

# PathSoft: Hybrid Soft Prompting with Ranked Reasoning Paths for Knowledge Graph Question Answering

Yan Teng<sup>1,3</sup>, Xiaoding Zhou<sup>1,3</sup>, Jinshuo Zhang<sup>1,3</sup>, Ying Lian<sup>2</sup>,  
Guoqiang Chen<sup>2</sup>, Hongjiao Guan<sup>1,3</sup>, Weiyu Zhang<sup>1,3,\*</sup>, Wenpeng Lu<sup>1,3</sup>

<sup>1</sup>Key Laboratory of Computing Power Network and Information Security,  
Ministry of Education, Shandong Computer Science Center  
(National Supercomputer Center in Jinan),

Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

<sup>2</sup>The First Affiliated Hospital of Shandong First Medical University  
(Shandong Provincial Qianfoshan Hospital), Jinan, China

<sup>3</sup>Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing,  
Shandong Fundamental Research Center for Computer Science, Jinan, China  
zwy@qlu.edu.cn

## Abstract

Large language models (LLMs) have demonstrated strong reasoning abilities, but hallucinations and unverifiable intermediate reasoning steps remain key challenges. In knowledge graph question answering (KGQA), entity–relation paths from knowledge graphs (KGs) provide structured and verifiable evidence for grounded reasoning. However, existing KG-enhanced LLM methods commonly linearize all retrieved reasoning paths into textual prompts, introducing noise and underexploiting their ordered structure. To address these limitations, we propose **PathSoft**, a hybrid soft prompting framework that combines textual path evidence with soft prompts. PathSoft first employs PathMLP to select reliable reasoning paths, and then uses PathSoftAdapter to convert the selected paths into continuous soft prompts while preserving their ordered path structure. Finally, the resulting soft prompt tokens are combined with the textual prompt at the embedding layer of the frozen LLM, enabling it to exploit both readable path evidence and path structure information. Experiments on WebQSP and CWQ show that PathSoft outperforms existing KGQA baselines. The code is available at <https://github.com/tengyan1786-blip/PathSoft>.

## 1 Introduction

Large language models (LLMs) have demonstrated strong reasoning abilities in complex tasks. However, their reasoning processes can still be prone to hallucinations and unverifiable intermediate steps, limiting their reliability in factual reasoning tasks.

Knowledge graph question answering (KGQA) is a representative task for grounded reasoning, in which models answer natural language questions using factual knowledge from KGs. Knowledge graphs (KGs) can ground LLM reasoning in explicit entity–relation facts, providing structured and verifiable evidence (Pan et al., 2024; Agrawal et al., 2024).

Recent KG-augmented LLM studies have investigated two major paradigms for incorporating KG evidence into LLMs for KGQA. The first paradigm retrieves or explores relevant graph evidence from KGs and linearizes it into textual context for LLMs. This line includes subgraph-based methods (Li et al., 2025; Gao et al., 2025; Li et al., 2024) and path-based methods (Luo et al., 2024; Sun et al., 2024; Sui et al., 2025; Lin et al., 2025). For example, SubgraphRAG (Li et al., 2025) retrieves relevant subgraphs for answer prediction, while RoG (Luo et al., 2024) uses KG paths to guide LLM reasoning. The second paradigm employs soft prompting or adapters to inject structured knowledge into frozen or partially frozen LLMs (Ao et al., 2025; Liu et al., 2025). For instance, LightPROF (Ao et al., 2025) encodes reasoning graphs into continuous prompt embeddings. Overall, these studies demonstrate the effectiveness of textual and continuous prompts for enabling LLMs to use KG evidence in grounded reasoning.

Among these textual KG evidence forms, multi-hop reasoning paths serve as a key form of evidence for guiding LLM reasoning, yet two major challenges remain. **First, a question may be associated with many candidate paths, while only a small subset is likely to lead to the correct an-**

\* Corresponding author

**swer.** Prior work has similarly shown that retrieved evidence should be selectively incorporated into LLM reasoning (Zhang et al., 2026a). Therefore, selecting reliable paths before generation is crucial for robust answer prediction. **Second, reasoning paths encode ordered dependencies among entities and relations, yet existing methods often represent them as plain text or aggregate them into graph-level representations, leaving their internal sequential structure underexploited.** Explicitly organizing connected evidence into structured reasoning forms has also been explored for complex reasoning tasks (Si et al., 2024). This motivates preserving the internal order of reasoning paths when incorporating them into LLMs.

To address these challenges, we propose **PathSoft**, a hybrid soft prompting framework for KGQA. In the first stage, PathSoft uses a lightweight path ranker, PathMLP, to estimate whether each candidate path can reach a correct answer and retains the top- $k$  paths as evidence. In the second stage, PathSoft employs PathSoftAdapter to encode the selected paths as ordered entity–relation sequences and convert them into continuous soft prompts. Finally, these soft tokens are integrated with the embeddings of the textual prompt for answer generation, enabling the model to exploit both readable path evidence and path structure information.

Experiments on WebQSP and CWQ demonstrate that PathSoft consistently outperforms representative KGQA methods. Further analyses show that filtering unreliable paths and encoding ordered path structure contribute substantially to PathSoft’s overall effectiveness.

Our main contributions are summarized as follows:

- We propose PathSoft, a hybrid soft prompting framework that combines ranked textual reasoning paths with continuous path-structure information for KGQA.
- We design PathMLP and PathSoftAdapter to select reliable reasoning paths and encode their ordered entity–relation structures into continuous soft prompts.
- Experiments on WebQSP and CWQ demonstrate the effectiveness and efficiency of PathSoft, while further analyses validate the contributions of path selection, path-structure encoding, and hybrid prompting.

## 2 Related Work

### 2.1 Knowledge Graph Question Answering

KGQA methods are broadly categorized into semantic parsing methods and graph retrieval methods. Semantic parsing methods map natural language questions to executable logical forms, whereas graph retrieval methods retrieve question-relevant graph evidence for answer prediction. Within the graph retrieval line, GraftNet (Sun et al., 2018) integrates KB facts and text into a graph for answer prediction, SR+NSM (Zhang et al., 2022) retrieves question-specific subgraphs for subsequent reasoning, and ReaRev (Mavromatis and Karypis, 2022) guides KG reasoning with adaptively updated question instructions. These approaches generally rank candidate answer entities based on graph evidence. In contrast, PathSoft uses retrieved reasoning paths as evidence to guide LLM reasoning during answer generation.

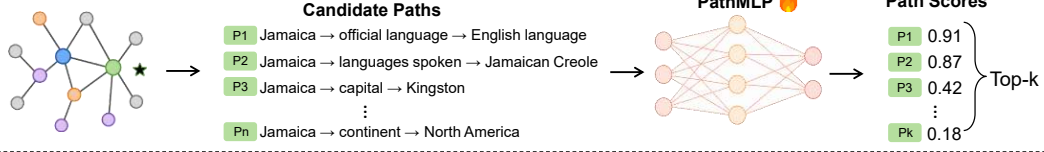
### 2.2 KG-Augmented LLM Reasoning

Recent KG-augmented LLM methods enhance LLM reasoning through KG evidence retrieval, iterative graph exploration, or graph-constrained decoding. One line of work focuses on path-based reasoning. RoG (Luo et al., 2024) retrieves reasoning paths to guide LLM reasoning, while ToG (Sun et al., 2024) enables LLMs to iteratively explore knowledge graphs during inference. FiDeLiS (Sui et al., 2025) validates reasoning steps using retrieved paths and deductive scoring. More recently, MIAoG (Zhang et al., 2026b) introduces multi-view instructed adaptive reasoning, enabling LLMs to plan, evaluate, and adapt reasoning paths from a global perspective. GCR (Luo et al., 2025) further constrains LLM decoding with KG structures. Beyond KGQA, Reason-on-Graph (Si et al., 2026) constructs an evidence knowledge graph and isolates causal reasoning paths to guide structured LLM reasoning.

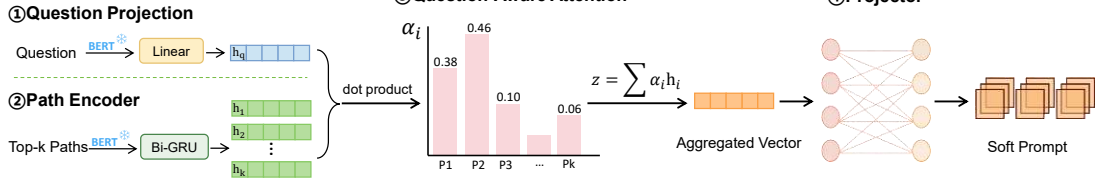
Another line of work focuses on subgraph-based retrieval. SubgraphRAG (Li et al., 2025) retrieves query-relevant subgraphs to support LLM answer prediction, while D-RAG (Gao et al., 2025) develops differentiable subgraph retrieval for KGQA. KELDaR (Li et al., 2024) further decomposes questions and retrieves atomic KG evidence for LLM reasoning. Beyond path-based reasoning and subgraph-based retrieval, recent studies further improve KG-augmented LLM reasoning through graph neural retrieval in GNN-RAG (Mavromatis

**Question** What language do Jamaican people speak?

### Stage 1: Candidate Path Ranking



### Stage 2: PathSoftAdapter



### Stage 3: Hybrid Prompting

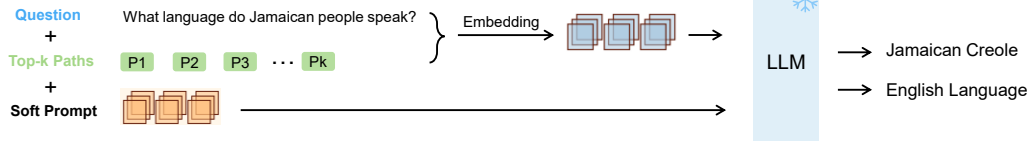


Figure 1: Overview of PathSoft. Candidate reasoning paths are first ranked by PathMLP, and the selected paths are encoded by PathSoftAdapter into soft prompts. The resulting soft prompts are prepended to the embeddings of a textual prompt containing the selected paths and question, forming a hybrid input for the frozen LLM.

and Karypis, 2025), strategic multi-model collaboration in EffiQA (Dong et al., 2025), and enhanced web and KG retrieval in EWEK-QA (Dehghan et al., 2024).

These methods typically linearize triples, subgraphs, or paths into textual prompts for LLM reasoning. PathSoft also retains textualized paths as readable evidence, but further encodes the ordered structure of ranked reasoning paths into continuous soft prompts, forming a hybrid representation for answer generation.

## 2.3 Soft Prompting for KG-Augmented LLMs

Soft prompting is a parameter-efficient adaptation technique that prepends trainable continuous vectors to the input embeddings of a frozen language model, instead of manually designing discrete textual prompts or updating all model parameters. These continuous prompts are optimized for downstream tasks and can encode task-specific or external knowledge in the model’s embedding space. Recent KG-augmented LLM studies have explored this paradigm from different perspectives. LightPROF (Ao et al., 2025) retrieves reasoning graphs and maps them into prompt embeddings through a Knowledge Adapter. QAP (Liu et al., 2025) derives question-aware KG soft prompts to enhance LLMs in multiple-choice QA. OntoSCPrompt (Jiang et al.,

2025) incorporates ontology-level information into prompt learning for improving KGQA generalization.

Rather than encoding reasoning graphs, general KG representations, or ontology-level information, PathSoft selects reliable candidate paths and explicitly encodes their ordered entity–relation structure into soft prompts for generative KGQA.

## 3 Preliminary

Let  $G = \{(h, r, t) \mid h, t \in E, r \in R\}$  denote a knowledge graph, where  $E$  and  $R$  are the entity and relation sets. Given a question  $q$ , KGQA aims to predict an answer set  $A = \{a_1, \dots, a_m\}$  using relevant facts in  $G$ . In this work, we focus on generative KGQA, where the model generates textual answers rather than ranking candidate entities.

For each question, we use a pre-constructed local graph and enumerate candidate reasoning paths from topic entities. Let  $P_q = \{p_i\}_{i=1}^{N_q}$  denote the candidate path set for question  $q$ , where each path is represented as  $p_i = (e_0^i, r_1^i, e_1^i, \dots, r_{H_i}^i, e_{H_i}^i)$ , with  $e_0^i$  being a topic entity and  $e_{H_i}^i$  being the terminal entity of the path. PathSoft selects reliable candidate paths, uses their textualized forms as readable evidence, and encodes their ordered structures into soft prompts for LLM answer generation.

## 4 The PathSoft Framework

Given a question  $q$  and candidate reasoning paths  $\mathcal{P}_q$  enumerated from the question-relevant local graph, PathSoft performs KGQA through a hybrid soft prompting framework with frozen LLMs. In the first stage, a lightweight PathMLP estimates the answer reachability of each candidate path and selects the top- $k$  paths as evidence. In the second stage, PathSoftAdapter transforms the selected paths into continuous soft prompts. Finally, these soft prompts are prepended to the embeddings of a textual prompt containing the textualized paths and the question, forming a hybrid representation for answer generation. Throughout the framework, the backbone LLM remains frozen, and only PathMLP and PathSoftAdapter are trained. Figure 1 illustrates the overall architecture of PathSoft.

### 4.1 Candidate Path Ranking with PathMLP

In the first stage, PathSoft enumerates candidate reasoning paths from the question-relevant knowledge graph and ranks them with PathMLP, a lightweight path ranker that estimates answer reachability. A graph may contain many candidate paths, but only a small subset is likely to reach correct answers. Directly feeding all paths into the LLM would increase the context length and introduce noise. PathMLP is designed to estimate how likely each candidate path is to reach an answer and to retain the top- $k$  paths for the subsequent PathSoftAdapter.

Given a question  $q$  and its candidate path set  $\mathcal{P}_q = \{p_i\}_{i=1}^{N_q}$ , we formulate path ranking as binary classification over question-path pairs  $(q, p_i)$ . Let  $e_{H_i}^i$  denote the terminal entity of path  $p_i$ . If  $e_{H_i}^i$  matches any gold answer, the path is labeled as positive, with  $y_i = 1$ ; otherwise, it is labeled as negative, with  $y_i = 0$ . This supervision only requires the terminal entity to match a gold answer, without relying on annotated reasoning chains.

To represent questions and paths, we use a text encoder, GTE-Large-en-v1.5 (Zhang et al., 2024), to encode the question text and the textualized path into a question vector  $\mathbf{q}$  and each textualized path into a path vector  $\mathbf{p}_i$ . We then construct the following interaction feature:

$$\mathbf{x}_i = [\mathbf{q}; \mathbf{p}_i; \mathbf{q} - \mathbf{p}_i; \mathbf{q} \odot \mathbf{p}_i], \quad (1)$$

where  $[\cdot; \cdot]$  denotes vector concatenation and  $\odot$  denotes element-wise multiplication. The difference term captures question-path mismatch, whereas the

element-wise product captures their semantic alignment.

PathMLP maps the interaction vector  $\mathbf{x}_i$  to a scalar score through an MLP with two hidden layers, followed by a sigmoid function:

$$s_i = \sigma(\text{MLP}(\mathbf{x}_i)), \quad (2)$$

where  $s_i \in (0, 1)$  is interpreted as the probability that path  $p_i$  reaches a correct answer. The model is trained with the binary cross-entropy loss:

$$\mathcal{L}_{rank} = - \sum_i [y_i \log s_i + (1 - y_i) \log(1 - s_i)]. \quad (3)$$

After training, PathMLP scores all candidate paths for each question and ranks them in descending order. The top- $k$  paths are retained as evidence for the second stage.

### 4.2 PathSoftAdapter

After PathMLP selects the top- $k$  candidate paths, PathSoftAdapter converts the selected path evidence into continuous soft prompts for the frozen LLM. Unlike PathMLP, which encodes each textualized path into a single vector, PathSoftAdapter explicitly models the ordered entity-relation sequence within each path. This helps preserve the sequential structure of reasoning paths when converting the selected paths into soft prompts.

For a question  $q$ , let  $\hat{\mathcal{P}}_q = \{p_i\}_{i=1}^k$  denote the selected path set. We use a frozen BERT encoder (Devlin et al., 2019) to encode both the question and the paths. The question representation is taken from the BERT [CLS] vector and denoted as  $\mathbf{c}_q$ .

For each path  $p_i$ , we treat its entities and relations as an ordered sequence of path elements and obtain BERT-based element embeddings:

$$\mathbf{X}_i = [\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,L_i}], \quad (4)$$

where  $\mathbf{x}_{i,j}$  is the BERT [CLS] embedding of the  $j$ -th path element, and  $L_i$  is the number of retained path elements excluding padding.

**Question projection.** The question vector  $\mathbf{c}_q$  is first projected into the adapter hidden space:

$$\mathbf{h}_q = \mathbf{W}_{qp} \mathbf{c}_q + \mathbf{b}_{qp}, \quad (5)$$

where  $\mathbf{h}_q$  is the projected question representation.

**Path encoder.** Each path embedding sequence  $\mathbf{X}_i$  is encoded with a bidirectional GRU. The forward GRU reads the path from the topic entity to the terminal entity, while the backward GRU reads it in the reverse direction. We concatenate the final hidden states from both directions and project the concatenated representation into the adapter hidden space:

$$\mathbf{h}_i = \mathbf{W}_{pe}[\vec{\mathbf{g}}_{i,L_i}; \overleftarrow{\mathbf{g}}_{i,1}] + \mathbf{b}_{pe}, \quad (6)$$

where  $\mathbf{h}_i$  denotes the representation of path  $p_i$ . This encoder preserves the entity–relation order of each reasoning path while producing one vector for each selected path.

**Question-aware path attention.** PathSoftAdapter aggregates the  $k$  path representations according to the question. Specifically, the projected question representation  $\mathbf{h}_q$  is used as the query, and the path representations  $\{\mathbf{h}_i\}_{i=1}^k$  are used as keys and values in multi-head scaled dot-product attention:

$$\mathbf{z} = \text{Attn}(\mathbf{h}_q, \{\mathbf{h}_i\}_{i=1}^k). \quad (7)$$

The resulting vector  $\mathbf{z}$  is a question-aware aggregation of the selected path evidence, allowing the adapter to emphasize different paths for different questions.

**Soft prompt projector.** Finally, the aggregated representation  $\mathbf{z}$  is mapped into the embedding space of the frozen LLM:

$$\mathbf{S} = \text{Projector}(\mathbf{z}) \in \mathbb{R}^{n \times d_{\text{LLM}}}, \quad (8)$$

where  $n$  is the number of soft prompts and  $d_{\text{LLM}}$  is the embedding dimension of the frozen LLM. The resulting matrix  $\mathbf{S}$  contains the soft prompts that summarize the selected path evidence and are prepended to the embeddings of the textual prompt.

### 4.3 Hybrid Prompting and Optimization

The soft prompts generated by PathSoftAdapter provide embedding-level representations of the selected path evidence, but they do not explicitly expose entity names or relation labels to the LLM. Therefore, PathSoft uses a hybrid input that combines continuous soft prompts with a textual prompt. Specifically, the textual prompt contains the textualized paths and the original question, while the soft prompts are prepended to its embeddings. Under this formulation, the textual

prompt provides readable entity–relation evidence, whereas the soft prompts provide question-aware path-structure information at the embedding level.

Let  $\mathbf{S} \in \mathbb{R}^{n \times d_{\text{LLM}}}$  denote the soft prompts produced by PathSoftAdapter, and let  $T(q, \tilde{\mathcal{P}}_q)$  denote the textual prompt constructed from the question  $q$  and the selected path set  $\tilde{\mathcal{P}}_q$ . The input embeddings to the frozen LLM are formed by prepending the soft prompts to the embeddings of the textual prompt:

$$\mathbf{H} = [\mathbf{S}; \text{Embed}(T(q, \tilde{\mathcal{P}}_q))], \quad (9)$$

where  $\text{Embed}(\cdot)$  is the token embedding layer of the frozen LLM. The textual prompt contains a task instruction, the selected paths, the original question, and an answer format instruction.

**Training.** PathMLP is trained separately with the ranking objective described in Section 4.1. In the second stage, the backbone LLM remains frozen, and only PathSoftAdapter is optimized. Given a gold answer sequence  $a = (a_1, \dots, a_T)$ , we minimize the teacher-forcing loss over answer-token positions:

$$\mathcal{L}_{\text{gen}} = - \sum_{t \in Y} \log p(a_t \mid a_{<t}, \mathbf{S}, T(q, \tilde{\mathcal{P}}_q)), \quad (10)$$

where  $Y$  denotes the answer-token positions. Tokens corresponding to the soft prompts and the textual prompt are masked out.

**Inference.** At inference time, PathSoft uses the trained PathMLP to score candidate paths and retain the top- $k$  paths  $\tilde{\mathcal{P}}_q$ . PathSoftAdapter then converts the selected paths into soft prompts  $\mathbf{S}$ , which are prepended to the embeddings of the textual prompt  $T(q, \tilde{\mathcal{P}}_q)$ . The frozen LLM generates answers from the resulting hybrid representation, and the final predictions are obtained by extracting lines prefixed with `ans:`.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** We evaluate PathSoft on two widely used KGQA benchmarks, WebQSP (Yih et al., 2016) and CWQ (Talmor and Berant, 2018), both of which are grounded in Freebase (Bollacker et al., 2008). Following RoG (Luo et al., 2024), we use the same question splits and Freebase-grounded local graphs. We enumerate candidate reasoning

| Method                                   | WebQSP       |              | CWQ          |              | Avg.         |              |
|------------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                          | Hit          | F1           | Hit          | F1           | Hit          | F1           |
| <i>Pure LLM</i>                          |              |              |              |              |              |              |
| Qwen2.5-7B-Instruct                      | 44.78        | 29.89        | 26.22        | 21.09        | 35.50        | 25.49        |
| Llama-3.1-8B-Instruct                    | 55.50        | 34.80        | 39.65        | 22.40        | 47.58        | 28.60        |
| GPT-4o-mini                              | 55.41        | 60.28        | 33.76        | 39.85        | 44.59        | 50.07        |
| <i>Graph Reasoning</i>                   |              |              |              |              |              |              |
| GraftNet                                 | 66.70        | 62.40        | 36.80        | 32.70        | 51.75        | 47.55        |
| SR+NSM                                   | 68.90        | 64.10        | 50.20        | 47.10        | 59.55        | 55.60        |
| ReaRev                                   | 76.40        | 70.90        | 52.90        | 47.80        | 64.65        | 59.35        |
| <i>KG+LLM with Tuned or Adapted LLMs</i> |              |              |              |              |              |              |
| KD-CoT                                   | 68.60        | 52.50        | 55.70        | –            | 62.15        | –            |
| RoG (Llama-2-7B)                         | <u>85.70</u> | 70.80        | 62.60        | 56.20        | 74.15        | 63.50        |
| GNN-RAG                                  | <u>85.70</u> | 71.30        | 66.80        | <u>59.40</u> | <u>76.25</u> | <u>65.35</u> |
| GCR                                      | 83.71        | 76.43        | 67.47        | 51.19        | 75.59        | 63.81        |
| <i>Frozen or Prompting-based KG+LLM</i>  |              |              |              |              |              |              |
| EffiQA                                   | 82.90        | –            | <u>69.50</u> | –            | 76.20        | –            |
| EWEK-QA                                  | 71.30        | –            | <u>52.50</u> | –            | 61.90        | –            |
| LightPROF                                | 74.14        | –            | 49.64        | –            | 61.89        | –            |
| ToG                                      | 67.94        | 39.82        | 41.66        | 40.19        | 54.80        | 40.01        |
| FiDeLiS                                  | 73.05        | 69.21        | 63.50        | 58.58        | 68.28        | 63.90        |
| SubgraphRAG (Qwen2.5-7B-Inst.)           | 83.48        | 71.63        | 50.84        | 46.99        | 67.16        | 59.31        |
| SubgraphRAG (Llama-3.1-8B-Inst.)         | 85.57        | <u>77.80</u> | 57.07        | 52.30        | 71.32        | 65.05        |
| PathSoft (Qwen2.5-7B-Inst.)              | 86.17        | <u>76.69</u> | 72.56        | 59.09        | 79.37        | 67.89        |
| PathSoft (Llama-3.1-8B-Inst.)            | <b>88.81</b> | <b>78.27</b> | <b>73.21</b> | <b>62.07</b> | <b>81.01</b> | <b>70.17</b> |

Table 1: Main results on WebQSP and CWQ. Best overall results are shown in bold, and the strongest baseline results are underlined. Avg. Hit and Avg. F1 denote the average scores over WebQSP and CWQ. Inst. denotes the instruction-tuned version.

paths from topic entities within two hops for WebQSP and three hops for CWQ. Appendix B provides statistics of the full candidate path pools before PathMLP ranking. PathSoft then performs path ranking, evidence filtering, and answer generation over these candidate paths.

**Baselines.** We compare PathSoft with four groups of baselines following the order in Table 1: pure LLMs, including Qwen2.5-7B-Instruct (Qwen Team, 2024), Llama-3.1-8B-Instruct (Llama Team, 2024), and GPT-4o-mini (OpenAI, 2024); graph reasoning methods, including GraftNet (Sun et al., 2018), SR+NSM (Zhang et al., 2022), and ReaRev (Mavromatis and Karypis, 2022); KG+LLM methods with tuned or adapted LLMs, including KD-CoT (Wang et al., 2023), RoG (Luo et al., 2024), GNN-RAG (Mavromatis and Karypis, 2025), and GCR (Luo et al., 2025); and frozen or prompting-based KG+LLM methods, including EffiQA (Dong et al., 2025), EWEK-QA (Dehghan et al., 2024), LightPROF (Ao et al., 2025), ToG (Sun et al., 2024), FiDeLiS (Sui et al., 2025), and SubgraphRAG (Li et al., 2025).

**Evaluation Metrics.** Following prior KGQA studies, we report Hit and token-level F1. Hit

measures whether predicted answers match gold answers. Token-level F1 measures word overlap between predicted and gold answers. All scores are averaged over the test set.

**Implementation Details.** For path ranking, we use GTE-Large-en-v1.5 (Zhang et al., 2024) as the frozen text encoder and cache all question and path embeddings before training PathMLP. PathMLP is trained with binary cross-entropy and contains approximately 1.07M trainable parameters. For PathSoftAdapter, we use a frozen BERT encoder (Devlin et al., 2019) to encode questions and individual path elements, including entities and relations. PathMLP retains the top-20 paths, while PathSoftAdapter uses 16 soft prompt tokens, with a sensitivity analysis provided in Appendix C.1. We use Qwen2.5-7B-Instruct (Qwen Team, 2024) and Llama-3.1-8B-Instruct (Llama Team, 2024) as LLM backbones. In the second stage, all LLM parameters are frozen, and only PathSoftAdapter is updated. During inference, we set the maximum number of generated tokens to 64.

## 5.2 Main Results

Table 1 reports the main results on WebQSP and CWQ. PathSoft achieves the best overall perfor-

| Variant         | Rank | Soft | WebQSP       |              | CWQ          |              |
|-----------------|------|------|--------------|--------------|--------------|--------------|
|                 |      |      | Qwen         | Llama        | Qwen         | Llama        |
| LLM-only        | ×    | ×    | 44.78        | 55.50        | 26.22        | 39.65        |
| Text-only       | ✓    | ×    | 80.52        | 83.16        | 45.57        | 57.50        |
| Soft-only       | ×    | ✓    | 71.24        | 78.30        | 56.56        | 64.46        |
| <b>PathSoft</b> | ✓    | ✓    | <b>86.17</b> | <b>88.81</b> | <b>72.56</b> | <b>73.21</b> |

Table 2: Ablation results of PathSoft in terms of Hit. Rank denotes Stage 1 path selection with PathMLP, and Soft denotes Stage 2 soft prompts generated by PathSoftAdapter.

mance across both datasets. With Llama-3.1-8B-Instruct, PathSoft obtains 88.81% Hit and 78.27% F1 on WebQSP, and 73.21% Hit and 62.07% F1 on CWQ. Compared with the strongest baselines, the gains are more pronounced on CWQ, suggesting that PathSoft is especially effective for complex compositional questions that require reliable evidence selection.

Under the same backbone, PathSoft consistently outperforms SubgraphRAG on both datasets. The improvement is particularly large on CWQ: PathSoft improves Hit from 50.84% to 72.56% with Qwen2.5-7B-Instruct and from 57.07% to 73.21% with Llama-3.1-8B-Instruct. On WebQSP, PathSoft also brings stable gains over SubgraphRAG under both backbones. F1 follows the same trend, as shown in Table 1. These results indicate that path-level evidence selection and soft prompt generation provide benefits beyond textualized KG evidence alone, especially for complex multi-hop questions.

We further evaluate PathSoft under matched LLM backbones and identical PathMLP-selected top-20 path evidence, where it remains consistently superior to representative baselines (Appendix C.2 and Appendix C.3). Additional evaluations with a larger 14B backbone, an additional KGQA benchmark, and the medical-domain BioGraphletQA dataset are reported in Appendix D.

### 5.3 Ablation Study

Table 2 reports the ablation results in terms of Hit. The LLM-only setting performs the worst across datasets and backbones, showing that LLMs struggle without external KG evidence. Adding ranked textual paths substantially improves performance, especially on WebQSP, where Text-only improves Qwen2.5-7B-Instruct from 44.78% to 80.52% and Llama-3.1-8B-Instruct from 55.50% to 83.16%. This confirms that selected reasoning paths provide useful readable evidence.

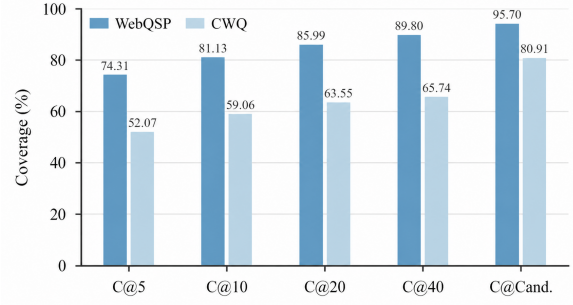


Figure 2: PathMLP ranking quality on WebQSP and CWQ. C@k denotes the coverage of answer-reaching paths among the top- $k$  ranked paths, and C@Cand. denotes the full candidate-pool oracle coverage before PathMLP ranking.

In Soft-only, we remove the PathMLP ranking module and randomly sample 20 candidate paths for PathSoftAdapter to generate soft prompts. Soft-only also improves over LLM-only, particularly on CWQ, suggesting that soft prompts can capture informative signals from reasoning paths even without ranked path selection. The full PathSoft model achieves the best results in all settings, with especially large gains over Text-only on CWQ: Hit improves from 45.57% to 72.56% with Qwen2.5-7B-Instruct and from 57.50% to 73.21% with Llama-3.1-8B-Instruct. These results show that ranked textual paths and soft prompts provide complementary evidence. Additional controlled ablations further confirm the complementary contributions of textual and soft evidence and the importance of ordered path structure; detailed results are provided in Appendix C.4 and Appendix C.5.

### 5.4 Path Selection Analysis

**Ranking quality of PathMLP.** Figure 2 evaluates whether PathMLP ranks answer-reaching paths near the top of the candidate list. Coverage increases consistently as  $k$  grows on both datasets, showing that PathMLP provides effective path prioritization rather than simply relying on a large candidate set. On WebQSP, C@20 reaches 85.99, approaching the full candidate-pool oracle coverage of 95.70, indicating that most answer-reaching paths can be recovered within a small top-ranked subset. On CWQ, C@20 reaches 63.55, compared with a full candidate-pool oracle coverage of 80.91. The larger gap suggests that CWQ is more challenging not only for ranking but also for candidate path construction, as complex multi-hop questions require more complete and less noisy path candi-

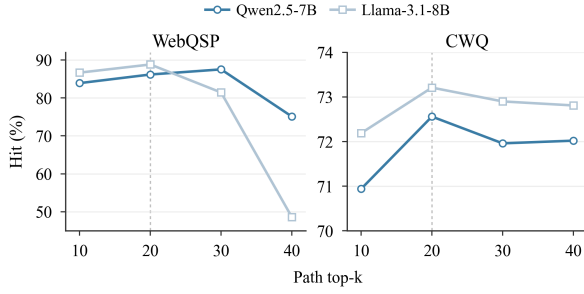


Figure 3: Effect of the number of selected paths. We vary top- $k$  from 10 to 40 and report Hit on WebQSP and CWQ with Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct backbones.

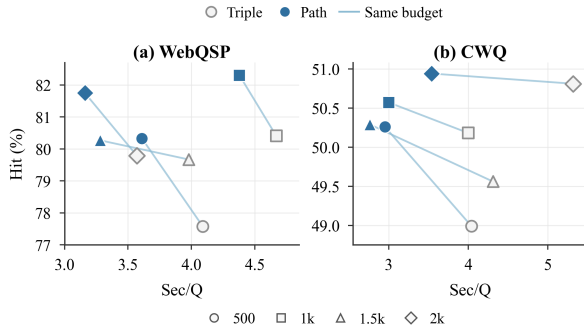


Figure 4: Performance and efficiency comparison between path evidence and triple evidence under fixed token budgets. Each connected pair uses the same token budget. Higher Hit and lower Sec/Q indicate better performance.

dates. Overall, these results show that PathMLP supplies high-quality evidence candidates for PathSoftAdapter, while also highlighting the importance of stronger path construction for complex questions. Additional analyses of answer-reaching path ranks and error sources are provided in Appendix E.

**Effect of the number of selected paths.** Figure 3 analyzes the effect of top- $k$  on answer generation. Overall,  $k = 20$  provides stable performance across datasets and backbones. On WebQSP, increasing  $k$  does not lead to consistent gains and even causes a clear drop with Llama-3.1-8B-Instruct, suggesting that excessive paths may introduce noisy or irrelevant evidence and distract generation. In contrast, CWQ is more stable across different values of  $k$ , with the best results still observed around  $k = 20$ . These results show that PathSoft needs to balance evidence coverage against path noise, and we therefore use  $k = 20$  as the default setting in the main experiments.

| Method              | Tuned | Params | Hit          | Tokens   | Sec/Q       | Calls |
|---------------------|-------|--------|--------------|----------|-------------|-------|
| GCR                 | Yes   | 8.03B  | 67.47        | 112.83   | 1.50        | 2     |
| ToG (GPT-4o-mini)   | No    | 0      | 41.66        | 1984.93  | 9.50        | 2.89  |
| FiDeLiS (GPT-3.5)   | No    | 0      | 63.50        | 45926.43 | 180.96      | 16.37 |
| SubgraphRAG (Qwen)  | No    | MLP    | 50.84        | 2635.45  | 4.61        | 1     |
| SubgraphRAG (Llama) | No    | MLP    | 57.07        | 2543.65  | 6.48        | 1     |
| Text-only (Qwen)    | No    | 1.07M  | 45.57        | 645.44   | 2.10        | 1     |
| Text-only (Llama)   | No    | 1.07M  | 57.50        | 625.92   | 2.09        | 1     |
| PathSoft (Qwen)     | No    | 63.46M | <b>72.56</b> | 661.44   | 1.38        | 1     |
| PathSoft (Llama)    | No    | 71.85M | <b>73.21</b> | 641.92   | <b>1.26</b> | 1     |

Table 3: Overall efficiency comparison on CWQ. Tokens denotes the average number of input tokens, Sec/Q denotes inference time per question, and Calls denotes the average number of LLM calls per question.

## 5.5 Efficiency Analysis

**Path evidence versus triple evidence.** We compare path evidence with triple evidence under fixed token budgets. For a fair comparison, both settings use the same plain textual instruction and differ only in how KG evidence is presented to the LLM. The detailed prompt templates are provided in Appendix A. As shown in Figure 4, path evidence consistently achieves higher Hit and lower Sec/Q than triple evidence on both WebQSP and CWQ. This suggests that reasoning paths provide more compact and ordered KG evidence than unordered triples, allowing LLMs to use graph information more effectively and efficiently.

**Overall efficiency comparison.** We further compare the overall efficiency of PathSoft with representative KG-augmented LLM methods on CWQ. As shown in Table 3, PathSoft achieves the highest Hit while keeping the backbone LLM frozen and requiring only one LLM call per question. Compared with iterative prompting methods such as ToG and FiDeLiS, PathSoft uses substantially fewer input tokens and achieves much lower inference latency. Under the same LLM backbone, PathSoft is also more efficient than SubgraphRAG: with Llama-3.1-8B-Instruct, it improves Hit from 57.07% to 73.21%, while reducing the average input length from 2,543.65 to 641.92 tokens and Sec/Q from 6.48 to 1.26. These results show that compact path selection and soft prompt-based evidence injection improve KGQA performance without sacrificing efficiency. The WebQSP results in Appendix F show a similar trend.

## 5.6 Case Study

Table 4 provides a qualitative example of how PathSoft handles relation-level distractors. Although the rank-1 path receives the highest PathMLP score

|             |                                                                                     |
|-------------|-------------------------------------------------------------------------------------|
| Question    | Where did the actor nominated for the award for <i>The Catcher in the Rye</i> die?  |
| Gold answer | Cornish                                                                             |
| Text-only   | ans: Natural causes                                                                 |
| PathSoft    | ans: Cornish                                                                        |
| Rank 1      | <i>The Catcher in the Rye</i> → J. D. Salinger<br>→ cause of death → Natural causes |
| Rank 2      | <i>The Catcher in the Rye</i> → J. D. Salinger<br>→ place of death → Cornish        |

Table 4: A CWQ case with identical top-20 ranked paths for Text-only and PathSoft.

and reaches the correct person, J. D. Salinger, it follows the *cause of death* relation and therefore answers a different question. The text-only variant tends to copy the salient endpoint of this distractor path and predicts “Natural causes”. In contrast, PathSoft predicts “Cornish” from the rank-2 path, which matches the required *place of death* relation. This suggests that the soft prompts help the LLM use ordered relation information rather than relying only on high-ranked textual evidence. A rank-conditioned analysis further examines this behavior quantitatively in Appendix E.3, and the full case analysis is provided in Appendix G.

## 6 Conclusion

We proposed PathSoft, a hybrid soft prompting KGQA framework that addresses noisy path selection and the insufficient use of reasoning path structure. First, PathSoft ranks candidate reasoning paths with PathMLP to select reliable evidence. Second, PathSoft encodes the ordered entity–relation structures of the selected paths into continuous soft prompts with PathSoftAdapter. Finally, PathSoft prepends these soft prompts to the embeddings of a textual prompt containing the selected paths and question, enabling the LLM to use both readable path evidence and path-structure information. Experiments on WebQSP and CWQ show that PathSoft improves answer accuracy while maintaining low inference cost. Future work will explore stronger candidate path construction and broader knowledge-intensive reasoning tasks.

## Limitations

PathSoft has several limitations. First, its performance depends on the quality and coverage of candidate reasoning paths. If answer-reaching paths are missing from the candidate set, PathSoft cannot recover the missing evidence. Second, although our additional experiments provide preliminary

evidence of generalization across model scales, KGQA benchmarks, and application domains, the evaluations on GrailQA and BioGraphletQA are conducted only on randomly sampled subsets. Full-scale evaluation across more knowledge graphs, domains, question types, and backbone families is still needed to establish broader generality. Third, although PathSoft avoids full LLM fine-tuning, it still introduces additional trainable modules and requires candidate path enumeration, which bring extra preprocessing and training costs.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62376130, No.62402258, No.72501151), Natural Science Foundation of Shandong Province (No.ZR2024MF088), Program of New Twenty Policies for Universities of Jinan (No.202333008), the Pilot Project for Integrated Innovation of Science, Education, and Industry of Qilu University of Technology (Shandong Academy of Sciences) (Grant No. 2025ZDZX01), and the Key Research and Development Program of Shandong Province (Major Science and Technology Innovation Project) (No.2026CXGC010604).

## References

- Garima Agrawal, Tharindu Kumarage, Zeyad Alghamdi, and Huan Liu. 2024. Can knowledge graphs reduce hallucinations in LLMs? : A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3947–3960.
- Tu Ao, Yanhua Yu, Yuling Wang, Yang Deng, Zirui Guo, Liang Pang, Pinghui Wang, Tat-Seng Chua, Xiao Zhang, and Zhen Cai. 2025. LightPROF: A lightweight reasoning framework for large language model on knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23424–23432.
- 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*, pages 1247–1250.
- Mohammad Dehghan, Mohammad Alomrani, Sunyam Bagga, David Alfonso-Hermelo, Khalil Bibi, Abbas Ghaddar, Yingxue Zhang, Xiaoguang Li, Jianye Hao, Qun Liu, Jimmy Lin, Boxing Chen, Prasanna Parthasarathi, Mahdi Biparva, and Mehdi Reza-gholizadeh. 2024. EWEK-QA : Enhanced web and

- efficient knowledge graph retrieval for citation-based question answering systems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 14169–14187.
- 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*, pages 4171–4186.
- Zixuan Dong, Baoyun Peng, Yufei Wang, Jia Fu, Xiaodong Wang, Xin Zhou, Yongxue Shan, Kangchen Zhu, and Weiguo Chen. 2025. EffiQA: Efficient question-answering with strategic multi-model collaboration on knowledge graphs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7180–7194.
- Guangze Gao, Zixuan Li, Chunfeng Yuan, Jiawei Li, Wu Jianzhuo, Yuehao Zhang, Xiaolong Jin, Bing Li, and Weiming Hu. 2025. D-RAG: Differentiable retrieval-augmented generation for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35398–35417.
- Longquan Jiang, Junbo Huang, Cedric Möller, and Ricardo Usbeck. 2025. Ontology-guided, hybrid prompt learning for generalization in knowledge graph question answering. In *2025 19th International Conference on Semantic Computing (ICSC)*, pages 28–35.
- Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *Proceedings of the International Conference on Learning Representations*, pages 6061–6089.
- Yading Li, Dandan Song, Changzhi Zhou, Yuhang Tian, Hao Wang, Ziyi Yang, and Shuhao Zhang. 2024. A framework of knowledge graph-enhanced large language model based on question decomposition and atomic retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11472–11485.
- Can Lin, Zhengwang Jiang, Ling Zheng, Qi Zhao, Yuhang Zhang, Qi Song, and Wangqiu Zhou. 2025. RJE: A retrieval-judgment-exploration framework for efficient knowledge graph question answering with LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17277–17294.
- Haochen Liu, Song Wang, Chen Chen, and Jundong Li. 2025. Question-aware knowledge graph prompting for enhancing large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1388–1400.
- Kangjun Liu, Zhenyu Li, Jianlei Wang, Hongjiao Guan, Ying Lian, Guoqiang Chen, Tao Xin, and Wenpeng Lu. 2026. JointCoder: Exploring automated ICD coding on real-world Chinese EHRs with a multi-agent framework. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pages 263–274.
- Llama Team. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Wenpeng Lu, Kangjun Liu, Jianlei Wang, Xueping Peng, Tao Shen, Fa Zhu, Weiyu Zhang, Jiabing Zhu, Tao Xin, and Athanasios V Vasilakos. 2025. Advancing chinese conversation-based patient guidance with a benchmark and knowledge-evolvable assistant. *IEEE Journal of Biomedical and Health Informatics*.
- Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *Proceedings of the International Conference on Learning Representations*, pages 14400–14423.
- Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. 2025. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 41540–41565.
- Costas Mavromatis and George Karypis. 2022. ReaRev: Adaptive reasoning for question answering over knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2447–2458.
- Costas Mavromatis and George Karypis. 2025. GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16682–16699.
- OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiaapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.
- Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Jiasheng Si, Yibo Zhao, Yingjie Zhu, Haiyang Zhu, Wenpeng Lu, and Deyu Zhou. 2024. CHECKWHY: Causal fact verification via argument structure. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 15636–15659.
- Jiasheng Si, Yingjie Zhu, Yeqing Teng, Rui Wang, Tianyi Wang, Weiyu Zhang, Chaoqun Zheng, Wenpeng Lu, Xiaoming Wu, and Deyu Zhou. 2026. Scaffolding thought: Imposing logical structure on LLMs with knowledge graphs for counterfactual generation. *Knowledge-Based Systems*, 346:116177.

- Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. 2025. FiDeLiS: Faithful reasoning in large language models for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8315–8330.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.
- Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *Proceedings of the International Conference on Learning Representations*, pages 3868–3898.
- 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*, pages 641–651.
- Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven CoT: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.
- 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*, pages 201–206.
- Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 5773–5784.
- Jinshuo Zhang, Xiaoding Zhou, Weiyu Zhang, Guoqiang Chen, Ying Lian, Xiaoyang Meng, Yonghe Chen, Hongjiao Guan, Jiasheng Si, and Wenpeng Lu. 2026a. D<sup>2</sup>-RAG: Dual-decision retrieval-augmented generation via multi-dimensional uncertainty and utility-aware decoding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15355–15382.
- Minghan Zhang, Shu Zhao, Zhen Yang, Hongsheng Wu, Yongxing Lin, Haodong Zou, Jie Chen, and Zhen Duan. 2026b. Thinking beyond the local: Multi-view instructed adaptive reasoning in KG-enhanced LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 6173–6188.
- Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

## A Prompt Templates

**Main PathSoft prompt.** For PathSoft, PathSoftAdapter generates soft prompt embeddings that are prepended to the embeddings of the following textual prompt. The soft prompt embeddings are continuous vectors rather than literal text tokens. The textual prompt contains the selected top- $k$  reasoning paths, the question, and the answer-format instruction. During training, the gold target answers are appended after Answer:; during inference, the frozen LLM generates answers after this prefix.

### Main PathSoft Prompt

```
[SOFT PROMPT EMBEDDINGS]

Based on the knowledge graph paths, please answer the
question. Please return formatted answers as a list,
each prefixed with "ans:".

Paths:
{paths}

Question:
{question}

Answer:
```

A concrete example is shown below. Here, [SOFT PROMPT EMBEDDINGS] denotes the 16 continuous soft prompt embeddings generated by PathSoftAdapter and prepended before the textual prompt.

### Example Main PathSoft Prompt

```
[SOFT PROMPT EMBEDDINGS]

Based on the knowledge graph paths, please answer the
question. Please return formatted answers as a list,
each prefixed with "ans:".

Paths:
1. Scarlett Johansson -> celebrities.celebrity.
   sexual_relationships -> m.07bb827 -> celebrities.
   romantic_relationship.celebrity -> Justin Timberlake
2. Scarlett Johansson -> celebrities.celebrity.
   sexual_relationships -> m.04xwldp -> celebrities.
   romantic_relationship.celebrity -> Josh Hartnett
3. Scarlett Johansson -> people.person.spouse_s -> m.0
   ygrd3d -> people.marriage.spouse -> Ryan Reynolds

Question:
Scarlett Johansson dated a famous person who been popular
since 2000, who was he?

Answer:
```

The expected output format is:

### Expected Output Format

```
ans: Justin Timberlake
```

**Text-only prompt.** The text-only variant uses the same textual prompt as PathSoft, but removes the learned soft prompt embeddings. Therefore, the input contains only the selected textualized paths, the question, and the same answer-format instruction.

### Text-only Prompt

```
Based on the knowledge graph paths, please answer the
question. Please return formatted answers as a list,
each prefixed with "ans:".

Paths:
{paths}

Question:
{question}

Answer:
```

**Plain prompts for path-versus-triple comparison.** For the path-versus-triple efficiency comparison, we use plain textual prompts. The path-based and triple-based settings share the same instruction style and differ only in the evidence format. Unlike the main PathSoft and text-only prompts, these plain prompts require the model to output only the answer entity name without the ans: prefix.

### Path evidence prompt.

#### Plain Prompt for Path-based Evidence

```
Based on the following knowledge graph paths, answer the
question.
Output ONLY the answer entity name, nothing else.

Paths:
{path_evidence}

Question: {question}
Answer:
```

A concrete example is shown below.

#### Example Path-based Evidence Prompt

```
Based on the following knowledge graph paths, answer the
question.
Output ONLY the answer entity name, nothing else.

Paths:
1. Scarlett Johansson [celebrities.celebrity.
   sexual_relationships] m.07bb827 [celebrities.
   romantic_relationship.celebrity] Justin Timberlake
2. Scarlett Johansson [celebrities.celebrity.
   sexual_relationships] m.04xwldp [celebrities.
   romantic_relationship.celebrity] Josh Hartnett
3. Scarlett Johansson [people.person.spouse_s] m.0ygrd3d
   [people.marriage.spouse] Ryan Reynolds

Question: Scarlett Johansson dated a famous person who
been popular since 2000, who was he?
Answer:
```

### Triple evidence prompt.

| Dataset | #Test Questions |
|---------|-----------------|
| WebQSP  | 1628            |
| CWQ     | 3531            |

Table 5: Number of test questions.

| Dataset | Split | Avg. Cand. | Med. Cand. | Avg. Hop | Oracle |
|---------|-------|------------|------------|----------|--------|
| WebQSP  | dev   | 3367.47    | 2771.00    | 1.87     | 95.92  |
| WebQSP  | test  | 3380.78    | 2778.00    | 1.87     | 95.70  |
| CWQ     | dev   | 2881.11    | 2368.00    | 1.90     | 81.09  |
| CWQ     | test  | 3083.86    | 2238.00    | 1.90     | 80.91  |

Table 6: Development and test statistics of the full candidate path pools before PathMLP ranking.

### Plain Prompt for Triple-based Evidence

Based on the following knowledge graph triplets, answer the question.  
Output ONLY the answer entity name, nothing else.

Triplets:  
{triple\_evidence}

Question: {question}  
Answer:

A concrete example is shown below.

### Example Triple-based Evidence Prompt

Based on the following knowledge graph triplets, answer the question.  
Output ONLY the answer entity name, nothing else.

Triplets:

- (Scarlett Johansson, celebrities.celebrity.sexual\_relationships, m.07bb827)
- (m.07bb827, celebrities.romantic\_relationship.celebrity, Justin Timberlake)
- (Scarlett Johansson, celebrities.celebrity.sexual\_relationships, m.04xwldp)
- (m.04xwldp, celebrities.romantic\_relationship.celebrity, Josh Hartnett)

Question: Scarlett Johansson dated a famous person who been popular since 2000, who was he?  
Answer:

## B Dataset and Path Statistics

Table 5 reports the number of test questions in WebQSP and CWQ. Table 6 reports development and test statistics of the full candidate path pools before PathMLP ranking. Avg. Cand. and Med. Cand. denote the average and median numbers of candidate paths per question, respectively. Avg. Hop is measured by path hop count. Oracle denotes the percentage of questions with at least one candidate path whose terminal entity reaches a gold answer.

| Soft Prompt Tokens | Hit          | F1           |
|--------------------|--------------|--------------|
| 8                  | 68.00        | 49.56        |
| 16                 | <b>72.56</b> | <b>59.09</b> |
| 32                 | 70.00        | 52.10        |

Table 7: Sensitivity to the number of soft prompt tokens on CWQ with Qwen2.5-7B-Instruct. All settings use the same PathMLP-selected top-20 paths.

| Method          | Qwen Hit / F1               | Llama Hit / F1              |
|-----------------|-----------------------------|-----------------------------|
| <i>WebQSP</i>   |                             |                             |
| GCR             | 72.80 / <u>73.05</u>        | 74.60 / 74.86               |
| SubgraphRAG     | <u>83.48</u> / 71.63        | <u>85.57</u> / <u>77.80</u> |
| LightPROF       | 64.00 / 56.35               | 72.26 / 69.45               |
| FiDeLiS         | 62.00 / 50.28               | 66.67 / 64.45               |
| <b>PathSoft</b> | <b>86.17</b> / <b>76.69</b> | <b>88.81</b> / <b>78.27</b> |
| <i>CWQ</i>      |                             |                             |
| GCR             | <u>55.60</u> / 44.80        | <u>57.80</u> / 46.42        |
| SubgraphRAG     | 50.84 / <u>46.99</u>        | <u>57.07</u> / <u>52.30</u> |
| LightPROF       | 40.06 / 41.61               | 45.37 / 47.00               |
| FiDeLiS         | 50.02 / 45.38               | 54.12 / 50.57               |
| <b>PathSoft</b> | <b>72.56</b> / <b>59.09</b> | <b>73.21</b> / <b>62.07</b> |

Table 8: Comparison with representative baselines under identical LLM backbones. Best overall results are shown in bold, and the strongest baseline results are underlined.

## C Additional Controlled Experiments

### C.1 Sensitivity to the Number of Soft Prompt Tokens

We examine the sensitivity of PathSoft to the number of soft prompt tokens using Qwen2.5-7B-Instruct on CWQ, while keeping the same PathMLP-selected top-20 paths. We vary the number of soft prompt tokens over  $n \in \{8, 16, 32\}$ .

As shown in Table 7, the 16-token setting achieves the best performance in terms of both Hit and F1. Increasing the number of soft prompt tokens from 16 to 32 does not yield further improvements, suggesting that using more soft prompt tokens does not necessarily lead to better performance. We therefore use  $n = 16$  in the main experiments.

### C.2 Comparison under Identical LLM Backbones

The baselines in the main comparison use different LLM backbones, which may confound the comparison across methods. To control for this factor, we re-evaluate GCR, SubgraphRAG, LightPROF, FiDeLiS, and PathSoft using the same Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct backbones.

| Method          | Qwen Hit / F1        | Llama Hit / F1       |
|-----------------|----------------------|----------------------|
| <i>WebQSP</i>   |                      |                      |
| GCR             | 68.44 / 69.00        | 71.50 / 70.64        |
| SubgraphRAG     | 80.03 / 62.90        | 82.00 / 61.56        |
| LightPROF       | 63.11 / 60.13        | 67.24 / 65.58        |
| FiDeLiS         | 79.00 / 56.66        | 79.00 / 64.31        |
| <b>PathSoft</b> | <b>86.17 / 76.69</b> | <b>88.81 / 78.27</b> |
| <i>CWQ</i>      |                      |                      |
| GCR             | 46.50 / 44.62        | 49.50 / 44.10        |
| SubgraphRAG     | 44.50 / 35.74        | 47.00 / 41.53        |
| LightPROF       | 39.28 / 39.78        | 40.70 / 45.47        |
| FiDeLiS         | 55.00 / 44.25        | 49.00 / 50.19        |
| <b>PathSoft</b> | <b>72.56 / 59.09</b> | <b>73.21 / 62.07</b> |

Table 9: Comparison with representative baselines using identical PathMLP-selected top-20 path evidence and matched LLM backbones. Best overall results are shown in bold, and the strongest baseline results are underlined.

| Backbone | Variant   | Textual Paths | Soft Prompts | WebQSP Hit   | CWQ Hit      |
|----------|-----------|---------------|--------------|--------------|--------------|
| Qwen     | Text-only | ✓             | ×            | 80.52        | 45.57        |
|          | Soft-only | ×             | ✓            | 58.60        | 38.80        |
|          | PathSoft  | ✓             | ✓            | <b>86.17</b> | <b>72.56</b> |
| Llama    | Text-only | ✓             | ×            | 83.16        | 57.50        |
|          | Soft-only | ×             | ✓            | 68.20        | 43.40        |
|          | PathSoft  | ✓             | ✓            | <b>88.81</b> | <b>73.21</b> |

Table 10: Controlled ablation of textual paths and soft prompts. All variants use the same PathMLP-selected top-20 paths.

This comparison controls the LLM backbone without additionally enforcing identical input evidence.

As shown in Table 8, PathSoft consistently achieves the best performance across both datasets and both LLM backbones. These results show that the improvements of PathSoft cannot be explained solely by differences in the underlying LLM backbone.

### C.3 Comparison under Identical Path Evidence

To further isolate the effect of how different methods utilize KG evidence, we conduct a stricter controlled comparison using the same PathMLP-selected top-20 paths. Specifically, the same selected paths are provided directly to FiDeLiS and LightPROF, converted into a path-induced subgraph for SubgraphRAG, and converted into the required path/trie representation for GCR. Thus, both the LLM backbone and the source of input evidence are controlled across methods.

As shown in Table 9, PathSoft consistently out-

| Variant                | WebQSP Hit / F1      | CWQ Hit / F1         |
|------------------------|----------------------|----------------------|
| <i>Qwen</i>            |                      |                      |
| <b>PathSoft</b>        | <b>86.17 / 76.69</b> | <b>72.56 / 59.09</b> |
| Shuffled path order    | 85.80 / 76.41        | 70.40 / 58.11        |
| Shuffled path elements | 81.62 / 67.69        | 63.40 / 51.52        |
| Entity-only            | 83.80 / 69.33        | 66.45 / 55.63        |
| Mean-pooling encoder   | 81.00 / 69.59        | 65.07 / 54.67        |
| <i>Llama</i>           |                      |                      |
| <b>PathSoft</b>        | <b>88.81 / 78.27</b> | <b>73.21 / 62.07</b> |
| Shuffled path order    | 87.61 / 75.55        | 69.47 / 60.33        |
| Shuffled path elements | 85.40 / 75.84        | 68.80 / 57.93        |
| Entity-only            | 85.57 / 70.20        | 70.20 / 60.11        |
| Mean-pooling encoder   | 84.20 / 72.60        | 69.60 / 58.23        |

Table 11: Structure-sensitivity analysis of PathSoft. All variants use the same PathMLP-selected top-20 paths.

performs the representative baselines across both datasets and backbones. Since the candidate-path source is held fixed, these results indicate that the improvements are not solely due to PathMLP path selection, but also depend on how PathSoft represents and integrates the selected path evidence.

### C.4 Controlled Hybrid Ablation

To more directly isolate the contributions of textual paths and soft prompts, we conduct a controlled ablation in which all variants use the same PathMLP-selected top-20 paths. Text-only uses only the textualized paths, Soft-only uses only the soft prompts generated from the selected paths, and PathSoft combines both forms of evidence.

As shown in Table 10, PathSoft consistently outperforms both Text-only and Soft-only across datasets and backbones. This indicates that textual paths and soft prompts provide complementary information rather than redundant representations of the same selected evidence.

Note that this controlled Soft-only setting differs from the Soft-only variant in the main ablation study. Here, Soft-only uses the same PathMLP-selected top-20 paths as PathSoft but removes their textualized forms from the LLM input, whereas the main ablation removes PathMLP ranking and randomly samples 20 candidate paths.

### C.5 Sensitivity to Ordered Path Structure

To examine whether PathSoft benefits from the ordered structure of reasoning paths, we conduct structure-sensitivity experiments using the same PathMLP-selected top-20 paths. We consider four variants: shuffling the order of the selected paths, shuffling entity and relation elements within each

| Dataset       | Backbone              | Hit   |
|---------------|-----------------------|-------|
| WebQSP        | Qwen2.5-14B-Instruct  | 90.89 |
| GrailQA       | Qwen2.5-7B-Instruct   | 78.80 |
| BioGraphletQA | Llama-3.1-8B-Instruct | 40.75 |

Table 12: Additional generalization results across model scales, KGQA benchmarks, and domains. GrailQA and BioGraphletQA results are evaluated on randomly sampled subsets.

path, removing relation labels, and replacing the Bi-GRU path encoder with mean pooling.

As shown in Table 11, shuffling the ranked path order causes moderate degradation, while disrupting the entity–relation order within each path results in larger performance drops. Removing relation labels or replacing the Bi-GRU encoder with mean pooling also consistently reduces performance, particularly on CWQ. These results show that PathSoft benefits from ordered entity–relation structure rather than treating the selected paths as unordered evidence.

## D Additional Generalization Results

We further evaluate PathSoft beyond the primary settings in terms of model scale, an additional KGQA benchmark, and a domain-specific dataset. Recent studies have explored LLMs in healthcare applications such as automated ICD coding (Liu et al., 2026) and patient guidance (Lu et al., 2025), motivating our evaluation on the medical-domain BioGraphletQA dataset.

As shown in Table 12, PathSoft achieves 90.89% Hit on WebQSP with Qwen2.5-14B-Instruct, 78.80% on 500 sampled GrailQA questions with Qwen2.5-7B-Instruct, and 40.75% on 2,000 sampled BioGraphletQA questions with Llama-3.1-8B-Instruct. These results provide preliminary evidence of generalization across model scales, KGQA benchmarks, and application domains.

## E Additional Path Analysis

### E.1 Distribution of Answer-Reaching Path Ranks

We analyze where the first answer-reaching path appears among the PathMLP-ranked paths on CWQ. The first answer-reaching path occurs at rank 1 for 32.39% of the test questions, at ranks 2–5 for 19.69%, at ranks 6–10 for 6.98%, and at ranks 11–20 for 4.49%. Among questions whose top-

| Error Source                                               | Qwen  | Llama |
|------------------------------------------------------------|-------|-------|
| No answer-reaching path in candidate pool                  | 31.78 | 32.03 |
| Answer-reaching path missed by top-20                      | 48.14 | 49.10 |
| Top-20 contains answer-reaching path, prediction incorrect | 20.08 | 18.88 |

Table 13: Distribution of error sources among incorrect predictions on CWQ (%).

| Rank         | Text-only<br>Hit / F1 | PathSoft<br>Hit / F1 | Improvement<br>Hit / F1 |
|--------------|-----------------------|----------------------|-------------------------|
| <i>Qwen</i>  |                       |                      |                         |
| Rank 1       | 80.18 / 64.03         | 98.25 / 91.88        | +18.07 / +27.85         |
| Ranks 2–20   | 54.88 / 36.50         | 84.14 / 55.44        | <b>+29.26 / +18.94</b>  |
| <i>Llama</i> |                       |                      |                         |
| Rank 1       | 86.58 / 71.89         | 97.81 / 92.34        | +11.23 / +20.45         |
| Ranks 2–20   | 70.74 / 46.05         | 86.05 / 62.11        | <b>+15.31 / +16.06</b>  |

Table 14: Rank-conditioned comparison on CWQ. Questions are grouped by the rank of the first answer-reaching path within the PathMLP-selected top-20 paths.

20 paths contain an answer-reaching path, 49.03% have their first answer-reaching path at ranks 2–20. This shows that useful evidence is often not ranked first and motivates retaining multiple ranked paths rather than relying only on the top-ranked path.

### E.2 Candidate-Path Error Decomposition

We further analyze the remaining errors on CWQ according to whether answer-reaching evidence is available in the candidate pool and the selected top-20 paths. Over the full test set, 19.09% of questions have no answer-reaching path in the candidate pool, while 17.36% contain such a path in the full pool but miss it in the top-20 ranking. Even when the top-20 contains an answer-reaching path, the final prediction remains incorrect for 5.51% of questions with Qwen2.5-7B-Instruct and 5.06% with Llama-3.1-8B-Instruct.

As shown in Table 13, top-20 ranking misses account for the largest fraction of incorrect predictions for both backbones, followed by failures of candidate-path coverage. Generation errors are comparatively less frequent when the selected top-20 paths already contain an answer-reaching path. These results identify candidate-path construction and ranking as the main remaining bottlenecks of PathSoft.

| Method              | Tuned | Params | Hit          | Tokens   | Sec/Q       | Calls |
|---------------------|-------|--------|--------------|----------|-------------|-------|
| GCR (GPT-4o-mini)   | Yes   | 8.03B  | 83.71        | 102.22   | 2.24        | 2     |
| ToG (GPT-4o-mini)   | No    | 0      | 67.94        | 1969.33  | 8.55        | 2.88  |
| FiDeLiS (GPT-3.5)   | No    | 0      | 73.05        | 35161.55 | 104.17      | 7.74  |
| SubgraphRAG (Qwen)  | No    | MLP    | 83.48        | 2271.45  | 3.75        | 1     |
| SubgraphRAG (Llama) | No    | MLP    | 85.57        | 2624.45  | 3.61        | 1     |
| Text-only (Qwen)    | No    | 1.07M  | 80.52        | 608.55   | 1.90        | 1     |
| Text-only (Llama)   | No    | 1.07M  | 83.16        | 590.80   | 2.12        | 1     |
| PathSoft (Qwen)     | No    | 63.46M | 86.17        | 624.55   | 1.47        | 1     |
| PathSoft (Llama)    | No    | 71.85M | <b>88.81</b> | 606.80   | <b>1.20</b> | 1     |

Table 15: Overall efficiency comparison on WebQSP. Tokens denotes the average number of input tokens, Sec/Q denotes inference time per question, and Calls denotes the average number of LLM calls per question.

### E.3 Utilization of Lower-Ranked Evidence

We further examine whether PathSoft can utilize useful evidence when the first answer-reaching path is not ranked first. Using the same PathMLP-selected paths for Text-only and PathSoft, we divide CWQ questions according to whether the first answer-reaching path appears at rank 1 or at ranks 2–20.

As shown in Table 14, PathSoft consistently outperforms Text-only in both rank conditions. The improvement remains substantial when the first answer-reaching path appears at ranks 2–20, indicating that PathSoft does not simply follow the highest-ranked path and can utilize useful evidence appearing below higher-ranked distractors.

Within the ranks 2–20 subset, PathSoft corrects substantially more Text-only errors than it introduces new errors. For Qwen2.5-7B, 32.45% of questions change from incorrect to correct, compared with 3.19% from correct to incorrect. For Llama-3.1-8B, the corresponding rates are 20.69% and 5.38%, respectively.

## F Additional Efficiency Results

Table 15 reports the overall efficiency comparison on WebQSP. PathSoft achieves the highest Hit on WebQSP while keeping the backbone LLM frozen and requiring only one LLM call per question. Compared with iterative prompting methods such as ToG and FiDeLiS, PathSoft uses far fewer input tokens and substantially lower inference latency. For example, PathSoft with Llama-3.1-8B-Instruct obtains 88.81% Hit with 606.80 input tokens and 1.20 Sec/Q, whereas FiDeLiS uses 35,161.55 tokens and 104.17 Sec/Q while reaching 73.05% Hit. Although GCR uses a shorter input, it requires tuning an 8.03B-parameter model and still underperforms PathSoft.

| Dataset       | CWQ                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------|
| Question      | Where did the actor nominated for the award for <i>The Catcher in the Rye</i> die?                       |
| Gold answer   | Cornish                                                                                                  |
| Backbone      | Llama-3.1-8B-Instruct                                                                                    |
| Text-only     | ans: Natural causes                                                                                      |
| PathSoft      | ans: Cornish                                                                                             |
| Random20      | ans: New York City                                                                                       |
| Rank 1, 0.389 | <i>The Catcher in the Rye</i> → author → J. D. Salinger → cause of death → Natural causes                |
| Rank 2, 0.056 | <i>The Catcher in the Rye</i> → author → J. D. Salinger → place of death → Cornish                       |
| Rank 3, 0.050 | <i>The Catcher in the Rye</i> → author → J. D. Salinger → works written → For Esme-with Love and Squalor |
| Rank 4, 0.047 | <i>The Catcher in the Rye</i> → author → J. D. Salinger → spouse → m.0n4tvj6                             |
| Rank 5, 0.039 | <i>The Catcher in the Rye</i> → author → J. D. Salinger → spouse → m.0n4tvls                             |

Table 16: A CWQ case where Text-only and PathSoft use the same top-20 ranked paths. The top-ranked path is a relation-mismatched distractor, while the second-ranked path reaches the gold answer.

Under the same LLM backbone, PathSoft is also more efficient than SubgraphRAG. With Qwen2.5-7B-Instruct, PathSoft improves Hit from 83.48% to 86.17%, while reducing the average input length from 2,271.45 to 624.55 tokens and Sec/Q from 3.75 to 1.47. With Llama-3.1-8B-Instruct, it improves Hit from 85.57% to 88.81%, while reducing the average input length from 2,624.45 to 606.80 tokens and Sec/Q from 3.61 to 1.20. Compared with the Text-only variant, PathSoft brings a 5.65% Hit improvement under both Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, with only a small increase in input length. These results are consistent with the CWQ efficiency analysis and show that soft prompt based path structure encoding improves answer accuracy without introducing substantial inference overhead.

## G Case Study

### G.1 Soft Prompt Helps Distinguish Relation-Mismatched Paths

Table 16 presents a CWQ example where PathSoft succeeds while the text-only variant fails under the same Llama-3.1-8B-Instruct backbone and the same top-20 ranked paths. For readability, we omit intermediate Freebase IDs and shorten relation names.

The top-ranked path is topically relevant because it reaches the correct person, J. D. Salinger, but it follows the wrong relation, *cause of death*. The

|               |                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------|
| Dataset       | CWQ                                                                                                       |
| Question      | Who was the first President of the United States during WW2?                                              |
| Gold answer   | Franklin D. Roosevelt                                                                                     |
| Backbone      | Llama-3.1-8B-Instruct                                                                                     |
| Random20      | George Washington, John Adams, ...                                                                        |
| Text-only     | George Washington                                                                                         |
| PathSoft      | Franklin D. Roosevelt                                                                                     |
| Rank 1, 0.560 | World War II → speeches on this topic → speaker → Franklin D. Roosevelt                                   |
| Rank 2, 0.025 | World War II → speeches on this topic → speaker → Franklin D. Roosevelt → government position → m.03q2h_9 |
| Rank 3, 0.018 | President of the United States → office holders → Barack Obama                                            |
| Rank 4, 0.003 | President of the United States → office holders → William Henry Harrison → vice president → John Tyler    |
| Rank 5, 0.003 | President of the United States → image → George Washington, first President of the United States          |

Table 17: A CWQ case illustrating the benefit of path ranking. PathMLP places the Roosevelt path at rank 1, while random or text-only evidence leads the model toward early U.S. presidents.

|               |                                                                                                        |
|---------------|--------------------------------------------------------------------------------------------------------|
| Dataset       | CWQ                                                                                                    |
| Question      | What kind of paintings, that influenced Edvard Munch, did that Frida Kahlo do?                         |
| Gold answer   | Surrealism                                                                                             |
| Backbone      | Llama-3.1-8B-Instruct                                                                                  |
| Text-only     | Symbolic Landscape, Paisaje de Tehuantepec, ...                                                        |
| PathSoft      | Surrealism                                                                                             |
| Random20      | Frida Kahlo painting titles                                                                            |
| Rank 1, 0.432 | Frida Kahlo → influenced by → Diego Rivera                                                             |
| Rank 2, 0.389 | Edvard Munch → influenced → Surrealism                                                                 |
| Rank 3, 0.305 | Frida Kahlo → associated periods or movements → Surrealism                                             |
| Rank 4, 0.239 | Edvard Munch → influenced by → Henri de Toulouse-Lautrec                                               |
| Rank 5, 0.195 | Frida Kahlo → influenced by → Diego Rivera → artworks → Symbolic Landscape                             |
| Rank 6, 0.147 | Frida Kahlo → influenced by → Diego Rivera → films → m.0k7s2g3                                         |
| Rank 7, 0.076 | Frida Kahlo → influenced by → Diego Rivera → artworks → Paisaje de Tehuantepec (Tehuantepec Landscape) |

Table 18: A compositional CWQ case. Text-only focuses on concrete artwork titles, while PathSoft predicts the shared abstract movement supported by the ranked paths.

text-only variant copies this distractor endpoint and predicts “Natural causes”. In contrast, PathSoft predicts “Cornish”, which is supported by the second-ranked *place of death* path. Since both variants use the same top-20 textual paths, this case suggests

|                          |                                                                                                                    |
|--------------------------|--------------------------------------------------------------------------------------------------------------------|
| Dataset                  | CWQ                                                                                                                |
| Question                 | Which geographical feature is near Whistler Mountain?                                                              |
| Gold answer              | North American Cordillera                                                                                          |
| Positive path in top-20  | None                                                                                                               |
| Positive path in top-100 | None                                                                                                               |
| Text-only                | Whistler Blackcomb, Snow Dome, Mount Columbia, ...                                                                 |
| PathSoft                 | Columbia River, Saint Elias Mountains, Rocky Mountains, ...                                                        |
| Random20                 | Saint Elias Mountains, Rocky Mountains, ...                                                                        |
| Rank 1, 0.354            | Whistler Mountain → contained by → British Columbia → works → Travels with Samantha                                |
| Rank 2, 0.255            | Whistler Mountain → contained by → British Columbia                                                                |
| Rank 3, 0.215            | Whistler Mountain → contained by → Canada                                                                          |
| Rank 4, 0.098            | Whistler Mountain → contains → Whistler Blackcomb                                                                  |
| Rank 5, 0.082            | Whistler Mountain → contained by → British Columbia → tourist attractions → Rocky Mountains → works → Monkey Beach |
| Rank 6, 0.078            | Whistler Mountain → contained by → British Columbia → partially contains → Snow Dome                               |
| Rank 7, 0.072            | Whistler Mountain → contained by → British Columbia → works → Notes from the century before                        |
| Rank 8, 0.049            | Whistler Mountain → contained by → British Columbia → tourist attractions → Rocky Mountains                        |
| Rank 9, 0.038            | Whistler Mountain → contained by → British Columbia → capital → Victoria                                           |
| Rank 10, 0.030           | Whistler Mountain → contained by → British Columbia → partially contains → Mount Columbia                          |

Table 19: A failure case caused by missing answer-reaching paths. The gold answer does not appear in the saved top-100 candidate paths, so PathSoft cannot recover the correct evidence.

that soft prompts help the frozen LLM use relation information and ordered path structure more effectively.

## G.2 Path Ranking Provides Useful Evidence

Table 17 presents a CWQ example where ranked PathSoft succeeds while the random-path variant fails. This case illustrates how PathMLP helps prioritize evidence that is relevant to the temporal constraint in the question.

The question is ambiguous because it contains both “first President of the United States” and the temporal constraint “during WW2”. Without reliable ranked evidence, the model is distracted by early-president entities such as George Washing-

ton and John Adams. PathMLP instead assigns the highest score to a World War II path that reaches Franklin D. Roosevelt. With this ranked evidence, PathSoft predicts the correct answer. This case suggests that path ranking is important for surfacing question-relevant evidence before soft prompt generation.

### **G.3 PathSoft Helps Compositional Reasoning**

Table 18 presents a CWQ example involving compositional reasoning. The question requires identifying an abstract art movement rather than simply selecting a nearby concrete entity from one path.

The text-only variant is distracted by concrete artwork titles appearing in Frida-related paths, such as “Symbolic Landscape” and “Paisaje de Tehuantepec”. However, the question asks for the kind of paintings, which corresponds to an abstract movement. PathSoft predicts “Surrealism”, supported by paths related to both Edvard Munch and Frida Kahlo. This case illustrates how soft prompts can help the frozen LLM aggregate information across multiple ranked paths and avoid simply copying a nearby entity from the textual evidence.

### **G.4 Failure Case: Missing Answer-Reaching Paths**

Table 19 presents a failure case where PathSoft cannot recover the gold answer because no answer-reaching path appears in the saved candidate paths. This example illustrates the dependence of PathSoft on candidate path coverage.

In this example, the gold answer “North American Cordillera” is absent from both the top-20 ranked paths and the saved top-100 candidate paths. As a result, all variants are forced to rely on incomplete or indirect evidence about Whistler Mountain, British Columbia, and nearby geographical entities. PathSoft still produces plausible geographic answers such as “Rocky Mountains”, but it cannot recover the missing gold evidence. This failure case supports our limitation that PathSoft depends on the quality and coverage of the candidate path pool.