisco Giants']] and 0
    unnamed objects
```

Sample response:

```
2014 World Series
```

B Method Details

Figure 2 illustrates the Entity-Agnostic Relation Path construction.

C Overview of Datasets

Table 3 shows the basic information of datasets used in this thesis.

D Case Study

Table 4 and Table 5 are cases that shows the reasoning outcomes of ToG and ToP.

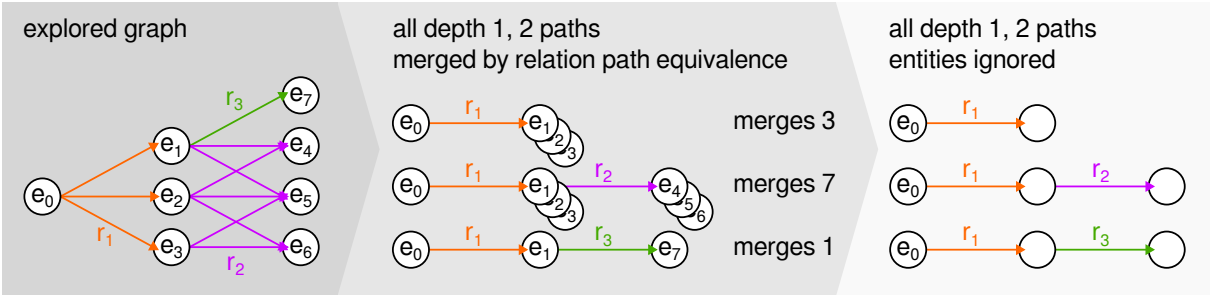

Figure 2: **Entity-Agnostic Relation Path Pruning** The explored subgraph (left) is first grouped by relation path equivalence (center). Then, intermediate entities are ignored, abstracting paths into relational paths (right).

Dataset	KG	Complexity	Reasoning	Annotation	Source
CWQ	Freebase	Complex Qs	Multi-hop	SPARQL, Ans.	Web-derived
WebQSP	Freebase	Moderate Qs	Single-hop (most)	SPARQL, Ans.	Web-derived
WebQuestions	Freebase	Simple-Moderate Qs	Single-hop (most)	Ans. only	Web-derived
GrailQA	Freebase	Complex (varied)	Multi-hop	Logic forms, Ans.	Human-controlled

Table 3: Overview of KGQA Datasets (Wang and Yu, 2026)

Question: "The person who attended education institution in East Whittier Elementary School died where?"

ToG Output

```
{
    "question": "The person who attended education institution in East Whittier Elementary School
                 died where?",
   "results": "{Yes}. Based on the given knowledge triplets, we can identify that Richard Nixon attended
               East Whittier Elementary School. The triplets also state that Richard Nixon died in
               New York City. Therefore, the answer to the question is {New York City}.",
    "reasoning_chains": [
        [
            [
                [
                    "East Whittier Elementary School",
                    "education.educational_institution.students_graduates",
                    "UnName_Entity"
                ]
            ]
        ]
    ]
}
```

ToP Paths and Output

```
Paths:
-> GIVEN_TOPIC_ENTITY: m.0_sp3dx <- education.education.institution -> education.education.student ->
people.deceased_person.place_of_death query result: [['New York City']] and
0 unnamed objects
-> GIVEN_TOPIC_ENTITY: m.0_sp3dx <- education.education.institution -> education.education.student <-
people.place_of_interment.interred_here query result: [['Richard Nixon Library & Birthplace']] and
0 unnamed objects
-> GIVEN_TOPIC_ENTITY: m.0_sp3dx <- education.education.institution -> education.education.student ->
people.deceased_person.place_of_burial query result: [['Richard Nixon Library & Birthplace']] and
0 unnamed objects
-> GIVEN_TOPIC_ENTITY: m.0_sp3dx -> education.educational_institution.students_graduates query result:
[] and 1 unnamed objects
-> GIVEN_TOPIC_ENTITY: m.0_sp3dx <- education.education.institution query result: [] and
1 unnamed objects

{"response": "['New York City']"}
```

Table 4: Both ToP and ToG generated correct answer.

Question: "Lou Seal is the mascot for the team that last won the World Series when?"
ToG Output

{
"question": "Lou Seal is the mascot for the team that last won the World Series when?",
"results": "Based on the given knowledge triplets, the mascot for the San Francisco Giants is Lou Seal. However, the triplets do not contain information about when the San Francisco Giants last won the World Series. Therefore, additional knowledge is required to answer the question.",
"reasoning_chains": [
[
[
[
"Lou Seal",
"sports.mascot.team",
"San Francisco Giants"
]
]
]
]
}

ToP Paths and Output
Paths: -> GIVEN_TOPIC_ENTITY: m.03_dwn <- sports.sports_team.team_mascot <- sports.sports_championship_event.champion query result: [['2010 World Series'], ['2012 World Series'], ['2014 World Series']] and 0 unnamed objects -> GIVEN_TOPIC_ENTITY: m.03_dwn <- sports.sports_team.team_mascot <- sports.sports_league_participation.team query result: [] and 3 unnamed objects -> GIVEN_TOPIC_ENTITY: m.03_dwn <- sports.sports_team.team_mascot query result: [['San Francisco Giants']] and 0 unnamed objects -> GIVEN_TOPIC_ENTITY: m.03_dwn <- sports.sports_team.team_mascot <- baseball.baseball_league.teams query result: [['National League']] and 0 unnamed objects -> GIVEN_TOPIC_ENTITY: m.03_dwn -> sports.mascot.team query result: [['San Francisco Giants']] and 0 unnamed objects {"response": "2014 World Series"}

Table 5: ToG failed to generate enough reasoning chains and fail to answer the question. In contrast, ToP explored plenty of paths, which are sufficient to give an answer.