
KG-TRACES: Enhancing Large Language Models with Knowledge Graph-constrained Trajectory Reasoning and Attribution Supervision

Rong Wu^{1,2}, Pinlong Cai², Jianbiao Mei^{1,2}, Licheng Wen², Tao Hu^{2,3},

Xuemeng Yang², Daocheng Fu^{2,4}, Botian Shi^{2,*}

¹ Zhejiang University ² Shanghai Artificial Intelligence Laboratory

³ University of Science and Technology of China ⁴ Fudan University

Abstract

Large language models (LLMs) have made remarkable strides in various natural language processing tasks, but their performance on complex reasoning problems remains hindered by a lack of explainability and trustworthiness. This issue, often manifesting as hallucinations or unattributable reasoning processes, limits their applicability in complex reasoning scenarios. To address this, we propose **Knowledge Graph-constrained Trajectory Reasoning Attribution and Chain Explanation Supervision (KG-TRACES)**, a novel framework that enhances the reasoning ability of LLMs through explicit supervision over reasoning paths and processes. KG-TRACES jointly supervises the model to: (1) predict symbolic relation paths, (2) predict full triple-level reasoning paths, and (3) generate attribution-aware reasoning processes grounded in the reasoning paths. At inference phase, the model adapts to both KG-available and KG-unavailable scenarios, retrieving reasoning paths from a KG when possible or predicting plausible reasoning paths with only intrinsic knowledge when not. This design enables the model to reason in an explainable and source-attributable pattern. Through extensive experiments on complex reasoning tasks, we demonstrate that KG-TRACES significantly outperforms existing SOTA: it improves Hits@1 by 1.6% and F1 by 4.7% on WebQSP, and achieves improvements of 4.8% in Hits@1 and 2.1% in F1 on CWQ. Moreover, we show its transferability to specialized domains such as medicine. By visualizing the intermediate steps of reasoning processes, we further show that the explicit supervision introduced by KG-TRACES leads to more stable and goal-directed reasoning processes, aligning closely with correct answers. Code is available at <https://github.com/Edaizi/KG-TRACES>.

1 Introduction

Large language models (LLMs) have achieved remarkable success across a wide range of natural language processing tasks. Yet, their performance on complex multi-hop reasoning remains hindered by a lack of explainability and attribution [6, 14]. In particular, current models often generate hallucinated intermediate steps or ungrounded conclusions, which severely limits their applicability in domains that demand explainable and faithful reasoning, such as open-domain question answering, scientific discovery or clinical decision support [7, 21].

*corresponding author

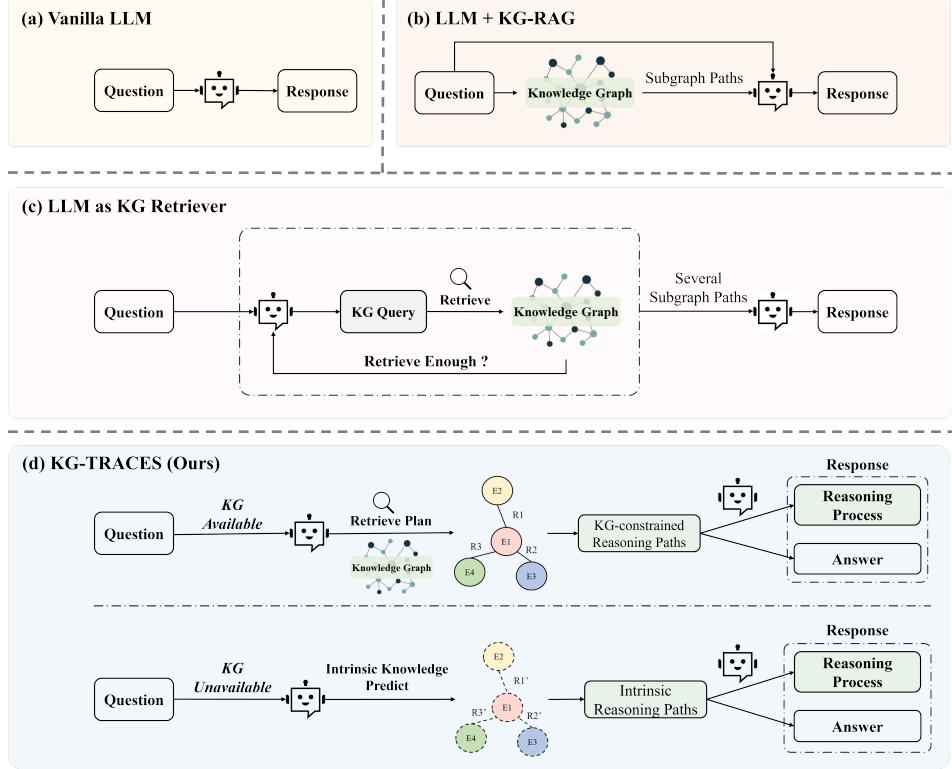


Figure 1: Comparison of representative reasoning methods in LLMs-based frameworks: (a) **Vanilla LLMs**, where the model generates responses directly from the question; (b) **LLMs + KG-RAG**, which uses KG to retrieve relevant subgraph paths to aid the reasoning; (c) **LLMs as KG-Retriever**: where LLMs is a active retriever, querying KG for relevant information, determining whether sufficient knowledge has be retrieved; (d) **KG-TRACES (Ours)**, which can generate faithful and attributable response based on symbolic subgraph reasoning paths under different KG access conditions.

Recent advances have explored several directions to mitigate these issues. Chain-of-Thought (CoT) prompting encourages step-by-step reasoning by examples [10, 13, 31], while retrieval-augmented generation (RAG) methods attempt to ground model outputs with external knowledge [12, 9, 35]. Knowledge graph (KG) capture abundant factual world knowledge, organize the structural relationships between entities from fragmented information and store knowledge in the form of triples. Researchers have developed KG-enhanced LLMs approaches (which augmented LLMs with structured factual knowledge from KG to reasoning) attempting to mitigate LLM’s hallucination and lack of faithful knowledge [32, 8, 34, 18]. Nevertheless, these methods either only rely on loose prompt-based constraints with external KG information [34, 27], or depend heavily on external knowledge retrieval, suffering significant performance degradation under limited KG access [39, 17]. Moreover, while they incorporate structured and factual KG information, they fail to provide attributable reasoning processes or explain the sources of their conclusions. This lack of reasoning attribution fundamentally undermines the goal of improving trustworthiness, as it fails to align with the very premise of using structured world knowledge to enhance reasoning explainable and attributable reasoning.

In this work, we propose **Knowledge Graph-constrained Trajectory Reasoning Attribution and Chain Explanation Supervision (KG-TRACES)**, a novel framework for supervised reasoning that enables LLMs to generate explainable, attribution-aware reasoning processes under KG access and KG limited scenarios. KG-TRACES is built on the hypothesis that complex reasoning should follow structured and explainable patterns, and LLMs can be trained to internalize such patterns through fine-grained supervision. Therefore, we incorporate reasoning paths derived from KG, which store factual knowledge in the form of triples $\langle \text{subject}, \text{relation}, \text{object} \rangle$. A reasoning path consists of a sequence of linked triples that connect a question to its answer through intermediate entities and relations.

Compared to entity-level facts, relation sequences offer greater abstraction and stability, making them more robust to changes in entity coverage. KG-TRACES is built on three key design principles: (1) supervising LLMs to predict symbolic relation paths and triple-level symbolic reasoning paths that connect questions to answers; (2) training the model to generate step-by-step reasoning processes that are explicitly attributed to either symbolic reasoning paths or just inference based on intrinsic knowledge; and (3) keep faithful reasoning under varying knowledge access conditions—including the presence or absence of KG.

We validate KG-TRACES through extensive experiments on complex reasoning benchmarks, demonstrating significant improvements over previous methods. Furthermore, we evaluate the transferability of our approach by adapting it to specialized domains (medicine) question answering task, where robust and explainable reasoning is critical. To gain deeper insights into the reasoning behaviors enabled by KG-TRACES, we conduct reasoning process visualization analysis [47], which reveals that KG-TRACES can exploring large latent reasoning space and converging to correct answers regions accurately.

The contributions of this paper can be summarized as follows:

- KG-TRACES is proposed as a unified supervision framework that enables LLMs to perform attribution-aware symbolic reasoning with or without access to external KG.
- The framework incorporates a multi-task fine-tuning strategy over constructed structured relation paths, triple paths, and attributable reasoning processes datasets, fostering explainable and attributable reasoning behavior.
- Extensive experiments on both general-domain and medical-domain benchmarks validate the effectiveness of KG-TRACES, demonstrating notable reasoning performance improvements due to our supervision design.

2 Related Works

2.1 LLM Complex Reasoning with Prompt

Large language models (LLMs) have demonstrated strong emergent abilities in reasoning, which has led to a surge of interest in prompting-based methods that aim to elicit more clear reasoning processes. CoT prompting encourages step-by-step problem solving by introducing intermediate reasoning steps through examples [13]. Building on this idea, Tree-of-Thought (ToT) methods introduce branching and self-evaluation mechanisms, enabling models to explore multiple candidate paths and choose among them [37]. ReAct interleaves reasoning and acting, allowing the model to decide when to retrieve, reflect, or infer [38]. Self-consistency further enhances answer robustness by aggregating multiple sampled reasoning chains [33].

Although effective, these methods operate entirely through inference-time prompting. The intermediate reasoning steps are unconstrained, lack ground truth alignment, and provide no attribution to external or internal sources. Several works enhance prompting with plan-and-verify strategies [19, 46, 43], but still leave the model’s reasoning process unsupervised during training. As a result, hallucinations and inconsistencies persist. In contrast, our work introduces training-time supervision over both symbolic paths and natural language reasoning process sequences, enabling models to generate interpretable and attribution-aware outputs.

2.2 Knowledge Graph Enhanced LLMs Reasoning

knowledge graph (KG) offer a structured and factual foundation for augmenting the reasoning capabilities of large language models (LLMs). Approaches utilize KG by retrieving and injecting triples directly into model inputs [44, 16], or by translating questions into KG queries for entity linking and context construction [11, 40]. These methods are generally loosely coupled, treating KG as external information sources. More recent work has explored tighter KG-LLM integration. Think-on-Graph (ToG) [27] introduces an iterative mechanism in which models traverse KG paths step by step via beam search , while Think-on-Graph 2.0 extends this paradigm by combining KG traversal with context retrieval in a hybrid reasoning loop [18]. Reasoning on Graphs (RoG) [17] introduces a stronger coupling between path planning and reasoning, using retrieved KG paths to supervise model outputs via posterior distillation. Other efforts, such as KELP, focus on scoring and selecting semantically relevant paths to guide model generation [16].

However, these approaches generally assume availability of high-coverage KG, and most do not support reasoning under varying access conditions. Moreover, while RoG incorporates path supervision, it does not generate full reasoning processes, nor does it explicitly model provenance or attribution. Our framework, KG-TRACES, differs in three key ways: (1) it trains LLMs to generate both symbolic reasoning paths and natural language processes with step-level attribution; (2) it enables more robust generalization under KG-present or KG-absent scenarios; and (3) it applies a unified generation scheme regardless of the source of retrieved or predicted paths. These enable improved robustness, interpretability, and domain transferability.

3 KG-TRACES: Structured Reasoning with KG-Guided Supervision

KG-TRACES is a structured reasoning framework designed to enhance the explainable and attributable abilities of LLMs via explicit supervision over symbolic reasoning paths and natural language reasoning processes. It operates under a unified generation paradigm applicable in both KG-accessible and KG-absent scenarios.

3.1 Overview of KG-TRACES Framework

As Figure 2 shows, we supervise LLMs with two type reasoning targets during training: (1) symbolic paths (relation level paths over KG and triple level paths aligned with KG facts), and (2) natural language reasoning processes annotated with step-level attribution provenance. During inference, KG-TRACES can retrieve supporting paths from an external KG when available, or rely on intrinsic knowledge predicted reasoning paths when external KG is unavailable. In both cases, KG-TRACES generates a structured reasoning process and attributes intermediate steps to its knowledge source.

3.2 Symbolic Reasoning Paths Prediction Supervision

To enable structured and faithful multi-step reasoning, KG-TRACES supervises the model to learn retrieve planning over KG and full reasoning paths which is helpful to answer the question. We frame the retrieve planning as the prediction of symbolic relation paths which link the intermediate relations of reasoning paths from question to its answers.

Notation. Let q denote an input question and \mathcal{G} the underlying knowledge graph, composed of factual triples (e, r, e') where e and e' are entities, and $r \in \mathcal{R}$ is a relation. A symbolic relation path² is defined as $r = (r_1, \dots, r_k)$ and a symbolic triple path (reasoning path)³ as $p = ((e_1, r_1, e_2), (e_2, r_2, e_3), \dots, (e_k, r_k, e_{k+1}))$, both representing multi-hop reasoning trajectories. For each q , we denote $\mathcal{R}(q)$ and $\mathcal{T}(q)$ as the retrieved relation and triple path sets, respectively.

Given a dataset $\mathcal{D} = \{(q, \mathcal{R}(q), \mathcal{T}(q))\}$, we train the model to approximate posterior distributions over r and t by minimizing the KL divergence between predicted distributions $P_\theta(\cdot | q)$ and empirical distributions $Q(\cdot)$ constructed from retrieval. The reasoning paths prediction supervision objective consists of two components:

1) Relation Path Distribution Learning. The model learns to approximate the posterior distribution $Q(r)$ over retrieved relation paths via KL minimization:

$$\mathcal{L}_{\text{relation}} = \mathbb{E}_{r \sim Q(r)} [\log Q(r) - \log P_\theta(r | q)] = D_{\text{KL}}(Q(r) \| P_\theta(r | q)) \quad (1)$$

2) Triple Path Distribution Learning. The model similarly matches the posterior distribution $Q(p)$ over triple paths p from retrieval:

$$\mathcal{L}_{\text{triple}} = D_{\text{KL}}(Q(p) \| P_\theta(p | q)) \quad (2)$$

We encourage the model to internalize plausible multi-hop reasoning paths that are discoverable via KG traversal. These symbolic paths serve as structured latent reasoning plan that guide the model toward explainable multi-hop reasoning. Rather than enforcing them as hard-labeled ground truth

²A symbolic relation path example: teams → mascot in Figure 2.

³A symbolic triple path (reasoning path) example: (American League West, teams, Seattle Mariners) → (Seattle Mariners, mascot, Mariner Moose) in Figure 2.

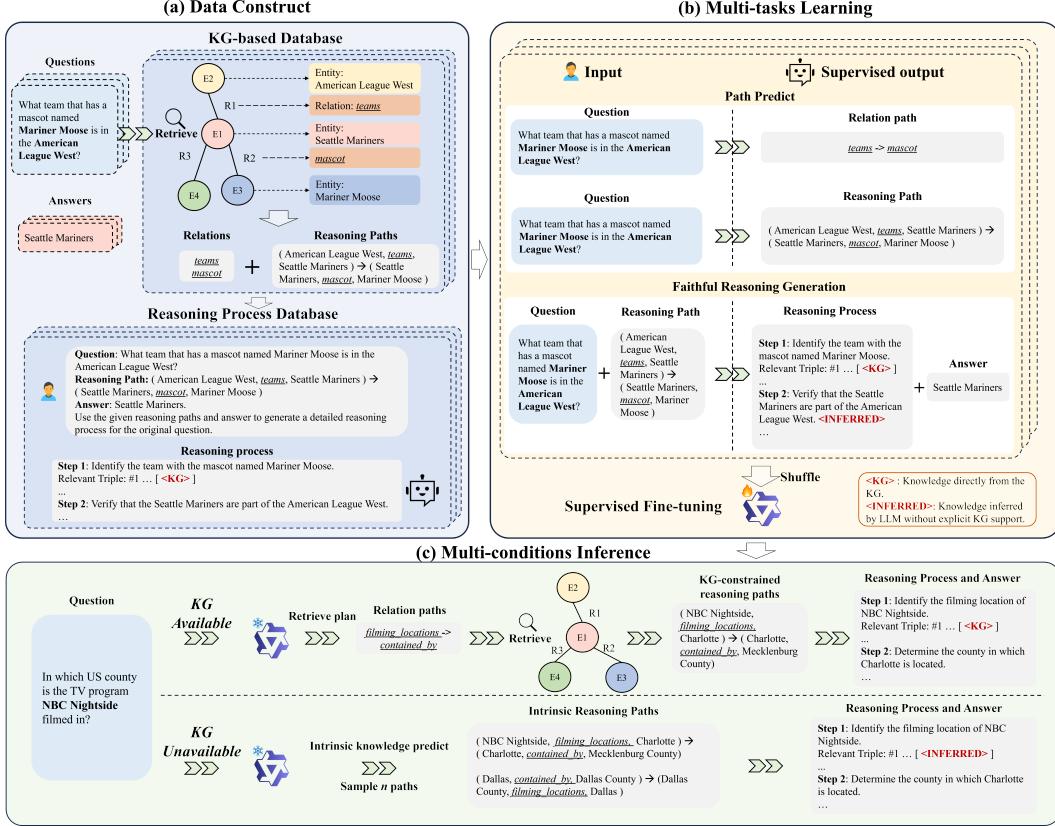


Figure 2: Overview of the KG-TRACES framework. The framework consists of three key components: **(a) Data Construction:** KG-TRACES integrates original datasets with QA data and symbolic reasoning paths from KG to build the KG-based database and reasoning process database for multi-task learning; **(b) Multi-task Learning:** Model is trained for two types of tasks: path prediction (including relation paths prediction used for KG retrieval, and full reasoning paths prediction) and faithful reasoning process generation (supervised model produces attributable, interpretable reasoning processes based on symbolic reasoning paths); **(c) Multi-conditions Inference:** The inference process varies depending on KG availability. With KG access, KG-TRACES predicts relation paths and retrieves whole reasoning paths from KG, generating faithful reasoning process and answer. Without KG access, KG-TRACES predicts reasoning paths relying on intrinsic knowledge to support faithful reasoning process and answer.

(which may be incomplete or noisy), we treat these paths as soft supervision signals—posterior samples that regularize the model’s prediction behavior. This approach enables KG-TRACES to align its internal reasoning with plausible symbolic paths without being overly constrained by KG completeness, thus remaining robust with partial or sparse KG coverage. When inference, we use beam search to sample several relation paths or triple paths as the paths distribution of model-inferred.

3.3 Attribution-Aware Reasoning Supervision

While symbolic reasoning paths reflect model’s abstract structured latent plan, practical reasoning requires transforming these plans into concrete, explainable natural language justifications. To this end, KG-TRACES supervises the generation of full reasoning processes annotated with attribution that reflect attribution provenance and causal relevance.

Notation. Let $y = (y_1, \dots, y_T)$ be a reasoning process sequence. The model generates y conditioned on both the question q and symbolic reasoning paths p . y includes attribution labels such as: <KG> and <INFERRED> denoting whether a reasoning step is grounded in retrieved KG knowledge or inferred by

model’s intrinsic knowledge; <EFFECTIVE> and <INEFFECTIVE> denoting whether a reasoning step is essential for deriving the final answer. These labels help model learn to attribute causal relevance to different parts of the reasoning chain⁴. To construct the reasoning process augmented dataset, we employ a general LLM (Qwen-72B-Instruct) to generate explicit step-by-step reasoning processes for each question in the dataset, conditioned on the question, the gold answer, and potential useful reasoning paths. During generation, we instruct the LLM to attribute the basis of each reasoning step using attribution labels (e.g., <KG>, <INFERRED>), enabling fine-grained attribution labels for subsequent training. Detail prompt is in the appendix B.1.

Objective. Conditioned on the question q and a reasoning path p , the model is supervised to generate a structured natural language reasoning process y as formula (3).

$$\mathcal{L}_{\text{process}} = - \sum_{j=1}^T \log P_\theta(y_j | q, p, y_1, \dots, y_{j-1}) \quad (3)$$

This enables the model to generate reasoning explanations aligned with symbolic reasoning paths while explicitly learning to attribute evidence source and causal effectiveness.

Unified Training Objective. The final objective is the sum of all three components:

$$\mathcal{L}_{\text{KG-TRACES}} = \mathcal{L}_{\text{relation}} + \mathcal{L}_{\text{triple}} + \mathcal{L}_{\text{process}} \quad (4)$$

Although all supervision is explicit, the reasoning paths can be viewed as approximations to latent reasoning trajectories that the model would ideally recover on its own. Rather than treating these structures as fixed ground-truths, we treat them as soft supervision to guide the model’s internal reasoning alignment. Special formally, the likelihood of generating a valid reasoning process can be bounded by conditioning on reasoning paths is:

$$\log P_\theta(y | q) \gtrsim \mathbb{E}_{p \sim Q(\cdot)} [\log P_\theta(y_j | q, p, y_1, \dots, y_{j-1})] \quad (5)$$

This training view motivates our use of reasoning paths as soft supervision rather than fixed constraints. Such a soft supervision encourages the model to internalize generalizable reasoning patterns, making it capable of predicting coherent reasoning paths for previously unseen questions. By aligning the model’s internal reasoning with plausible symbolic paths, we approximate latent inference behavior without relying on variational modeling. Unlike prior work that focuses solely on symbolic retrieval planning, we jointly supervise both structured path prediction and attribution-aware reasoning process generation, enabling KG-TRACES to learn faithful and attributable reasoning under a unified objective.

4 Experiments

4.1 Experimental Setups

Tasks and Datasets. We conduct comprehensive experiments to evaluate the effectiveness and generalization ability of KG-TRACES across both general-domain and domain-specialized (medicine) reasoning tasks. We evaluate the general reasoning ability of KG-TRACES on two open-domain multi-hop KGQA benchmarks: WebQuestionsSP (WebQSP) [41] and Complex WebQuestions (CWQ) [29]. To evaluate cross-domain generalization, we additionally use GenMedGPT-5k [15], a medical QA benchmark constructed from ChatGPT-patient interactions and supported by a curated medical KG of symptoms, diseases, drugs, and treatments [34]. The details of the datasets are provided in Appendix A.

⁴Intuitively, not all retrieved or generated paths are useful for answering the question. By marking whether each reasoning step is causally dependent on a specific path, the model is encouraged to distinguish between distractive information and core inferential paths. This token-level attribution allows finer-grained supervision of reasoning quality and helps promote verifiable and goal-directed explanation generation.

Implementations. We use Qwen2.5-Chat-7B [36] as the LLM backbone, which is SFT with multi-tasks on the WebQSP and CWQ for 3 epochs. During inference stage, we sample the top-3 relation paths and top-3 triple paths using beam search for each question. The detailed settings are described in Appendix C.

Baselines. In the general domain KGQA benchmarks, we compare KG-TRACES with 25 baselines grouping into 6 categories: 1) Embedding-based methods, 2) Retrieval-augmented methods, 3) Semantic parsing methods, 4) Vanilla LLMs, 5) Prompt Augment LLMs, and 6) LLMs+KG methods. In the medical-domain benchmarks, we compare KG-TRACES with baselines with 3 categories: 1) Retrieval-augmented methods, 2) Vanilla LLMs, 3) LLMs+KG. The details of each baseline are described in Appendix C.1.

Evaluation Metrics. In the general-domain KGQA benchmarks, we adopt Hits@1 and F1 as our primary evaluation metrics to measure model’s performance. In the medical-domain benchmarks, since GenMedGPT-5k consists of generated dialogue between patients and GPT-3.5, conventional string-matching metrics are inadequate. Instead, we adopt LLM-based scoring to assess response quality. The details of medical evaluation metrics are described in Appendix D.1.

4.2 Performance on General Reasoning Tasks

As summarized in Table 1, our model achieves state-of-the-art performance across both datasets. Compared to the SOTA method RoG [17], which also integrates KG-based planning, KG-TRACES improves Hits@1 by 1.6% and F1 by 4.7% on WebQSP, and achieves 4.8% and 2.1% gains respectively on CWQ. These gains are more notable on CWQ, which contains a larger proportion of complex 3-hop questions. This highlights the value of explicit reasoning process supervision in guiding multi-step reasoning task. Explainable explicit reasoning process supervision may lead to better complex reasoning performance. And we illustrate a case in Figure 3 to show explainable and attributable reasoning ability of KG-TRACES.

KG-TRACES also consistently surpasses retrieval-based and prompting-based LLM baselines, underscoring the value of explicit supervision over latent reasoning trajectories. In contrast to prior methods that rely solely on prompt heuristics or sparse external knowledge, KG-TRACES enables LLMs to learn explainable and attributable reasoning process aligned with symbolic structure knowledge. These improvements generalize across both simple (WebQSP) and complex (CWQ) reasoning questions, confirming the generalization of our approach.

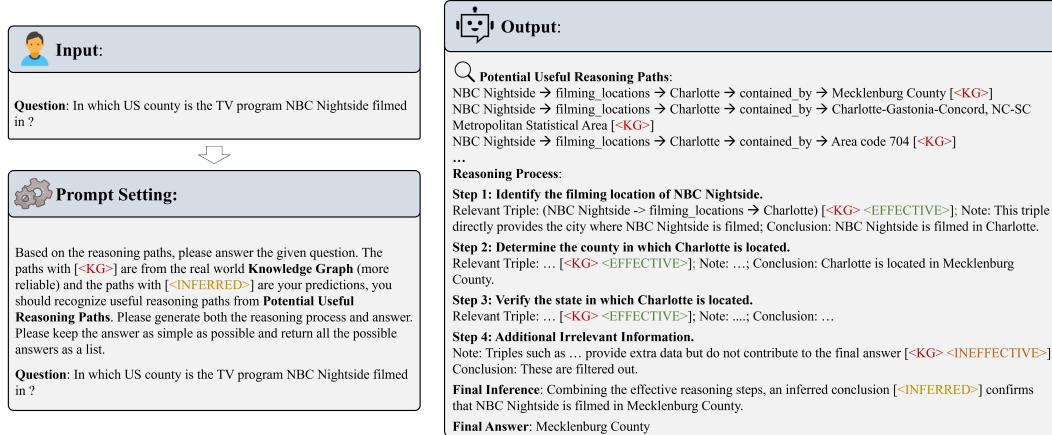


Figure 3: Example of attributable and explainable reasoning of KG-TRACES.

4.3 Analysis Under Limited KG Access

We evaluate four settings to analyze KG-TRACES under varying KG access conditions: (1) no reasoning path usage (*No KG-aug*), (2) using predicted relation paths for KG retrieval (*KG-aug (rel)*),

Table 1: Performance comparison on WebQSP and CWQ. Best results in each column are in bold.

Type	Methods	WebQSP			CWQ		
		Hits@1 ↑	F1 ↑		Hits@1 ↑	F1 ↑	
Embedding	KV-Mem [20]	46.7	34.5		18.4	15.7	
	EmbedKGQA [23]	66.6	-		45.9	-	
	NSM [5]	68.7	62.8		47.6	42.4	
	TransferNet [24]	71.4	-		48.6	-	
	KGTraces [22]	56.1	-		36.5	-	
Retrieval	GraftNet [26]	66.4	60.4		38.8	32.7	
	PullNet [25]	68.1	-		45.9	-	
	SR+NSM [45]	68.9	64.1		50.2	47.1	
	SR+NSM+E2E [45]	69.5	64.1		49.3	46.3	
Semantic Parsing	SPARQL [28]	-	-		31.6	-	
	QGG [11]	73.0	73.8		36.9	37.4	
	ArcaneQA [4]	-	75.3		-	-	
	RnG-KBQA [40]	-	76.2		-	-	
Vanilla LLMs	Flan-T5-xl [2]	31.0	-		14.7	-	
	Alpaca-7B [30]	51.8	-		27.4	-	
	LLaMA3.1-Chat-8B [3]	63.4	24.7		36.9	14.2	
	Qwen2.5-Chat-7B [36]	45.7	29.3		20.2	16.1	
	ChatGPT [17]	66.8	-		39.9	-	
Prompt Augmented LLMs	LLaMA3.1-Chat-8B + COT	64.6	22.9		40.6	12.3	
	Qwen2.5-Chat-7B + COT	49.1	26.6		32.1	8.6	
	ChatGPT + CoT [17]	75.6	-		48.9	-	
LLMs + KG	KD-CoT [32]	68.6	52.5		55.7	-	
	UniKGQA [8]	77.2	72.2		51.2	49.1	
	DECAP [42]	82.1	78.8		-	-	
	RoG [17]	85.7	70.8		62.6	56.2	
	KG-Traces (ours)	87.1	74.1		65.6	57.4	

(3) predicting triple paths without KG (*No KG-aug (triple)*), and (4) combining predicted relations and predicted triple paths (*KG-aug (rel + triple)*).

Results in Table 2 show that *KG-aug (rel)* yields the best performance, slightly outperforming *KG-aug (rel + triple)*, highlighting the effectiveness of high-quality reasoning paths. Even without KG access, *KG-aug (triple)* still significantly outperforms LLM-only baselines, demonstrating the model’s ability to generate meaningful reasoning paths through intrinsic knowledge. We observe varying trends between datasets: on WebQSP, *KG-aug (triple)* shows limited improvement over *No KG-aug*, while CWQ benefits significantly, with relative gains of 5.1% in Hits@1 and 4.4% in F1. These results suggest that even imperfect reasoning paths can provide valuable scaffolding in more complex multi-hop reasoning tasks, especially when questions require navigating longer reasoning chains. This supports the view that structured intermediate representations are especially helpful when navigating long or compositional reasoning chains. Overall, these findings demonstrate the strength of KG-Traces’s design: by equipping the model to leverage symbolic reasoning paths when KG is available—and fallback gracefully when not—it ensures robustness across different reasoning scenarios.

Table 2: Performance of KG-Traces under varying KG access conditions on WebQSP and CWQ.

Method	KG Usage	WebQSP				CWQ					
		Rel	Triple	Hits@1 ↑	F1 ↑	Precision ↑	Recall ↑	Hits@1 ↑	F1 ↑	Precision ↑	Recall ↑
No KG-aug				76.6	63.6	66.3	65.4	56.5	50.4	51.0	52.6
No KG-aug (triple)		✓		76.8	61.8	65.6	64.7	59.4	52.6	55.4	55.2
KG-aug (rel)	✓			87.1	74.1	74.8	79.5	65.6	57.4	57.7	61.7
KG-aug (rel + triple)	✓	✓		86.2	72.1	72.5	78.2	64.9	57.0	57.3	60.8

4.4 Cross-Domain Generalization Analysis

To evaluate the transferability of KG-Traces to specialized domains, we conduct experiments on dataset *GenMedGPT-5k* without any training. The evaluation details are on the Appendix D.1 and Appendix C.2.

Table 3: Model performance comparison on medical reasoning task (GenMedGPT-5k). Each score is reported as the mean \pm standard deviation across multiple runs using LLM-based evaluation.

Type	Method	Relevance \uparrow	Accuracy \uparrow	Completeness \uparrow	Clarity \uparrow	Conciseness \uparrow	Average \uparrow
Retrieval	BM25 Retriever	0.70 \pm 0.23	0.67 \pm 0.21	0.56 \pm 0.19	0.85 \pm 0.10	0.75 \pm 0.10	0.70 \pm 0.16
	Embedding Retriever	0.74 \pm 0.22	0.70 \pm 0.18	0.59 \pm 0.18	0.87 \pm 0.09	0.76 \pm 0.09	0.73 \pm 0.15
	KG Retriever	0.72 \pm 0.24	0.69 \pm 0.20	0.57 \pm 0.20	0.87 \pm 0.09	0.77 \pm 0.09	0.72 \pm 0.16
Vanilla LLMs	ChatGPT	0.80 \pm 0.14	0.76 \pm 0.10	0.62 \pm 0.12	0.91 \pm 0.05	0.80 \pm 0.05	0.78 \pm 0.09
	GPT-4	0.77 \pm 0.21	0.73 \pm 0.18	0.59 \pm 0.19	0.90 \pm 0.06	0.81 \pm 0.06	0.76 \pm 0.13
LLMs + KG	MindMap	0.77 \pm 0.21	0.72 \pm 0.18	0.61 \pm 0.18	0.89 \pm 0.08	0.80 \pm 0.07	0.76 \pm 0.14
KG-TRACES Variants	No KG-aug	0.83 \pm 0.20	0.77 \pm 0.20	0.68 \pm 0.16	0.92 \pm 0.08	0.79 \pm 0.07	0.80 \pm 0.13
	No KG-aug (triple)	0.76 \pm 0.22	0.71 \pm 0.22	0.63 \pm 0.19	0.87 \pm 0.11	0.73 \pm 0.10	0.74 \pm 0.16
	KG-aug (entity)	0.77 \pm 0.22	0.70 \pm 0.21	0.63 \pm 0.17	0.88 \pm 0.10	0.74 \pm 0.09	0.75 \pm 0.15
	KG-aug (entity + triple)	0.77 \pm 0.21	0.70 \pm 0.21	0.63 \pm 0.17	0.87 \pm 0.10	0.73 \pm 0.11	0.74 \pm 0.15

KG-TRACES Variants. As the model has never observed triples from medical domain, relation path prediction is infeasible. Instead, we only generate full triple paths. Specifically, We add two variants of KG-TRACES: 1) *KG-aug (entity)* (retrieve reasoning paths from question entities), and 2) *KG-aug (entity + triple)* (combine predicted reasoning paths and retrieved reasoning paths).

As shown in Table 3, KG-TRACES (*No KG-aug*) achieves the best average score (0.7970), surpassing strong baselines such as GPT-4 (0.7596) and MindMap (0.7582) by 4.9% and 5.1% respectively. This result underscores KG-TRACES’s complex reasoning ability remain effective even in unfamiliar medical domain without any external KG.

Interestingly, we observe that introducing explicit reasoning paths—either predicted paths or retrieved paths both degrades model’s performance. We attribute this to domain mismatch and path quality. For the *KG-aug (triple)* variant, KG-TRACES generates paths purely from its internal knowledge, having been trained on general domain KG. These generated paths may contain relation types or compositional patterns that are poorly aligned with medical domain, thereby introducing distributional noise during generation. This mismatch can disrupt the model’s response planning, especially in a zero-shot transfer setting. The resulting 7.5% average performance drop compared to the *No KG-aug* variant highlights the cost of injecting potentially misleading information. For the *KG-aug (entity)* variant, paths are retrieved based on question entities, such retrieval may suffer from two key limitations: 1) incomplete coverage of the medical KG leads to partial paths, and 2) even when paths exist, they may not lead to the answer. These issues may mislead the model’s reasoning.

4.5 Reasoning Process Quality and Visualization

We analyze the intermediate reasoning steps of KG-TRACES to understand whether multi-tasks supervision fosters structured and convergent reasoning behaviors. Following the prior work [47], we segment the reasoning process into stages and visualize the model’s evolving reasoning states, along with three metrics—*consistency*, *uncertainty*, and *perplexity*—to quantitatively track reasoning progression⁵. Metric definitions and visualization are detailed in Appendix D.2.

Case-level progression analysis. Figure 4 illustrates the trajectory of a representative QA instance over five reasoning stages. We observe that early thoughts are scattered and uncertain, with thoughts exploring in a large latent space. As reasoning progresses, reasoning thoughts distributions become increasingly concentrated around the correct answers, supported by the distribution density increasing in the correct answer regions. Specifically, in the early stages (0-20% and 20-40%), the reasoning process exhibits substantial exploration, where the model is uncertain about the answer. In subsequent stages (40-60%, 60-80%), the reasoning trajectory begins to narrow as the model identifies more promising directions. The model’s exploration in the latent space progressively refining the reasoning process. By the final stages (80-100%), KG-TRACES reaches all correct answers. This highlights how KG-TRACES, through explicit reasoning supervision and attribution-aware processes, guides the model to a stable and accurate conclusion.

Stage-wise metric distribution analysis. To further quantify this convergence behavior, we compute the mean and standard deviation for *Consistency*, *Uncertainty* and *Perplexity* across a subset

⁵We calculate *consistency* between reasoning process thoughts of each stage and final thoughts, *uncertainty* and *perplexity* between question and reasoning process thoughts of each stage

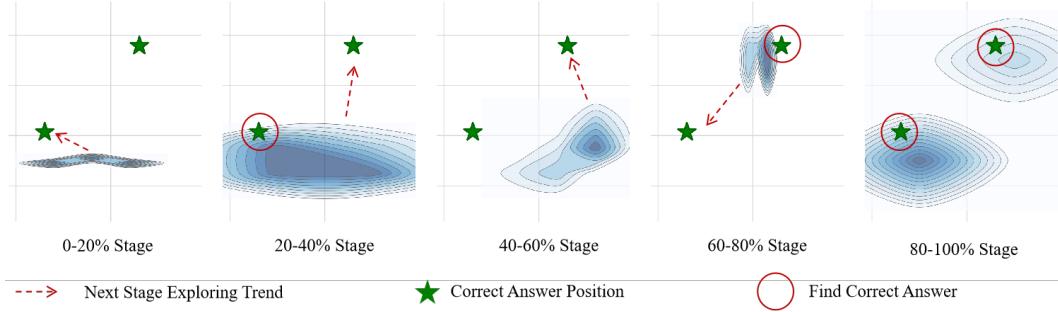


Figure 4: Visualization of model reasoning thoughts for a representative case in WebQSP. Darker color denotes higher reasoning process thoughts distribution density of the region. As reasoning progresses, thoughts distributions become sharper and align more closely with answers. **Example:** (Question: what year did the LA kings win the cup? Answers: 2012 Stanley Cup Finals, 2014 Stanley Cup Finals.)

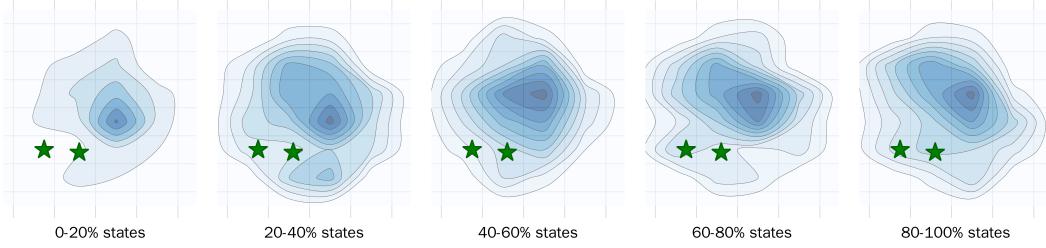


Figure 5: Visualization of model reasoning thoughts for a representative case in CWQ. **Example:** (Question: Who are the parents of one of the award nominees in the 18th Empire Awards ceremony? Answers: Gerald Hathaway, Kate McCauley Hathaway.)

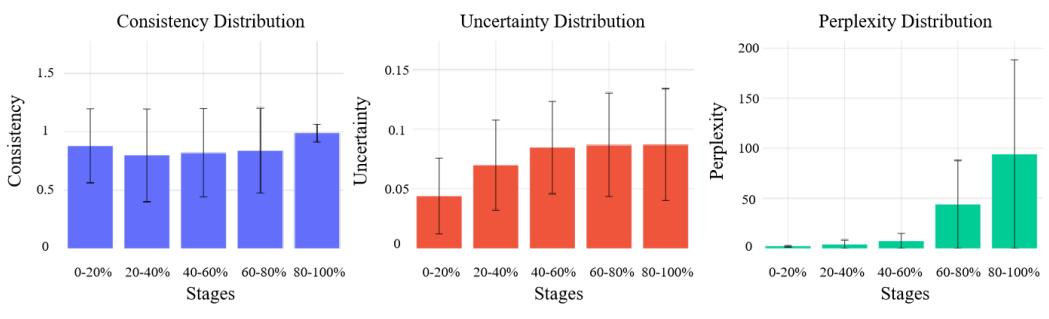


Figure 6: Visualization of step-wise reasoning process metrics distribution of KG-TRACES in WebQSP

of examples from the test dataset of WebQSP and CWQ. As shown in the Figure 6 and Figure 7, *consistency* steadily increases across stages, indicating that the model’s reasoning becomes more aligned with the correct answer as the reasoning process unfolds. *Uncertainty* and *Perplexity* also steadily increases across stages, indicating that the model explores more latent reasoning space as the reasoning process unfolds. These highlight the effectiveness of KG-TRACES in guiding the model toward explainable and attributable conclusions over reasoning steps. The analysis of more case and result of WebQSP and CWQ will be discussed in Appendix D.2.

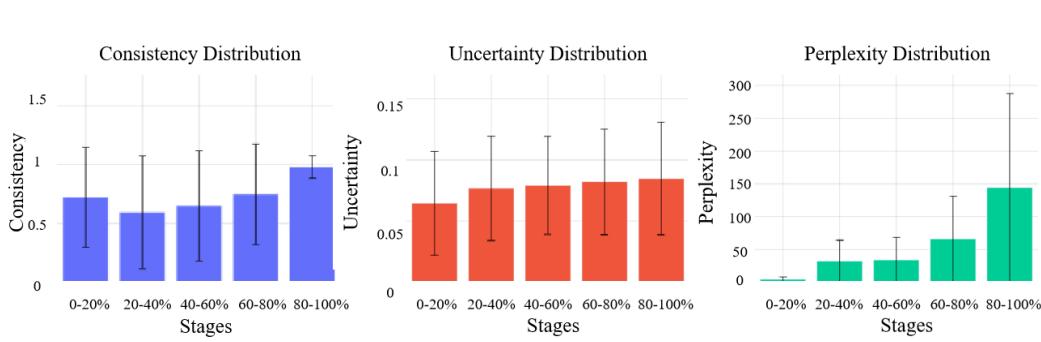


Figure 7: Visualization of step-wise reasoning metrics distributions of KG-TRACES in CWQ

5 Conclusion

We present KG-TRACES, a unified framework for training large language models to perform explainable, attributable reasoning guided by reasoning paths from knowledge graph. By supervising models on relation paths, triple paths, and attribution-aware reasoning processes, KG-TRACES enables faithful multi-step inference across both general and domain-specific scenarios. Through extensive experiments, we show that KG-TRACES achieves strong performance under varying KG accessibility, transfers effectively to unseen medical QA tasks, and produces stable reasoning trajectories. Our results highlight the value of structured symbolic path and attribution-aware reasoning processes supervision for enhancing both the accuracy and transparency of language model reasoning.

6 Limitations and broader impacts

While KG-TRACES demonstrates strong performance on both general and specialized QA tasks, several limitations remain. First, the reliance on symbolic supervision requires access to high-quality KG-derived paths, which may be incomplete or noisy in low-resource domains. Second, our current evaluation focuses on multi-hop QA task; extending to more diverse reasoning types (e.g., math, procedural) warrants further study. Future work includes exploring semi-supervised or reinforcement learning for symbolic path induction, and scaling our framework to more reasoning scenarios. Overall, this work has the potential to enhance interpretability and robustness in knowledge-intensive applications, facilitating more transparent and trustworthy AI systems.

References

- [1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [2] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.
- [3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [4] Yu Gu and Yu Su. ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering. In *Proceedings of the International Conference on Computational Linguistics*, pages 1718–1731, 2022.
- [5] Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 553–561, 2021.

- [6] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.
- [7] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1827–1843, 2023.
- [8] Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *International Conference on Learning Representations*, 2023.
- [9] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, 2023.
- [10] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213, 2022.
- [11] Yunshi Lan and Jing Jiang. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, 2020.
- [12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.
- [13] Jia Li, Ge Li, Yongmin Li, and Zhi Jin. Structured chain-of-thought prompting for code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–23, 2025.
- [14] Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian yun Nie, and Ji-Rong Wen. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Annual Meeting of the Association for Computational Linguistics*, pages 10879–10899, 2024.
- [15] Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.
- [16] Haochen Liu, Song Wang, Yaochen Zhu, Yushun Dong, and Jundong Li. Knowledge graph-enhanced large language models via path selection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6311–6321, 2024.
- [17] LINHAO Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*, 2024.
- [18] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. *arXiv preprint arXiv:2407.10805*, 2024.
- [19] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594, 2023.
- [20] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, 2016.
- [21] Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 11709–11724, 2024.
- [22] Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, 2022.

- [23] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, 2020.
- [24] Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, 2021.
- [25] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, 2019.
- [26] Haitian Sun, Bhuvan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, 2018.
- [27] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, 2024.
- [28] Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. SPARQA: Skeleton-based semantic parsing for complex questions over knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8952–8959, 2020.
- [29] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, 2018.
- [30] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.
- [31] Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 12047–12064, 2023.
- [32] Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. Knowledge-driven CoT: Exploring faithful reasoning in LLMs for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*, 2023.
- [33] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.
- [34] Yilin Wen, Zifeng Wang, and Jimeng Sun. MindMap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 10370–10388, August 2024.
- [35] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL*, pages 6233–6251, 2024.
- [36] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [37] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822, 2023.
- [38] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. Re-act: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.
- [39] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, 2021.

- [40] Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043, 2022.
- [41] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, 2016.
- [42] Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. DecAF: Joint decoding of answers and logical forms for question answering over knowledge bases. In *International Conference on Learning Representations*, 2023.
- [43] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488, 2022.
- [44] Han Zhang, Langshi Zhou, and Hanfang Yang. Learning to retrieve and reason on knowledge graph through active self-reflection. *arXiv preprint arXiv:2502.14932*, 2025.
- [45] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784, 2022.
- [46] Denny Zhou, Nathanael Schärlí, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*, 2023.
- [47] Zhanke Zhou, Zhaocheng Zhu, Xuan Li, Mikhail Galkin, Xiao Feng, Sanmi Koyejo, Jian Tang, and Bo Han. Landscape of thoughts: Visualizing the reasoning process of large language models. *arXiv preprint arXiv:2503.22165*, 2025.

Appendix

A Dataset Details	16
A.1 General Reasoning Dataset	16
A.2 Medical Domain Dataset	16
B Prompt Details	16
B.1 Multi-tasks Supervision Dataset Construct Prompts	16
B.2 Inference Prompts	18
C Training and Implementation Details	18
C.1 General Reasoning Task Baselines	18
C.2 Medical Reasoning Task Baselines	19
D Metrics and Scoring Details	19
D.1 Medical Reasoning Task Metrics	19
D.2 Reasoning Process Quality Metrics	19
E Additional Results	21
E.1 Influence of Reasoning Paths Number	21
E.2 Reasoning Process Quality and Visualizations (Additional Results)	21

A Dataset Details

A.1 General Reasoning Dataset

To evaluate KG-TRACES’s generative reasoning performance, we use two KGQA datasets: WebQuestionSP (WebQSP)[41] and Complex WebQuestions (CWQ) [29] as our main benchmark. Both datasets are grounded in the Freebase knowledge graph [1]. We follow previous works [17] to use the same train and test splits for fair comparison. The statistic of the datasets are given in Table 4. The statistics of the answer numbers and reasoning hops are presented in Table 7 and Table 8, respectively.

Table 4: Statistics of two generative reasoning datasets.

Dataset	Train	Validation	Test	Max hop
WebQSP	2826	246	1628	2
CWQ	27626	3519	3531	4

Table 5: Answer count distribution on WebQSP and CWQ. Ans_{num} denotes the number of answers per question.

Dataset	$Ans_{num} = 1$	$2 \leq Ans_{num} \leq 4$	$5 \leq Ans_{num} \leq 9$	$Ans_{num} \geq 10$
WebQSP	51.2%	27.4%	8.3%	12.1%
CWQ	70.6%	19.4%	6.0%	4.0%

Table 6: QA hops Statistics of WebQSP and CWQ.

Dataset	1 hop	2 hop	≥ 3 hop
WebQSP_train	62.89%	37.11%	0%
WebQSP_test	64.11%	35.88%	0%
CWQ_train	24.65%	57.23%	18.12%
CWQ_test	22.11%	57.89%	20.00%

A.2 Medical Domain Dataset

GenMedGPT-5k is a synthetic dataset consisting of 5,000 dialogues between patients and GPT-3.5, generated based on a structured disease knowledge base [15]⁶. Each dialogue begins with a patient question derived from real-world medical consultations in the iCliniq database, describing specific symptoms or health concerns. The model-generated responses include detailed medical reasoning, covering diagnosis, symptoms, treatment recommendations, and suggested medical tests. For fair comparison , we use the same 714 dialogues of previous work [34] to construct the test dataset used in our medical domain experiments.

B Prompt Details

B.1 Multi-tasks Supervision Dataset Construct Prompts

To generate training data for multi-tasks supervision, we design structured prompts that guide LLMs to produce symbolic relation paths, triple paths and explainable intermediate reasoning processes. These prompts are used in three contexts:

- **Full reasoning process generation:** The model is asked to simulate multi-step reasoning given the question, answer and potential supporting KG paths, while labeling each step’s supporting attribution (KG or model-inferred) and utility.
- **Relation path construction:** The model generates a valid sequence of relations that connects question entities to the answer.
- **Triple path construction:** The model outputs explicit triple-level paths, capturing symbolic reasoning paths supervision.

We show the detail prompts used for each setting in Table 7, 8 and 9.

⁶<https://github.com/KentOn-Li/ChatDoctor>

Table 7: Prompt template for constructing structured reasoning processes dataset of KG-TRACES

KG-TRACES Reasoning Process Construction Prompt
Question: {question}
Answer: {answer}
Potential useful reasoning path: The following reasoning paths are provided to help you understand relationships among entities and derive an answer: {reasoning_paths}
Task Instructions:
1. Goal:
<ul style="list-style-type: none"> • Use the given reasoning paths and answer to generate a detailed reasoning process for the original question, explicitly indicating the source of knowledge (e.g., from KG or inferred by LLMs). • Enhance the reasoning process by including special tokens to label each path's source and effectiveness: <ul style="list-style-type: none"> – <KG>: Knowledge directly from the knowledge graph. – <INFERRED>: Knowledge inferred by LLMs without explicit KG support. – <EFFECTIVE> / <INEFFECTIVE>: Whether the path effectively contributes to the final answer.
2. Specific Requirements:
<ul style="list-style-type: none"> • Path Selection and Labeling: <ul style="list-style-type: none"> – Filter out unnecessary paths: Only select paths directly relevant to the question. – Ignore paths marked as <INEFFECTIVE> when getting the final answer. – Label each selected path using the special tokens. • Dynamic Knowledge Utilization: <ul style="list-style-type: none"> – If no KG path applies, allow LLMs to infer logical connections using <INFERRED>, clearly marked.
3. Output Format:
Reasoning Process: [Output reasoning process here]
Example:
[Input] **Question**: Which film directed by Christopher Nolan starred Leonardo DiCaprio and was released in 2010? **Answer**: Inception **Retrieved Triples**: 1. Leonardo DiCaprio → film.actor.film → m.12345 2. m.12345 → film.director → Christopher Nolan ... 19. m.00000 → film.release_date → 2017 [Output] **Reasoning Process**: Step 1: Identify film starring Leonardo DiCaprio. - Relevant Triple: #1 [<KG> <EFFECTIVE>] - Note: Triples #10/#13 [<KG> <INEFFECTIVE>] Step 2: Directed by Christopher Nolan: #2 [<KG> <EFFECTIVE>] ... Final Answer: Inception

Table 8: Prompt template for constructing symbolic relation paths dataset of KG-TRACES

Path Construction Prompt (Relation Path)
Please generate a valid reasoning relation path that can be helpful for answering the following question.
Question: {question}

Table 9: Prompt template for constructing symbolic triple paths dataset of KG-TRACES

Path Construction Prompt (Triple Path)
Please generate a valid reasoning triple path that can be helpful for answering the following question.
Question: {question}

B.2 Inference Prompts

During evaluation, we apply two types of prompts depending on the model’s access to external knowledge:

- **No-path prompt:** The model answers each question independently, without any external symbolic reasoning paths information.
- **Path-informed prompt:** The model is supplied with retrieved or model generated reasoning paths, and must generate both the reasoning process and final answer while distinguishing between factual and inferred knowledge.

These inference prompts, shown in Table 10 and 11, are used consistently in all experimental evaluations of WebQSP and CWQ.

Table 10: Inference prompt for direct answer generation without any reasoning paths.

Inference Prompt (No Reasoning Path)
Please answer the following questions. Please keep the answer as simple as possible and return all the possible answers as a list. **Question**: {question}

Table 11: Inference prompt for reasoning process and answer generation with reasoning paths.

Inference Prompt (With Reasoning Paths)
Based on the reasoning paths, please answer the given question. The paths with [<KG>] are from the real world **Knowledge Graph** (more reliable) and the paths with [<INFERRED>] are your predictions. You should recognize useful reasoning paths from **Potential Useful Reasoning Paths**. Please generate both the reasoning process and answer. Please keep the answer as simple as possible and return all the possible answers as a list. **Potential Useful Reasoning Paths**: {reasoning_paths} **Question**: {question}

C Training and Implementation Details

For KG-TRACES, we use Qwen2.5-Chat-7B [36] as the LLM backbone, which is instruction finetuned on the training split of WebQSP and CWQ with Freebase for 3 epochs. In addition to the original datasets, we augment the training data by incorporating two additional types: QA-based and path-based SFT data (QA-based data consists of question-answer pairs, while path-based SFT data includes reasoning paths derived from KG). This data augmentation helps improve the model’s generalization and reasoning capabilities. The optimization objective during fine-tuning is minimizing the loss between generated text and target text. We set the maximum context length to 4096, padding each batch to match the longest sequence in that batch. The batch size is set to 4, the learning rate is set to 2e-5, and the gradient accumulation step is set to 16. We use the cosine learning rate scheduler policy with the warmup ratio set to 0.03. For LoRA fine-tuning, we utilized DeepSpeed, BF16 data type, and gradient checkpointing technology. The training is conducted on 6 A100-80G GPUs for 30 hours.

During inference stage of general reasoning task, we first adopt the model to generate top-K relation paths and triple paths with the highest probability. When inference with the predicted relation paths, we utilized the same method with [17] to retrieve reasoning paths in KG for each question to get reasoning paths, and prompt model to response. When inference with the predicted triple paths, we just link the triple paths according to head and tail entity to get reasoning paths.

During inference in the medical task, KG-TRACES is directly evaluated on the GenMedGPT-5k dataset without any further fine-tuning on medical knowledge. Since the model has not been exposed to the medical KG during training, it cannot generate relation paths; thus, only triple paths are predicted. We obtain reasoning paths by linking predicted triples through head–tail entity alignment, similar to the method used in the general reasoning task. For KG-augmented variants, we additionally retrieve reasoning paths by conducting entity linking based on the question entities to extract relevant subgraphs from the medical KG EMCKG follow previous work [34]. A fixed number (30) of reasoning paths are randomly sampled as retrieved context. All model response are generated conditioned on these reasoning paths via prompting.

C.1 General Reasoning Task Baselines

For the general reasoning tasks (WebQSP and CWQ), we evaluate KG-TRACES against a range of baseline methods that span several categories follow the previous work [17]:

- **Embedding-based methods:** These methods rely on embedding-based representations to match questions with relevant knowledge. This includes models like KV-Mem [20], which stores entities and their relationships

in memory for retrieval, and EmbedKGQA [23], which encodes knowledge graph into embeddings for question answering. Other models in this category include NSM [5], TransferNet [24] and KGT5 [22].

- **Retrieval-augmented methods:** These methods retrieve relevant information from knowledge graph or external databases to aid in answering questions. Notable models in this category include GraftNet [26], PullNet [25], and the more recent SR+NSM [45] and SR+NSM+E2E [45].
- **Semantic parsing methods:** These methods transform questions into formal queries over knowledge graph. SPARQL-based approaches, such as QGG [11], ArcaneQA [4] and RnG-KBQA [40], use graph querying to retrieve answers directly from knowledge graph.
- **Vanilla LLMs:** These methods rely solely on LLMs for question answering, including models like ChatGPT, Flan-T5 [2], and Alpaca-7B [30].
- **Prompt Augmented LLMs.** To evaluate the impact of prompting strategies on LLMs, we additionally consider a set of *Prompt Augmented LLMs* that incorporate CoT reasoning instructions into the input. Specially, we evaluate models like LLaMA3.1-Chat-8B [3], Qwen2.5-Chat-7B [36], and ChatGPT with a CoT prompt prepended to the question, encouraging the model to reason step by step before outputting an answer.
- **LLMs+KG:** These methods combine LLMs with knowledge graph to improve reasoning over structured data. Models such as KD-CoT [32], UniKGQA [8], and RoG [17] fall into this category.

For each of the baselines, we report Hits@1 and F1 scores on both WebQSP and CWQ datasets to evaluate their performance.

C.2 Medical Reasoning Task Baselines

In the medical reasoning domain, we compare KG-TRACES with baselines designed specifically for medical question answering. These baselines include retrieval-based methods, vanilla LLMs, and methods that integrate knowledge graph for medical reasoning. The following categories represent the main baselines evaluated:

- **Retrieval-augmented methods:** These models enhance question answering by retrieving relevant medical knowledge from external sources. Notable methods in this category include BM25 Retriever and Embedding Retriever [34], which retrieve relevant medical knowledge for ChatGPT to generate responses.
- **Vanilla LLMs:** This category includes LLMs that do not use external knowledge graph but rely solely on the model’s internal knowledge base. We use models such as ChatGPT and GPT-4 for comparison, as these models have shown strong performance in natural language understanding and generation tasks.
- **LLMs + KG:** These models combine LLMs with external medical knowledge graph to improve the quality of medical reasoning. MindMap [34] integrates symbolic reasoning using KG-based prompts and has been previously demonstrated to perform well in medical question answering tasks.

For the medical reasoning task, we evaluate the performance of each model using the same evaluation criteria and metrics, such as Relevance, Accuracy, Completeness, Clarity, and Conciseness, which are described in detail in Appendix D.1.

D Metrics and Scoring Details

D.1 Medical Reasoning Task Metrics

All models are scored using the *qwen-plus*⁷ API, with each response rated 3 times and averaged for stability. Detail evaluation prompt is in the Table 12.

D.2 Reasoning Process Quality Metrics

In this section, we describe the metrics used to evaluate the quality of the reasoning process in the context of LLMs. Follow the previous work [47], we focus on three key metrics—**Consistency**, **Uncertainty**, and **Perplexity**—which are designed to measure the model’s reasoning stability and confidence during its multi-step reasoning process. These metrics are calculated on a set of sampled data and provide insights into the model’s performance at different stages of reasoning.

Metric Definitions and Calculations. The three metrics used in this work are defined and calculated as follows:

1. **Consistency:** Consistency measures the degree to which the model’s reasoning remains stable over multiple reasoning steps. Specifically, we compute consistency by comparing the reasoning states at each step with the final state. If the model’s reasoning process converges toward the correct answer, we expect higher consistency. The formula for consistency is:

⁷*qwen-plus* refers to the API-accessible version of Alibaba’s language model, evaluated via <https://www.aliyun.com/product/bailian>.

Table 12: Prompt used for evaluating model responses in the medical reasoning task, with five human-aligned criteria and detailed instructions.

Medical Reasoning Task Evaluation Prompt
Reference Information: {reference}
Answer to Score: {answer}
Task: Evaluate the given answer based on the provided reference information using the following criteria. Assign a score between 0 and 1 (inclusive) for each criterion, in increments of 0.1. A score of 1 means the answer fully meets the criterion, while a score of 0 means the answer fails to meet the criterion at all.
Evaluation Criteria:
1. **Relevance** (Score: 0-1): This criterion assesses how well the answer aligns with the reference information, addressing the symptoms, diagnosis, and treatments mentioned. The answer should directly respond to the medical context and conditions outlined in the reference. Answers that focus on the core issues presented, without deviating into irrelevant areas, should score higher.
2. **Accuracy** (Score: 0-1): The accuracy score reflects how correctly the answer represents the facts outlined in the reference. This includes correct medical terminology, diagnosis, and treatment recommendations. An answer should avoid introducing false or unsupported information while accurately reflecting the key aspects of the reference, including the medical procedures and conditions described.
3. **Completeness** (Score: 0-1): Completeness is assessed based on how thoroughly the answer covers the key points mentioned in the reference, including diagnostic procedures, symptoms, and treatment options. A complete answer should address all aspects of the medical condition mentioned in the reference, offering a full response to the query with relevant details. Missing important diagnostic tests or treatment steps will reduce the score.
4. **Clarity** (Score: 0-1): This criterion evaluates the clarity and readability of the answer. A high score is awarded to responses that are well-structured, logically coherent, and easily understood. An answer that communicates its reasoning in a clear and concise manner without ambiguity or unnecessary complexity will score higher.
5. **Conciseness** (Score: 0-1): This criterion evaluates how succinctly the answer conveys necessary information. Answers should avoid redundancy and irrelevant details but should not be penalized for adding depth and reasoning to the response. A longer, well-reasoned response that covers all necessary aspects of the reference will be rewarded, provided it does not become excessively verbose.
Response Format: Provide the evaluation results in the following format:
Score Breakdown: <ul style="list-style-type: none"> - **Relevance**: X.X (Explanation: [Provide brief reasoning for the score based on how well the answer aligns with the reference information and medical context]) - **Accuracy**: X.X (Explanation: [Provide brief reasoning for the score based on the accuracy and consistency of the answer with the reference]) - **Completeness**: X.X (Explanation: [Provide brief reasoning for the score based on the coverage of key points in the reference information]) - **Clarity**: X.X (Explanation: [Provide brief reasoning for the score based on how clearly the answer is expressed]) - **Conciseness**: X.X (Explanation: [Provide brief reasoning for the score based on how focused and concise the answer is])

$$\text{Consistency}(s_i) = \mathbf{I}(\arg \min s_i = \arg \min s_n) \quad (6)$$

where s_i represents the reasoning state at the i -th step, and s_n is the final reasoning state. The indicator function \mathbf{I} outputs 1 if the states are identical (indicating convergence) and 0 otherwise.

2. **Uncertainty:** Uncertainty quantifies how confident the model is about its predictions at intermediate steps. Higher uncertainty values indicate less confidence in the reasoning process. The uncertainty at a given step is calculated as the entropy of the state probabilities:

$$\text{Uncertainty}(s_i) = - \sum_{d \in s_i} d \cdot \log d \quad (7)$$

where d represents the probability of a given state. This metric provides a measure of the model's confidence in the reasoning path taken.

3. **Perplexity:** Perplexity evaluates how well the model predicts the next token in the reasoning process, providing an indication of its confidence in the generated thoughts. Lower perplexity values correspond to more confident predictions. The formula for perplexity is:

$$\text{Perplexity}(t_i) = p_{\text{LLM}}(t_i|s_{i-1})^{-1/|t_i|} \quad (8)$$

where t_i is the i -th token and s_{i-1} is the previous state. The calculation measures how likely the model is to generate the reasoning tokens at each stage, normalized by the token length.

Data Sampling and Metrics Calculation. For evaluating the reasoning quality, we sampled 500 question-answer pairs from WebQSP and CWQ dataset. For each of these 500 QA pairs, we performed 10 independent inference runs using the model, generating 10 distinct reasoning paths per question. These repeated inferences allow for a more robust analysis of the reasoning process across multiple runs.

The reasoning paths were visualized for each case, and the consistency, uncertainty, and perplexity metrics were calculated for each of the 10 inference results per question. The metrics were then aggregated across the 500 samples to obtain an overall understanding of the reasoning behavior.

Case-Level Visualization. To qualitatively understand the model’s reasoning dynamics, we visualize each reasoning trajectory using landscape plots that depict the distribution of intermediate reasoning states. Following the method introduced in [47], the landscape is constructed by projecting the distance between intermediate thoughts and the final answer into a two-dimensional space. While the original work focuses on multiple-choice QA (where the distance is defined as perplexity between thoughts and answer options), we extend this formulation to open-ended QA by measuring the perplexity between thoughts and multi-answer targets. Each plot is divided into equal length ratio reasoning process segments, and density contours reflect the concentration of model-generated states. Darker regions indicate higher thought density, while green stars mark ground-truth answers. This allows us to track how the model’s hypotheses evolve and converge, highlighting its alignment behavior and the effectiveness of symbolic reasoning during multi-hop inference.

Distribution-Level Analysis. In addition to case-level visualizations, we also performed a statistical analysis over the 500 sampled QA pairs for WebQSP and CWQ. For the analysis, we calculated average of consistency, uncertainty, and median of perplexity across the reasoning steps. These values were then aggregated to analyze the overall trend of the model’s reasoning behavior across the entire set of questions. The results are presented as distributions, showing how these metrics change at each reasoning stage.

E Additional Results

E.1 Influence of Reasoning Paths Number

To understand how the number of candidate reasoning paths affects KG-TRACES, we varied the beam search number from 1 to 5 during relation path prediction. Figure 8 illustrates the performance (Hit@1, F1, Precision, Recall) and the average number of reasoning paths ('Paths num') on WebQSP and CWQ. As the beam search number increases, more candidate reasoning paths are retrieved, but this does not always translate to better performance. On WebQSP, key metrics like Hit@1 and F1 generally improve up to a beam search number of 3 or 4, after which performance slightly declines. For the more complex CWQ dataset, these metrics peak at a beam search number of 3, with a noticeable dip at 4 before a partial recovery at 5. These results suggest an optimal beam search size exists (around 3-4 for WebQSP and 3 for CWQ in this setting), balancing sufficient path exploration with the risk of introducing noise from less relevant paths.

E.2 Reasoning Process Quality and Visualizations (Additional Results)

As Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, and Figure 14 show, we provide other 3 reasoning process visualization case of KG-TRACES in WebQSP and CWQ respectively.

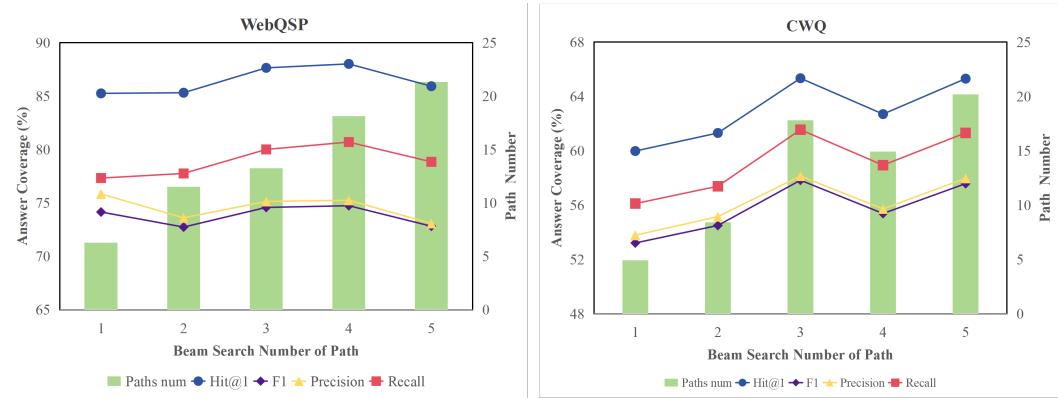


Figure 8: KG-TRACES performance comparison based on beam search number of reasoning path. We compare KG-TRACES with relation path using beam search number from 1 to 5 in WebQSP and CWQ.

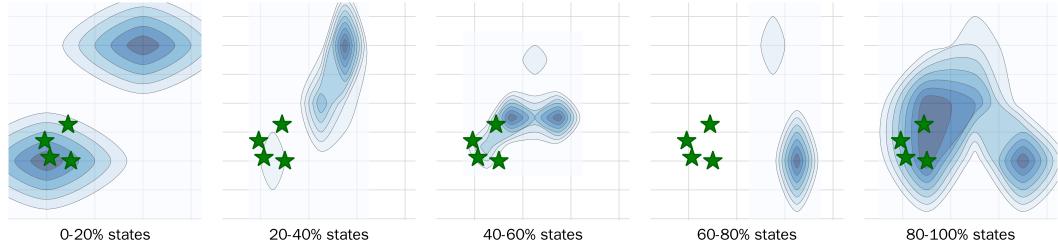


Figure 9: Visualization of model reasoning thoughts for a representative case 1 in WebQSP



Figure 10: Visualization of model reasoning thoughts for a representative case 2 in WebQSP

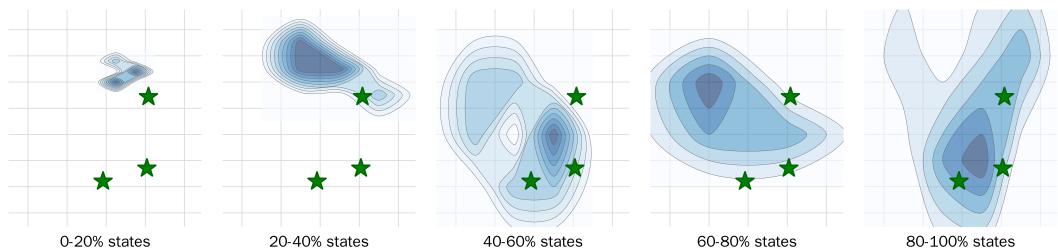


Figure 11: Visualization of model reasoning thoughts for a representative case 3 in WebQSP

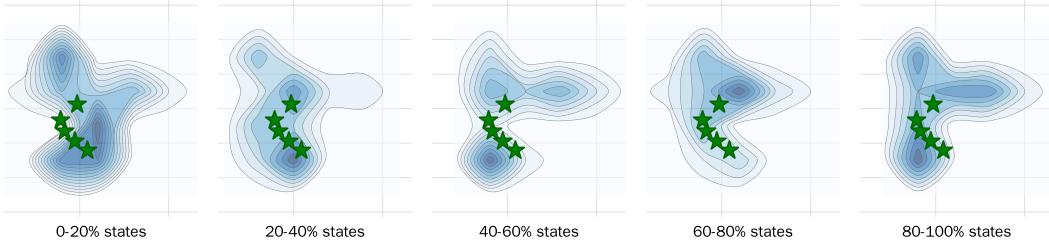


Figure 12: Visualization of model reasoning thoughts for a representative case 1 in CWQ

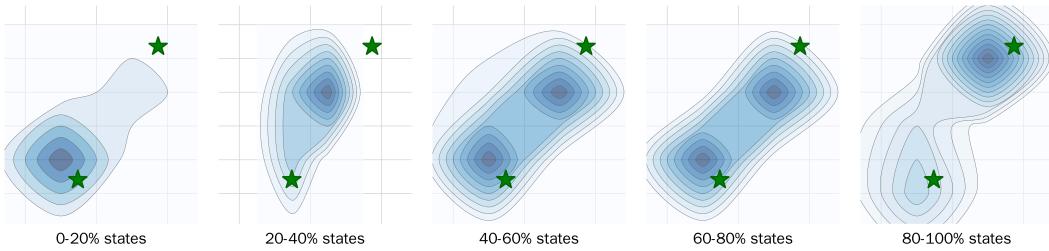


Figure 13: Visualization of model reasoning thoughts for a representative case 2 in CWQ

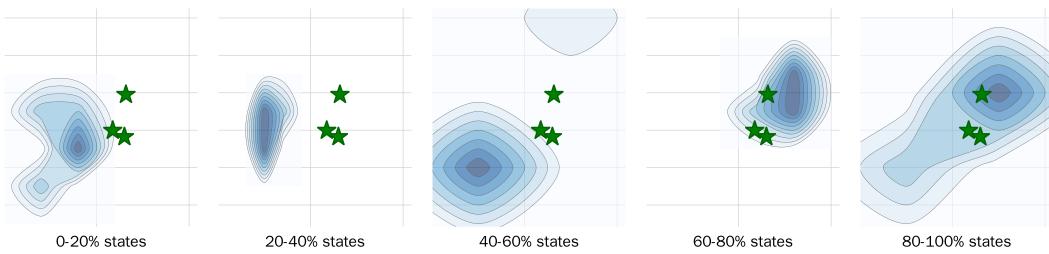


Figure 14: Visualization of model reasoning thoughts for a representative case 3 in CWQ